

ESERCITAZIONI CLASSE 3BI 2020/2021

Raccolta curata dall'alunno Alessandro Esposito

Sommario

Logo, Disegnare un Quadrato	1
Logo, Disegnare una Linea tratteggiata	2
Logo, Disegnare un Triangolo equilatero.....	3
Logo, Disegnare una Casa	4
Logo, Disegnare una Casa con due porte.....	5
Logo, Disegnare un Villaggio	6
Logo, Disegnare un Quadrato parametrizzato.....	7
Logo, Disegnare un Triangolo parametrizzato	8
Logo, Disegnare un Poligono con procedura	9
Logo, Usare i Colori	10
Logo, Disegnare una Linea tratteggiata parametrizzata.....	11
Logo, Disegnare un Quadrato di lato Random.....	12
Logo, Disegnare un Quadrato di lato Random con un parametro	13
Logo, Stampare se il numero dato in input è pari o dispari.....	14
Logo, Stampare i divisori del numero dato in input	15
C++, Calcolare il perimetro di un quadrato.....	16
C++, Stampare la tabellina del 2	17
C++, Stampare la tabellina del 2 con la variabile num.....	18
C++, Stampare la tabellina del 2 con il ciclo for	19
C++, Stampare i numeri in input maggiori di 10	20
C++, Stampare i numeri in input maggiori di 5	21
C++, Stampare se il numero in input è positivo o negativo.....	22
C++, Stampare il doppio del numero in input se è positivo altrimenti il triplo se è negativo	23
C++, Stabilire se il numero in input è pari o dispari	24
C++, Stampare i numeri in input moltiplicati per 10 se sono pari, per 1 se sono dispari.....	25
C++, Stampare un carattere	26
C++, Somma di due numeri.....	27
C++, Somma di tre numeri	28
C++, Menu con il ciclo do while	29

Domanda, Marte.....	30
Domanda, biologia marina.....	31
C++, Menu con switch.....	32
C++, Menu Area	33
Porte Logiche,(X and Ynegato or Z).....	34
Porte Logiche AND OR NOT	35
Porte Logiche,(NOT(X AND Y) OR Z).....	36
C++, stampare a video 6 numeri con un ciclo indefinito	37
C++, stampare a video 6 numeri con un ciclo definito	38
C++, tecnica della somma	39
C++, tecnica della somma con un ciclo indefinito.....	40
C++, tecnica del contatore	41
C++, connettivi logici.....	42
C++, cicli for, do while, while	43
C++,procedura ordina due caratteri	44
C++, funzione somma con tre parametri	45
C++, numeri dispari compresi tra 200 e 1000.....	46
C++,area trapezio, rettangolo e triangolo.....	47
C++, scelta del sottoproblema utilizzando menu con switch e do while.....	48
C++, somma di 5 numeri	49
Cittadinanza, nuove tecnologie	50
C++, dati n caratteri stampare quante volte viene inserito il carattere z o n.....	51
C++, creare un vettore, inserire 5 caratteri e mostrarli a video	52
C++, creare un vettore, inserire 9 caratteri e mostrare a video le a inserite	53
C++, creare un vettore, acquisire la dimensione logica compresa tra 0 e la dimensione fisica.....	54
C++, creare un vettore, acquisire la dimensione logica con il passaggio di parametri	55
C++, calcolare il perimetro di un esagono	56
C++, calcolare il perimetro di un esagono eliminando l'interfaccia utente.....	57
C++, calcolare il perimetro di un esagono usando una funzione parametrizzata	58
C++, calcolare il perimetro di un poligono.....	59
C++, dati n caratteri stampare a video quante volte viene inserito il carattere a.....	60
C++, somma degli elementi di un vettore.....	61

C++, caricare un vettore di interi e sommare i suoi elementi	62
C++, caricare un vettore e contare quanti dei suoi elementi sono maggiori o uguali a 10.....	63
C++, caricare un vettore di 5 numeri interi e sommare i negativi.....	64
C++, caricare un vettore di interi finchè non viene inserito 0, poi sommare i suoi elementi	65
C++, dato un vettore assegnare al terzo posto un valore inserito da tastiera	66
C++, caricare e leggere un vettore con valori interi casuali compresi tra 0 e 300	67
C++, caricare un vettore con valori interi casuali compresi tra 0 e 300 e raddoppiare i valori.....	68
C++, caricare un vettore con interi casuali tra 0 e 300 e raddoppiare i compresi tra 10 e 100	69
C++,Caricare un vettore di caratteri utilizzando le procedure caricaVet e stampaVet. Contare le "F" di un vettore utilizzando la funzione contaF.	70
C++,Scrivere la procedura che stampa le posizioni del carattere F nel vettore di caratteri	71
C++,Contare le F in un vettore e stampare le posizioni con i parametri.....	73
C++, Data la lunghezza e il numero di lati calcolare il perimetro	74
C++,scrivere la procedura per scambiare il valore di due variabili per indirizzo.....	75
C++,Dato un vettore di interi scambiare il primo elemento con l'ultimo	76
C++,Caricare 3 vettori di interi e stamparli a video	77
C++,Copiare gli elementi di un vettore in un altro vettore.....	78
C++, Raddoppiare gli elementi di un vettore e memorizzarli in un altro vettore.....	79
C++,Sommare due vettori.....	80
C++,Cicli annidati	81
C++,stampare a video la tavola pitagorica.....	82
C++,caricare un vettore di interi e ordinarlo.....	83
C++, Caricare un vettore di interi e ordinarlo usando le procedure	84
C++,Caricare un vettore di interi e ordinarlo usando le procedure e i parametri.....	85
C++,Ricerca Lineare	86
Python, sintesi vocale	87
Python, sintesi vocale e conteggio parole	88
Python,sintesi vocale e conteggio parole e vocali	89
Python, sintesi vocale e conteggio parole e vocali di un file esterno.....	90
Principali differenze tra Python e C++	91

C++, scambio di elementi al intero del vettore.....	92
C++, procedura update	93
C++, procedura update con due vettori.....	94
Python e C++, conto alla rovescia	95
Python e C++,conto in avanti e conto alla rovescia	96
Python, selezione a una via.....	97
Python e C++, somma di due vettori	98
Python, cambiare il colore di sfondo della tartaruga	99
Python, disegnare un cerchio in loop infinito con la tartaruga	100
Python e C++, data una città stampare a video il numero degli abitanti	101
Python e C++, data una città stampare a video il numero degli abitanti (stringhe).....	102
C++, operazioni con le temperature delle città (vettori paralleli)	103
HTML, realizzare una pagina contenente un messaggio	104
HTML, realizzare una pagina web contenente un titolo.....	105
HTML, realizzare una pagina web con i tag h1...h6 e center	106
HTML, realizzare una pagina web con i tag strong, p e l'attributo text.....	107
HTML, realizzare una pagina con un collegamento ipertestuale.....	108
HTML, realizzare una pagina con un collegamento ipertestualeed un segnalibro	109
HTML, realizzare una pagina con un collegamento ipertestuale ed un segnalibro esterno	110
HTML, realizzare una pagina contenente elenchi puntati annidati.....	111
HTML, realizzare una pagina contenente un elenco numerato	112
HTML, realizzare una pagina contenente un elenco misto (numerato/puntato)	113
HTML, realizzare una pagina contenente una tabella	114
HTML, realizzare una pagina contenente una tabella	115
HTML, realizzare una pagina contenente una tabella dei voti	116
HTML, realizzare una pagina contenente una tabella dei voti	117
HTML, realizzare una pagina contenente una CheckBox.....	118
HTML, realizzare una pagina contenente una ComboBox.....	119
C++, caricare e stampare una matrice	120
C++, caricare, calcolare la media degli elementi e stampare una matrice.....	121

C++, data una matrice 3x4 calcolare la somma degli elementi della seconda riga	122
C++, data una matrice MxN calcolare la media della colonna scelta	123
C++, dato un vettore trovare il numero massimo all'interno di esso.....	124
C++, dati due vettori paralleli trovare la città con temperatura minima	125
C++, gestire i voti si una classe.....	126
JavaScript, scrivere Ciao Mondo in una pagina web.....	127
JavaScript, scrivere nella pagina web la scelta effettuata dall'utente	128
JavaScript, scrivere nella pagina web la stringa inserita dall'utente	129
C++, Sommare gli elementi della diagonale principale di una matrice quadrata.....	130
C++, Sommare gli elementi della diagonale secondaria di una matrice quadrata.....	131
C++, dati due record visualizzare chi è la persona con età maggiore.....	132
C++, dati tre record visualizzare in ordine crescente rispetto l'anno e auto che costa meno	133
C++, caricare e stampare a tabella un vettore di array con le procedure	134
C++, calcolare la media delle altezze di un array di record	135
C++, eseguire lo shift a sinistra di un vettore di caratteri.....	136
C++, eseguire lo shift a destra di un vettore di caratteri	137
C++, contare il numero di vocali per ogni riga di una matrice.....	138
C++, ordinare in modo crescente ogni colonna di una matrice N*M.....	139
C++, Sommare gli elementi dispari di ogni riga di una matrice N*M	140
JavaScript, Stampare una matrice in JavaScript	141
C++, Contare i numeri dispari di una matrice N*M di interi in C++.....	142
Python, Contare i numeri dispari di una matrice N*M di interi in Python	143
JavaScript, Contare i numeri dispari di una matrice N*M di interi in JavaScript.....	144
C++, Sommare gli elementi la cui somma degli indici è pari	145
C++, Data una matrice di record trovare il prezzo massimo della terza riga.....	146
C++, rappresentare il voto finale di una materia degli alunni in una sezione	147
Prolog, rappresentare la base di conoscenza	148

ESERCITAZIONI

1. DISEGNARE UN QUADRATO

ITIS-LS “Francesco Giordani” Caserta

Anno scolastico: 2020/2021 Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 01/03/2020

Numero progressivo dell’esercizio: es1

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Logoit versione 6.4

Obiettivo didattico: Conoscere l’ambiente di programmazione

Obiettivo del programma: disegnare un quadrato

Codice:

per quadrato

a 50 d 90 a 50 d 90 a 50 d 90 a 50 d 90

fine

2. DISEGNARE UNA LINEA TRATTEGGIATA

ITIS-LS “Francesco Giordani” Caserta

Anno scolastico: 2020/2021 Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 01/03/2020

Numero progressivo dell’esercizio: es2

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Logoit versione 6.4

Obiettivo didattico: Conoscere l’ambiente di programmazione

Obiettivo del programma: disegnare una linea tratteggiata

Codice:

per lineaTratteggiata

giu a 50 su a 50 giu a 50 su a 50 giu a 50 su a 50 giu a 50 su

fine

3. DISEGNARE UN TRIANGOLO

ITIS-LS “Francesco Giordani” Caserta

Anno scolastico: 2020/2021 Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 02/03/2020

Numero progressivo dell’esercizio: es3

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Logoit versione 6.4

Obiettivo didattico: Conoscere i Cicli

Obiettivo del programma: disegnare un triangolo equilatero

Codice:

per triangolo

a 50 d 120 a 50 d 120 a 50

fine

4. DISEGNARE UNA CASA

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 05/03/2020

Numero progressivo dell'esercizio: es4

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Logoit versione 6.4

Obiettivo didattico: cicli e procedure, divide et impera, top down

Obiettivo del programma: disegnare una casa

Codice:

per casa

per quadrato

ripeti 10 [a 50 d 90]

fine

per rettangolo

ripeti 2[a 30 d 90 a 20 d 90]

fine

per triangolo

ripeti 3[a 50 d 120]

fine

quadrato

su a 50 giu

d 30 giu

triangolo

su s 30 su

i 50 giu su

d 90 a 15 s 90 giu

rettangolo

fine

5. DISEGNARE UNA CASA CON DUE PORTE

ITIS-LS “Francesco Giordani” Caserta

Anno scolastico: 2020/2021 Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 01/03/2020

Numero progressivo dell’esercizio: es5

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Logoit versione 6.4

Obiettivo didattico: cicli e procedure, top down, divide et impera

Obiettivo del programma: disegnare una casa con due porte

Codice:

per casaDuePorte

quadrato

su a 50 giu

d 30 giu

triangolo

su s 30 su

i 50 giu su

d 90 a 5 s 90 giu

rettangolo

su tana

d 90 a 25 s 90 giu

rettangolo

fine

6. DISEGNARE UN VILLAGGIO

ITIS-LS “Francesco Giordani” Caserta

Anno scolastico: 2020/2021 Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 06/03/2020

Numero progressivo dell’esercizio: es6

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Logoit versione 6.4

Obiettivo didattico: cicli e procedure, top down, divide et impera

Obiettivo del programma: disegnare un insieme di case come villaggio

Codice:

per villaggio

casa

su tana

d 90 a 100 giu

casa

su tana

s 90 a 50 giu casa

fine

7. DISEGNARE UN QUADRATO PARAMETRIZZATO

ITIS-LS “Francesco Giordani” Caserta

Anno scolastico: 2020/2021 Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 08/03/2020

Numero progressivo dell’esercizio: es7

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Logoit versione 6.4

Obiettivo didattico: procedure con parametri

Obiettivo del programma: disegnare un quadrato parametrizzato

Codice:

```
per quadratoParametrizzato :lato
```

```
ripeti 4 [ a :lato d 90]
```

```
fine
```

8. DISEGNARE UN TRIANGOLO PARAMETRIZZATO

ITIS-LS “Francesco Giordani” Caserta

Anno scolastico: 2020/2021 Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 08/03/2020

Numero progressivo dell’esercizio: es8

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Logoit versione 6.4

Obiettivo didattico: procedure con parametri

Obiettivo del programma: disegnare un triangolo con parametri

Codice:

```
per triangoloParametrizzato :lato
```

```
  ripeti 3[ a :lato d 360/:lato]
```

```
  fine
```

9. DISEGNARE UN POLIGONO CON PROCEDURA

ITIS-LS “Francesco Giordani” Caserta

Anno scolastico: 2020/2021 Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 08/03/2020

Numero progressivo dell’esercizio: es9

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Logoit versione 6.4

Obiettivo didattico: cicli e procedure

Obiettivo del programma: disegnare un poligono regolare sapendo il numero di lati e la loro lunghezza

Codice:

```
per poligono :numLati :lungLati
```

```
  ripeti :numLati[ a :lungLati d 360/:numLati]
```

```
  fine
```

10. USARE I COLORI

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 08/03/2020

Numero progressivo dell'esercizio: es10

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Logoit versione 6.4

Obiettivo didattico: usare i colori

Obiettivo del programma: disegnare un quadrato colorato

Codice:

```
per quadratoColorato
```

```
giu
```

```
ascp[0 0 0]
```

```
ascr[ 255 0 0 ]
```

```
ripeti 4[a 100 d 90]
```

```
su d 45 a 5
```

```
riempi
```

```
fine
```

11. DISEGNARE UNA LINEA TRATTEGGIATA PARAMETRIZZATA

ITIS-LS “Francesco Giordani” Caserta

Anno scolastico: 2020/2021 Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 08/03/2020

Numero progressivo dell’esercizio: es11

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Logoit versione 6.4

Obiettivo didattico: cicli e parametri

Obiettivo del programma: disegnare una linea tratteggiata con parametri

Codice:

```
per tratteggioParametrizzato :n :lung
```

```
ripeti :n[a :lung su a :lung giu]
```

```
fine
```


12. DISEGNARE UN QUADRATO DI LATO RANDOM

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 09/03/2020

Numero progressivo dell'esercizio: es12

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Logoit versione 6.4

Obiettivo didattico: assegnazione e numeri casuali

Obiettivo del programma: disegnare un quadrato di lato casuale

Codice:

```
per quadratoRandom
```

```
as "lato acaso 100
```

```
ripeti 4[ a :lato d 90]
```

```
fine
```

13. DISEGNARE UN QUADRATO DI LATO RANDOM CON PARAMETRI

ITIS-LS “Francesco Giordani” Caserta

Anno scolastico: 2020/2021 Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 09/03/2020

Numero progressivo dell’esercizio: es13

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Logoit versione 6.4

Obiettivo didattico: assegnazione, numeri casuali e procedure

Obiettivo del programma: disegnare un quadrato di lato random con un parametro che stabilisce la massima lunghezza del lato

Codice:

```
per quadratoRandom2 :maxLong
```

```
as "lato acaso :maxLong
```

```
ripeti 4[a :lato d 90]
```

```
fine
```

14. STAMPARE SE IL NUMERO IN INPUT E' PARI O DISPARI

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 10/03/2020

Numero progressivo dell'esercizio: es14

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Logoit versione 6.4

Obiettivo didattico: stampa e strutture di controllo (se)

Obiettivo del programma: stampare se il numero dato in input è pari o dispari

Codice:

```
per pariDispari :n
```

```
se (resto :n 2) = 0 [stampa [pari]]
```

```
se (resto :n 2) = 1 [stampa [dispari]]
```

```
fine
```

15. STAMPARE TUTTI I DIVISORI DEL NUMERO IN INPUT

ITIS-LS “Francesco Giordani” Caserta

Anno scolastico: 2020/2021 Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 10/03/2020

Numero progressivo dell’esercizio: es15

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Logoit versione 6.4

Obiettivo didattico: stampa e strutture di controllo (se)

Obiettivo del programma: stampare tutti i divisori del numero dato in input

Codice:

Per divisori :n

ripeti :n [se (resto :n iterazioni) = 0 [stampa iterazioni]]

fine

16. CALCOLARE IL PERIMETRO DI UN QUADRATO

ITIS-LS “Francesco Giordani” Caserta

Anno scolastico:2020/2021 Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 12/10/2020

Numero progressivo dell’esercizio: es16

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: code::blocks ver 17.12

Obiettivo didattico:iniziare a conoscere il C++

Obiettivo del programma:Calcolare il perimetro di un quadrato

Codice:

```
#include <iostream>
using namespace std;
int perimetro (int latoPar);
int main()
{
    int lato;
    cout << “inserisci la misura del lato: “;
    cin >> lato;
    cout << “il valore del perimetro è: “ <<perimetro ( lato );
    return 0
}
int perimetro (int latoPar)
{
    return latoPar * 4;
}
```

17. 18. 19. STAMPARE LA TABELLINA DEL 2

/**

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 23/10/2020

Numero progressivo dell'esercizio: es18

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: code::blocks ver 20.03

Obiettivo didattico: uso delle variabili, funzione if in C++

Obiettivo del programma: stampare la tabellina del 2

```
*/  
/**  
#include <iostream>  
using namespace std;  
int main()  
{  
    cout << 2*1 << " " << 2*2 << " " << 2*3 << " " << 2*4 << " " << 2*5 << endl << 2*6  
<< " " << 2*7 << " " << 2*8 << " " << 2*9 << " " << 2*10 << endl;  
    return 0;  
}  
*/  
/**  
#include <iostream>  
using namespace std;  
int main()  
{  
    int num = 2;  
    cout << num * 1 << " " << num * 2 << " "  
<< num * 3 << " " << num * 4 << " " << num * 5  
<< " " << num * 6 << " " << num * 7 << " "
```

```

        << num *8 << " " << num * 9 << " " << num * 10
        << " " << endl;
        return 0;
}
*/
/**
#include <iostream>
using namespace std;
int main()
{
    int i,a;
    int (a=2);
    for(i = 1; i <= 10; i++){
        cout<<a*i;
        cout<<"\n";
    }return 0;
}
*/
#include <iostream>
using namespace std;
int main()
{
    int i,num;
    int (num=2);
    for(i = 1; i <= 10; i++)
    {
        if (i==5) cout<<endl;
        cout<<" " << num*i;
        else:
        cout<<" " << num*i;
    }
    return 0;}

```

20. Stampare i numeri in input maggiori di 10

/**

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni

Data: 24/10/2020

Numero progressivo dell'esercizio: es20

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: code::blocks ver 20.03

Obiettivo didattico: l'alunno è in grado di usare for e if

Obiettivo del programma: acquisire 5 numeri da tastiera e stampare a video quelli maggiori di 10
*/

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int num;
```

```
    for (int i=1; i<=5; i++)
```

```
    {
```

```
        cout << "inserisci un numero intero: " ;
```

```
        cin >> num;
```

```
        if (num>10) cout << num << endl;
```

```
    }
```

```
    return 0;
```

```
}
```

21. Stampare i numeri in input maggiori di 5


```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 26/10/2020
Numero progressivo dell'esercizio: es21
Versione: 1.1
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: l'alunno è in grado di usare for e if
Obiettivo del programma: acquisire 5 numeri da tastiera e stampare a video quelli maggiori di 5
*/
#include <iostream>
using namespace std;
int main()
{
    int num;
    for (int i=1; i<=5; i++)
    {
        cout << i << " " <<"inserisci un numero intero: ";
        cin >> num;
        if (num>10) cout << num << endl;
    }

    return 0;
}
```

22. Stampare se il numero in input è positivo o negativo

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 27/10/2020
Numero progressivo dell'esercizio: es22
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: l'alunno è in grado di usare for e la selezione a due vie (if)
Obiettivo del programma: acquisire 3 numeri da tastiera e stampare a video se il numero inserito è positivo
o negativo
*/

#include <iostream>
using namespace std;
int main()
{
    int num;
    for (int i=1; i<=3; i++)
    {
        cout << i << ")" <<"inserisci un numero intero: ";
        cin >> num;
        if (num>=0)
        {
            cout << "il numero inserito e' positivo"<<endl;
        }
        else
        {
            cout << "il numero inserito e' negativo" << endl;
        }
    }
    return 0;
}
```

23. Stampare il doppio del numero in input se è positivo, altrimenti il triplo se è negativo

```
/**  
ITIS-LS "Francesco Giordani" Caserta  
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni  
Data: 27/10/2020  
Numero progressivo dell'esercizio: es23  
Versione: 1.0  
Programmatore/i: Alessandro Esposito  
Sistema Operativo: Windows 10  
Compilatore/Interprete: code::blocks ver 20.03  
Obiettivo didattico: l'alunno è in grado di usare for e la selezione a due vie (if)  
Obiettivo del programma: acquisire 3 numeri da tastiera e stampare a video il doppio del numero se il  
numero è positivo altrimenti il triplo  
*/
```

```
#include <iostream>  
using namespace std;  
int main()  
{  
    int num;  

```

24. Stabilire se il numero in input è pari o dispari

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 27/10/2020
Numero progressivo dell'esercizio: es24
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: l'alunno è in grado di usare for e la selezione a due vie (if)
Obiettivo del programma: acquisire 3 numeri da tastiera e stampare a video se il numero è pari o dispari
*/

#include <iostream>
using namespace std;
int main()
{
    int num;
    for (int i=1; i<=3; i++)
    {
        cout << i << ")" <<"inserisci un numero intero: " ;
        cin >> num;
        if (num %2 ==0)
        {
            cout << "il numero inserito e' pari"<<endl;
        }
        else
        {
            cout << "il numero inserito e' dispari" << endl;
        }
    }
    return 0;
}
```

25. Stampare se il numero è pari o dispari e moltiplicarlo per 10 se è pari, mentre per 1 se è dispari

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 28/10/2020
Numero progressivo dell'esercizio: es25
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: l'alunno è in grado di usare for e la selezione a due vie (if)
Obiettivo del programma: acquisire 3 numeri da tastiera e stampare a video il numero moltiplicato per 10 se
il numero è pari e moltiplicato per 1 se è dispari
*/

#include <iostream>
using namespace std;
int main()
{
    int num, res, mol;
    for (int i=1; i<=3; i++)
    {
        cout << i << " " <<"inserisci un numero intero: " ;
        cin >> num;
res=num%2;
        if (res==0)
        {
            mol= num*10;
            cout << "il numero inserito e' pari"<<endl;
            cout <<"il numero moltiplicato per 10 e': " << mol << endl;
        }
        else
        {
            mol = num*1;
            cout << "il numero inserito e' dispari" << endl;
            cout << "il numero moltiplicato per 1 e': " << mol << endl;
        }
    }
    return 0;
}
```

26. Stampare un carattere

```
/**
```

```
ITIS-LS "Francesco Giordani" Caserta
```

```
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
```

```
Data: 30/10/2020
```

```
Numero progressivo dell'esercizio: es26
```

```
Versione: 1.0
```

```
Programmatore/i: Alessandro Esposito
```

```
Sistema Operativo: Windows 10
```

```
Compilatore/Interprete: code::blocks ver 20.03
```

```
Obiettivo didattico: l'alunno è in grado di stampare un carattere identificato con la variabile char
```

```
Obiettivo del programma: stampare un carattere salvato in una variabile di tipo char
```

```
*/
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    char carattere='3';
```

```
    cout << carattere << endl;
```

```
    char carattere1='a', carattere2='b';
```

```
    cout << carattere1<<" "<<carattere2<<endl;
```

```
    return 0;
```

```
}
```

27. Somma di due numeri

```
/**  
ITIS-LS "Francesco Giordani" Caserta  
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni  
Data: 03/10/2020  
Numero progressivo dell'esercizio: es27  
Versione: 1.0  
Programmatore/i: Alessandro Esposito  
Sistema Operativo: Windows 10  
Compilatore/Interprete: code::blocks ver 20.03  
Obiettivo didattico: conoscere la funzione int  
Obiettivo del programma: acquisire 2 numeri da tastiera e stampare a video la somma dei due numeri  
*/
```

```
#include <iostream>  
using namespace std;  
int main()  
{  
    int num1, num2;  

```

28. Somma di tre numeri

```
/**  
ITIS-LS "Francesco Giordani" Caserta  
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni  
Data: 03/10/2020  
Numero progressivo dell'esercizio: es28  
Versione: 1.0  
Programmatore/i: Alessandro Esposito  
Sistema Operativo: Windows 10  
Compilatore/Interprete: code::blocks ver 20.03  
Obiettivo didattico: conoscere la funzione int  
Obiettivo del programma: acquisire 3 numeri da tastiera e stampare a video la somma dei tre numeri  
*/
```

```
#include <iostream>  
using namespace std;  
int main()  
{  
    int num1, num2, num3;  
    cout << "inserisci il primo numero: " ;  
    cin >> num1;  
    cout << "inserisci il secondo numero: ";  
    cin >> num2;  
    cout << "inserisci il terzo numero: ";  
    cin >> num3;  
    cout << num1 + num2 + num3 << endl;  
    return 0;  
}
```


29. Menu con do while

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 06/10/2020
Numero progressivo dell'esercizio: es29
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: i prototipi
Obiettivo del programma: acquisire numeri da tastiera e stampare a video l'area o il perimetro a seconda del
operazione selezionata
*/
#include <iostream>

using namespace std;

int perimetro (int lato);
int area (int lato);
int lato;

int main()
{
    int lato=10, scelta;
    do
    {
        cout << "1) Calcolare il perimetro del quadrato" << endl;
        cout << "2) Calcolare l'area del quadrato" << endl;
        cout << "3) Exit" << endl;
        cout << "inserisci il numero corrispondente alla tua scelta: ";
        cin >> scelta;
        if (scelta==1)
        {
            cout << "il valore del perimetro e' " << perimetro (lato) << endl;
            cout << " " << endl;
        }
        if (scelta==2)
        {
            cout << "il valore del area e' " << area (lato) << endl;
            cout << " " << endl;
        }
    }
    while(scelta != 3);
    return 0;
}

int perimetro (int lato)
{
    return lato*4;
}
```

```
int area (int lato)
{
    return lato * lato;
}
```

30. Marte

Se nel alfabeto marziano ci sono 300 simboli, quanti bit servirebbero per codificarlo in ASCII?

Servirebbero 2^9 bit

31. Biologia Marina

Quanti bit servirebbero per codificare in ASCII le parole "biologia marina"?

Servirebbero 2^7 bit

32. Menu con Switch

/**

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 09/10/2020

Numero progressivo dell'esercizio: es32

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: code::blocks ver 20.03

Obiettivo didattico: ciclo switch

Obiettivo del programma: acquisire numeri da tastiera e stampare a video l'area o il perimetro a seconda dell'operazione selezionata

*/

```
#include <iostream>
```

```
using namespace std;
```

```
int perimetro (int lato);
```

```
int area (int lato);
```

```
int lato;
```

```
int main()
```

```
{
```

```
    int lato=10, scelta;
```

```
        cout << "1) Calcolare il perimetro del quadrato" << endl;
```

```
        cout << "2) Calcolare l'area del quadrato" << endl;
```

```
        cout << "3) Exit" << endl;
```

```
        cout << "inserisci il numero corrispondente alla tua scelta: ";
```

```
        cin >> scelta;
```

```
        switch(scelta)
```

```
        {
```

```
            case 1:
```

```
                cout << "il valore del perimetro e' " << perimetro (lato) << endl;
```

```
                cout << " " << endl;
```

```
                main();
```

case 2:

```
cout <<"il valore del area e' " << area (lato) << endl;
```

```
cout << " " << endl;
```

```
main();
```

case 3:

```
break;
```

```
}
```

```
return 0;
```

```
}
```

```
int perimetro (int lato)
```

```
{
```

```
return lato*4;
```

```
}
```

```
int area (int lato)
```

```
{
```

```
return lato * lato;
```

```
}
```

33. Menu Area

/**

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 13/10/2020

Numero progressivo dell'esercizio: es33

Versione: 1.0

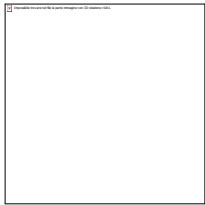
Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: code::blocks ver 20.03

Obiettivo didattico: ciclo switch, do while e prototipi

Obiettivo del programma: stampare a video l'area del quadrato, l'area del rettangolo o l'area del triangolo in base alla scelta dell'utente



*/

```
#include <iostream>
```

```
using namespace std;
```

```
int areaQuadrato(int latoPar);
```

```
int areaRettangolo(int basePar, int altezzaPar);
```

```
int areaTriangolo(int basePar, int altezzaPar);
```

```
int main()
```

```
{
```

```
    char scelta;
```

```
    int lato, base, altezza;
```

```
    cout<<"A) Area Quadrato" << endl;
```

```
    cout<<"B) Area Rettangolo" << endl;
```

```
    cout<<"C) Area Triangolo" << endl;
```

```
    cout<<"D) Esci" << endl;
```

```
    do
```

```
    {
```

```
        cout<<"inserisci la tua scelta: ";
```

```
        cin >> scelta;
```

```
        switch(scelta)
```

```
        {
```

```
            case 'A':
```

```
                cout << "inserisci il lato del quadrato: ";
```

```
                cin >> lato;
```

```
                cout << "area quadrato = " << areaQuadrato(lato) << endl;
```

```
                cout << " " << endl;
```

```
            break;
```

```
            case 'B':
```

```
                cout << "inserisci la base del rettangolo: ";
```

```
                cin >> base;
```

```
                cout << "inserisci l'altezza del rettangolo: ";
```

```
                cin >> altezza;
```

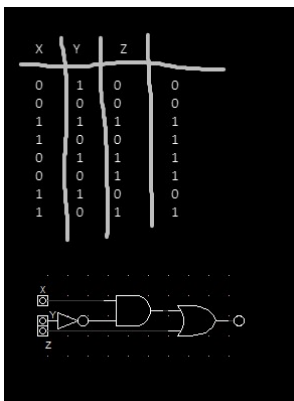
```
                cout << "area rettangolo = " << areaRettangolo(base, altezza) << endl;
```

```

        cout << " " << endl;
    break;
    case 'C':
        cout << "inserisci la base del triangolo: ";
        cin >> base;
        cout << "inserisci l'altezza del triangolo: ";
        cin >> altezza;
        cout << "area rettangolo = " << areaTriangolo(base, altezza) << endl;
        cout << " " << endl;
        break;
    }
}
while(scelta != 'D');
return 0;
}
int areaQuadrato(int latoPar)
{
    return latoPar*latoPar;
}
int areaRettangolo(int basePar, int altezzaPar)
{
    return basePar*altezzaPar;
}
int areaTriangolo(int basePar, int altezzaPar)
{
    return basePar*altezzaPar/2;
}
}

```

34. (X and Ynegato or Z)



35. AND OR NOT

AND

A	B	O
0	0	0
1	0	0
0	1	0
1	1	1

OR

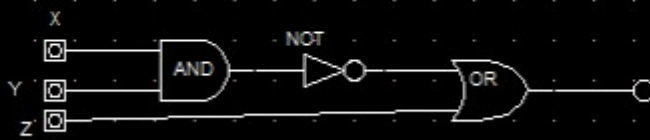
A	B	O
0	0	0
1	0	1
0	1	1
1	1	1

NOT

A	O
0	1
1	0

36. NOT(X and Y) or Z

X	Y	Z	
0	1	0	1
0	0	0	1
1	1	1	1
1	0	0	1
0	1	1	1
0	0	1	1
1	1	0	0
1	0	1	1



37. Stampare a video 6 numeri con ciclo indefinito

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 20/10/2020
Numero progressivo dell'esercizio: es37
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: ciclo indefinito
Obiettivo del programma: acquisire 6 numeri da tastiera e stamparli a video
*/
#include <iostream>
using namespace std;
int main()
{
    int num, i=1;
    cout << "quanti numeri vuoi inserire: " << endl;
    cin >> num;
    do
    {
        cout<<"inserire il "<<i<<"^ numero: "<< endl;
        cin>>num;
        i++;
    }
    while(i<num);
    return 0;
}
```

38. Stampare a video 6 numeri con ciclo definito

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 20/10/2020
Numero progressivo dell'esercizio: es38
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: ciclo definito
Obiettivo del programma: acquisire 6 numeri da tastiera e stamparli a video
*/
#include <iostream>
using namespace std;
int main()
{
    int num;
    for(int i=1;i<=6;i++)
    {
        cout<<"inserire il "<<i<<"^ valore: ";
        cin >> num;
        cout<<num<<endl;
    }
    return 0;
}
```

39. Tecnica della somma

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 23/10/2020
Numero progressivo dell'esercizio: es39
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: tecnica della somma
Obiettivo del programma: acquisire 5 numeri da tastiera e stampare a video la loro somma
*/
#include <iostream>
using namespace std;
int main()
{
    int somma=0;
    int num;
    for (int i=1;i<6;i++)
    {
        cout<<"inserire il "<<i<<"^ numero : ";
        cin>>num;
        somma=somma + num;
    }
    cout<<"la somma dei numeri e': "<<summa<<endl;
    return 0;
}
```

40. Tecnica della somma con ciclo indefinito

```
/**  
ITIS-LS "Francesco Giordani" Caserta  
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni  
Data: 23/10/2020  
Numero progressivo dell'esercizio: es40  
Versione: 1.0  
Programmatore/i: Alessandro Esposito  
Sistema Operativo: Windows 10  
Compilatore/Interprete: code::blocks ver 20.03  
Obiettivo didattico: tecnica della somma con ciclo indefinito  
Obiettivo del programma: acquisire 5 numeri da tastiera e stampare a video la loro somma  
*/  
#include <iostream>  
using namespace std;  
int main()  
{  
    int somma=0;  
    int num;  
    for (int i=1;i<6;i++)  
    {  
        cout<<"inserire il "<<i<<"^ numero : ";  
        cin>>num;  
        somma=somma + num;  
    }  
    cout<<"la somma dei numeri e': "<<summa<<endl;  
    return 0;  
}
```

41. Tecnica del contatore

/**

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni

Data: 26/10/2020

Numero progressivo dell'esercizio: es41

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: code::blocks ver 20.03

Obiettivo didattico: tecnica del contatore

Obiettivo del programma: acquisire numeri da tastiera e stampare a video la loro somma e il numero di numeri positivi inseriti

*/

```
#include<iostream>
```

```
Using namespace std;
```

```
Int main()
```

```
{
```

```
Int num, i=1;
```

```
Int somma=0, contaPositivi=0;
```

```
Do
```

```
{
```

```
Cout<<"inserire il "<<i<<"^ numero: ";
```

```
Cin>>num;
```

```
I++;
```

```
Somma=somma+num;
```

```
If(num>0)contaPositivi=contaPositivi+1;
```

```
}
```

```
While(i!=6);
```

```
Cout<<"la somma dei numeri inseriti e' "<<somma<<endl;
```

```
Cout<<"il numero dei positivi e' "<<contaPositivi;
```

```
Return 0;
```

```
}
```

42. Connettivi Logici

/**

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 26/10/2020

Numero progressivo dell'esercizio: es42

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: code::blocks ver 20.03

Obiettivo didattico: connettivi logici

Obiettivo del programma: acquisire numeri da tastiera e stampare a video la loro somma, interrompere il programma quando l'utente inserisce n o N

*/

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int num,i=1;
```

```
    int somma=0;
```

```
    Char scelta;
```

```
    do
```

```
    {
```

```
        cout<<"inserire il "<<i<<"^ numero: ";
```

```
        cin>>num,i++;
```

```
        somma=somma + num;
```

```
        Cout <<"vuoi continuare[s/n]: ";
```

```
        Cin >>scelta;
```

```
    }
```

```
    while (scelta!=n && scelta!=N);
```

```
    cout<<"la somma dei numeri inseriti e: ""<<somma<<endl;
```

```
}
```

43. Cicli for, do while, while

/*

Ciclo definito: Un ciclo che ripete le istruzioni per un numero definito di volte;

il programmatore conosce a priori il numero di ripetizioni;

c++ for (contatore= valore iniziale, contatore<= valore finale, incremento o decremento del contatore)

Ciclo indefinito precondizionale: un ciclo che ripete le istruzioni per un numero indefinito di volte;

Il programmatore non conosce quante volte le istruzioni si ripeteranno prima che la condizione diventi falsa;

c++ while(condizione){istruzioni da ripetere}

le istruzioni all'interno del ciclo vengono ripetute mentre la condizione è vera, il ciclo termina se la condizione

diventa falsa.

La condizione viene "verificata" prima di iniziare il ciclo, quindi le istruzioni all'interno del ciclo

potrebbero non essere eseguite neppure una volta.

Ciclo indefinito postcondizionale: un ciclo che ripete le istruzioni per un numero indefinito di volte;

Il programmatore non conosce quante volte le istruzioni si ripeteranno prima che la condizione diventi falsa;

c++ while(condizione){istruzioni da ripetere}

le istruzioni all'interno del ciclo vengono ripetute mentre la condizione è vera, il ciclo termina se la condizione

diventa falsa.

La condizione viene "verificata" dopo aver eseguito almeno una volta le istruzioni all'interno del ciclo.

Ciclo infinito: la condizione del while non si verifica mai.

*/

/**

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 27/10/2020

Numero progressivo dell'esercizio: es43

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: code::blocks ver 20.03

Obiettivo didattico: cicli for, do while e while

Obiettivo del programma: acquisire dei numeri da tastiera e stampare a video la loro somma, terminare il programma quando l'utente inserisce come input n o N

```
*/  
#include <iostream>  
using namespace std;  
int main()  
{  
    int const n=5;  
    int num, i,somma=0;  
    char risp;  
    cout<<"CONFRONTO DELLE STRUTTURE CICLICHE"<<endl;  
    cout<<"a) ciclo definito"<<endl;  
    cout<<"b) ciclo indefinito precondizionale"<<endl;  
    cout<<"c) ciclo indefinito postcondizionale"<<endl;  
    cout<<"d) ciclo indinito"<<endl;  
    cout<<"e) uscita"<<endl;  
    cout<<"inserisci la tua scelta";  
    cin>>risp;  
    switch(risp)  
    {  
        case 'a':  
            for(i=1;risp!='n' && risp!='N';i++)  
            {  
                cout<<"inserire il " << i <<"^numero:";  
                cin>>num;  
                somma=somma+num;  
                cout<<"vuoi continuare? [s/n]";  
                cin >> risp;  
            }  
        break;
```



```

case 'b':
    i=1;
    while(risp!=n)
    {
        cout<<"inserire il " << i <<"^numero:";
        cin>>num;
        somma=somma+num;
        i++;
        cout<<"vuoi continuare? [s/n]";
        cin >> risp;
    }
break;
case 'c':
    i=1;
    risp='s';
    do
    {
        cout<<"inserire il " << i <<"^numero:";
        cin>>num;
        somma=somma+num;
        i++;
        cout<<"vuoi continuare?[s/n] ";
        cin >>risp;
    }
    while(risp!='n');
break;
}
cout <<"la sommma dei numeri inseriti e: "<<somma<< endl;
return 0;
}

```

44. Procedura ordina due caratteri

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 17/12/2020
Numero progressivo dell'esercizio: es44
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: procedura ordina due caratteri
Obiettivo del programma: acquisire 2 caratteri e stamparli a video in ordine alfabetico
*/
#include <iostream>
using namespace std;
void ordina(char char1,char char2); //prototipo, firma o segnatura (manca il corpo)
int main()
{
    char c1,c2;
    cout<<"inserire il 1^ carattere: ";
    cin>> c1;
    cout<<"inserire il 2^ carattere: ";
    cin>> c2;
    ordina(c1, c2)
    return 0
}
void ordina(char char1,char char2)
{
    if (char1>char2)
        cout<<char2<<" "<< char 1;
    else
        cout<<char1<<" "<<char2;
}
```

45. Funzione somma con tre parametri

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 17/12/2020
Numero progressivo dell'esercizio: es45
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: funzione somma con tre parametri
Obiettivo del programma: acquisire 3 numeri da tastiera e stampare a video la loro somma
*/
#include <iostream>
using namespace std;
int somma(int num1,int num2,int num3); //prototipo, firma o segnatura (manca il corpo)
int main()
{
    int n1,n2,n3
    cout<<"inserire il 1^ numero: ";
    cin>> n1;
    cout<<"inserire il 2^ numero: ";
    cin>> n2;
    cout<<"inserire il 3^ numero: ";
    cin>> n3;
    cout<<"La somma dei tre numeri è uguale a: "<< somma(n1,n2,n3) << endl;
    return 0
}
int somma(int num1,num2,num3)
{
    return num1+num2+num3;
}
/*
int main()
{
    cout<<"La somma dei tre numeri è uguale a: "<< somma(10,4,5) << endl;
    return 0
}
}
}
```

46. numeri dispari compresi tra 200 e 1000

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 20/12/2020
Numero progressivo dell'esercizio: es46
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: ciclo for
Obiettivo del programma: stampare a video i numeri dispari maggiori di 200 e minori di 1000
*/
#include <iostream>
using namespace std;
int main()
{
    for(int i=201; i<=1000; i=i+2)
    {
        cout << i << endl;
    }
    return 0;
}
```

47. Area di Trapezio, rettangolo e triangolo

/**

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 21/12/2020

Numero progressivo dell'esercizio: es47

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: code::blocks ver 20.03

Obiettivo didattico: L'alunno è in grado di usare il top down, di descrivere l'algoritmo tramite pseudocodifica e flow-chart

Obiettivo del programma: scelta di ciascun sottoproblema usando un menu con do while e switch.

*/

```
#include <iostream>
```

```
using namespace std;
```

```
float areaTrap(int baseMagPar, int baseMinPar, int hPar); //prototipo, firma, segnatra. Con parametri formali
```

```
float areaRet(int basePar, int hPar);
```

```
float areaTri(int basePar, int hPar);
```

```
int main()
```

```
{
```

```
    int scelta, baseMag, baseMin, h, base;
```

```
    do
```

```
    {
```

```
        cout<<"1) Area Trapezio" << endl;
```

```
        cout<<"2) Area Rettangolo o Quadrato" << endl;
```

```
        cout<<"3) Area Triangolo" << endl;
```

```
        cout<<"4) Esci" << endl;
```

```
        cout<<"inserisci il numero della tua scelta: ";
```

```
        cin>>scelta;
```

```
        switch(scelta)
```

```
        {
```

```
            case 1:
```

```
                cout<<"inserisci base maggiore: ";
```

```
                cin >> baseMag; //primo parametro attuale
```

```
                cout<<"inserisci base minore: ";
```

```
                cin >> baseMin; //secondo parametro attuale
```

```
                cout<<"inserisci l'altezza: ";
```

```
                cin >> h; //terzo parametro attuale
```

```
                cout << "L'area del trapezio e' " << areaTrap(baseMag,baseMin,h) << endl;
```

```
                break;
```

```
            case 2:
```

```
                cout<<"inserisci base: ";
```

```
                cin >> base; //primo parametro attuale
```

```
                cout<<"inserisci l'altezza: ";
```

```
                cin >> h; //secondo parametro attuale
```

```
                cout << "L'area del rettangolo o del quadrato e' " << areaRet(base,h) << endl;
```

```
                break;
```

```
            case 3:
```

```
                cout<<"inserisci base: ";
```

```
                cin >> base; //primo parametro attuale
```

```
                cout<<"inserisci l'altezza: ";
```

```
    cin >> h; //secondo parametro attuale
    cout << "l'area del triangolo e' " << areaTri(base,h) << endl;
    break;
}
}
while(scelta != 4);
return 0;
}
float areaTrap(int baseMagPar, int baseMinPar, int hPar)
{
    int sommaBase;
    sommaBase = baseMagPar + baseMinPar;
    return sommaBase*hPar/2;
}
float areaRet(int basePar, int hPar)
{
    return basePar*hPar;
}
float areaTri(int basePar, int hPar)
{
    return basePar * hPar/2;
}
```

48. C++, scelta del sottoproblema utilizzando menu con switch e do while

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 07/01/2021
Numero progressivo dell'esercizio: es48
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: L'alunno è in grado di usare il top down, di descrivere l'algoritmo
tramite pseudocodifica e flow-chart
Obiettivo del programma: scelta di ciascun sottoproblema usando un menu con do while e switch.
*/
#include <iostream>
using namespace std;
int mediaAritm();
int magZero();
int somma();
int main()
{
    int scelta;
    do
    {
        cout << " MENU " << endl;
        cout << "0) Esci" << endl;
        cout << "1) Media Aritmetica" << endl;
        cout << "2) Numeri maggiori di Zero" << endl;
        cout << "3) Somma dei numeri compresi tra -8 e 8" << endl;
        cout << "inserisci il numero corrispondente alla tua scelta: ";
        cin >> scelta;
        switch(scelta)
        {
            case 1:
                mediaAritm();
                break;
            case 2:
                magZero();
                break;
            case 3:
                somma();
                break;

        }
    }
    while(scelta != 0);
    return 0;
}
int mediaAritm()
```

```

{
float somma = 0;
int num1;
int num2;
cout << "quanti numeri vuoi inserire: ";
cin >> num1;
for(int i=1; i<=num1; i++)
{
    cout << "inserisci il "<<i<<"^ numero: ";
    cin >> num2;
    somma = num2+somma;
}
cout << "la media aritmetica dei " << num1 << " numeri e' " << somma/num1<<endl;
cout << " " << endl;
return 0;
}
int magZero()
{
int num1;
int num2, contaMagZero;
cout << "quanti numeri vuoi inserire: ";
cin >> num1;
for(int i=1; i<=num1; i++)
{
    cout << "inserisci il "<<i<<"^ numero: ";
    cin >> num2;
    if(num2>0){
        contaMagZero+=1;
    }
}
cout <<"i numeri inseriti maggiori di zero sono "<<contaMagZero<<endl;
cout << " " <<endl;
return 0;
}
int somma()
{
int num1, num2;
int somma=0;
cout << "quanti numeri vuoi inserire: ";
cin >> num1;
for(int i=1; i<=num1; i++)
{
    cout << "inserisci il "<<i<<"^ numero: ";
    cin >> num2;
    if(num2> -8 && num2<8){
        somma = num2+somma;
    }
}
cout <<"la somma dei numeri inseriti compresi tra -8 ed 8 e' "<<somma<<endl;
cout << " " <<endl;
return 0;
}

```


49. Somma di 5 numeri

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 08/01/2021
Numero progressivo dell'esercizio: es49
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: L'alunno è in grado di usare la tecnica della somma e di descrivere l'algoritmo tramite
flow-chart
Obiettivo del programma: tecnica della somma, sommare 5 numeri.
*/
#include <iostream>
using namespace std;
int main()
{
    int somma, num, i=1;
    do
    {
        cout << "inserisci il " << i << "^ numero: ";
        cin >> num;
        somma = somma+num;
        i+=1;
    }
    while(i<=5);
    cout << "la somma e' " << somma << endl;
    return 0;
}
```

50. Cittadinanza, nuove tecnologie

Descrivi i punti di debolezza delle nuove tecnologie dal punto di vista dal cittadino

La privacy è completamente assente dato che i dispositivi ci ascoltano e osservano continuamente condividendo i dati con chiunque, la disinformazione dato che chiunque può scrivere tutto ciò che gli passa per la testa. Ma possiamo anche esporci a malware, truffe o furti dato che il web non è controllato da nessuno specialmente gli strati inferiori. Le menti deboli possono essere influenzate con propaganda e controllate.

Scrivi cosa si può fare per superarli o ridurre le conseguenze negative

Per quanto riguarda l'assenza privacy non è una cosa risolvibile, ormai siamo andati troppo oltre e perfino chi non ha dispositivi viene osservato ma per limitarlo basterebbe usare la tecnologia solo per il minimo necessario. Per la disinformazione lo stato dovrebbe eliminare i profili o siti che la causano e multare/arrestare gli Host/proprietari. Per le truffe/furti ecc. il cittadino dovrebbe evitare di fidarsi troppo e controllare ogni cosa due volte, così risolverebbe anche l'ultimo problema.

51.C++, dati n caratteri stampare quante volte viene inserito il carattere z o n

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 10/01/2021
Numero progressivo dell'esercizio: es51
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: L'alunno è in grado di eseguire il ciclo indefinito, di usare la selezione a una via e di
descrivere l'algoritmo tramite flow-chart e pseudocodifica
Obiettivo del programma: dati N caratteri stampare a video quante volte viene inserito il carattere Z oppure
il carattere R
*/
#include <iostream>
using namespace std;
int main()
{
    int ContaCar=0;
    char car;
    do
    {
        cout << "inserisci carattere: ";
        cin >> car;
        if(car=='Z' || car=='R')
        {
            ContaCar = ContaCar+1;
        }
    }
    while(car != "");
    cout << "Hai inserito Z o R circa " << ContaCar << " volte" << endl;
    return 0;
}
```

52.C++, creare un vettore, inserire 5 caratteri e mostrarli a video

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 14/01/2021
Numero progressivo dell'esercizio: es52
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: L'alunno è in grado di creare un array per inserire caratteri
Obiettivo del programma: creare un array vettoriale e inserire 5 caratteri in quel array con un ciclo for e
mostrarli a video con un altro ciclo
*/
#include<iostream>
using namespace std;
char vet[10];
int main()
{
    for(int i=0; i<5; i++)
    {
        cout << "inserisci il " << i+1 << "^ carattere: ";
        cin >> vet[i];
    }
    cout << endl;
    for(int i=0; i<5; i++)
    {
        cout << vet[i] << " ";
    }
    return 0;
}
```

53.C++, creare un vettore, inserire 9 caratteri e mostrare a video le a inserite

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 15/01/2021
Numero progressivo dell'esercizio: es53
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: L'alunno è in grado di creare un array vettoriale per inserire caratteri ed eseguire
operazioni su quei caratteri
Obiettivo del programma: creare un array vettoriale e inserire 9 caratteri e mostrare a video quante sono le
'a' inserite
*/
#include<iostream>
using namespace std;
char vet[10];
int main()
{
    int ContaA=0;
    for(int i=0; i<9; i++)
    {
        cout << "inserisci il " << i+1 << "^ carattere: ";
        cin >> vet[i];
    }
    cout << endl;
    for(int i=0; i<9; i++)
    {
        if (vet[i]=='a')
        {
            ContaA += 1;
        }
    }
    cout << "hai inserito la lettera a circa " << ContaA << " volte";
    cout << " " << endl;
    return 0;
}
```

54.C++, creare un vettore, acquisire la dimensione logica compresa tra 0 e la dimensione fisica

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 17/01/2021
Numero progressivo dell'esercizio: es54
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: L'alunno è in grado di creare un vettore per inserire numeri ed eseguire operazioni su quei numeri
Obiettivo del programma: caricare un vettore e stamparlo. Scrivere la funzione OttieniDim per acquisire la dimensione logica compresa tra 0 e la dimensione fisica
*/
#include <iostream>
using namespace std;
int dimlog, vet[10];
int OttieniDim();
void StampaVet();
void CaricaVet();
int main()
{
    dimlog=OttieniDim();
    CaricaVet();
    StampaVet();
    return 0;
}
int OttieniDim()
{
    do
    {
        cout<<"inserisci la dimensione max del vettore: ";
        cin>>dimlog;
    }
    while(dimlog<0 || dimlog>10);
    return dimlog;
}
void CaricaVet()
{
    for(int i=0; i<dimlog; i++)
    {
        cout<<"inserisci un numero: ";
        cin>>vet[i];
    }
}
void StampaVet()
{
    for(int i=0; i<10; i++)
```

```

{
    cout<<vet[i]<<" ";
}
cout <<" "<<endl;
}

```

55.C++, creare un vettore, acquisire la dimensione logica con il passaggio di parametri

```

/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 17/01/2021
Numero progressivo dell'esercizio: es55
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: L'alunno è in grado di creare un vettore per inserire numeri ed eseguire operazioni su
quei numeri
Obiettivo del programma:caricare un vettore e stamparlo. Scrivere la funzione OttieniDim per aquisire la
dimensione logica compresa tra 0 e la dimensione fisica, usare il passaggio di parametri
*/
#include <iostream>
using namespace std;
const int dimfis = 10;
int dimlog, vet[dimfis];
int OttieniDim(int dimspar);
void StampaVet(int vetpar[], int dimlogpar);
void CaricaVet(int vetpar[], int dimlogpar);
int main()
{
    dimlog=OttieniDim(dimfis);
    CaricaVet(vet, dimlog);
    StampaVet(vet, dimlog);
    return 0;
}
int OttieniDim(int dimspar)
{
    do
    {
        cout<<"inserisci la dimensione max del vettore: ";
        cin>>dimlog;
    }
    while(dimlog<0 || dimlog>10);
    return dimlog;
}
void CaricaVet(int vetpar[], int dimlogpar)
{

```

```
for(int i=0; i<dimlog; i++)
{
    cout<<"inserisci un numero: ";
    cin>>vet[i];
}
}
void StampaVet(int vetpar[], int dimlogpar)
{
    for(int i=0; i<10; i++)
    {
        cout<<vet[i]<<" ";
    }
    cout <<" "<<endl;
}
```

56.C++, calcolare il perimetro di un esagono

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 21/01/2021
Numero progressivo dell'esercizio: es56
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico:
Obiettivo del programma:
*/
/* Calcolare il perimetro di un esagono*/
#include <iostream>
using namespace std;
int lato;
int const numlati=8;
int perimetroEsagono();
int main()
{
    int perimetro = perimetroEsagono();
    cout <<"il perimetro dell' esagono e' "<< perimetro<< endl;
    return 0;
}
int perimetroEsagono ()
{
    cout<<" inserire il valore del lato"<<endl;
    cin>>lato;
    return lato*6; }
/*ver1.0
#include <iostream>
using namespace std;
int lato;
int main()
{
    cout << 12*6 << endl;
    return 0;
}
ver1.1
#include <iostream>
using namespace std;
int lato;
int main()
{
    lato=12;//assegnazione del valore della variabile da programma
    cout << lato*6 << endl;
    return 0;
}
ver1.2
#include <iostream>
```



```
using namespace std;
int lato;
int main()
{
    cout<<" inserire il valore del lato"<<endl;
    cin>>lato;//assegnazione del valore della variabile da tastiera
    cout << lato*6 << endl;
    return 0;
}
```

ver2.0

Calcolare il perimetro di un poligono

```
#include <iostream>
```

```
using namespace std;
```

```
int lato;
```

```
int const numlati=8;
```

```
int main()
```

```
{
```

```
    cout<<" inserire il valore del lato"<<endl;
```

```
    cin>>lato;//assegnazione del valore della variabile da tastiera
```

```
    cout <<"il perimetro del poligono che ha "<<numlati<<" lati e' "<< lato*numlati<< endl;
```

```
    return 0;
```

```
}*/
```

57.C++, calcolare il perimetro di un esagono eliminando l'interfaccia utente

/**

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 22/01/2021

Numero progressivo dell'esercizio: es57

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: code::blocks ver 20.03

Obiettivo didattico: uso di una funzione

Obiettivo del programma: Scrivere a video il perimetro di un esagono utilizzando la funzione perimetroEsagono (eliminando l'interfaccia con l'utente interna alla funzione che è un errore frequente dei principianti).

*/ #include <iostream>

using namespace std;

int lato,perimetro;

int perimetroEsagono();

int main()

{

cout<<" inserire il valore del lato: ";

cin>>lato; perimetro=perimetroEsagono();

cout <<"il perimetro dell' esagono e' "<< perimetro<< endl;

return 0;

}

int perimetroEsagono ()

{

return lato*6;

}

58.C++, calcolare il perimetro di un esagono usando una funzione parametrizzata

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 22/01/2021
Numero progressivo dell'esercizio: es58
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: uso di una funzione
Obiettivo del programma: Scrivere a video il perimetro di un esagono utilizzando la
funzione perimetroEsagono parametrizzata.
*/
#include <iostream>
using namespace std;
int lato;
int perimetroEsagono(int latoPar);
int main()
{
    cout<<" inserire il valore del lato: ";
    cin>>lato;
    cout <<"il perimetro dell' esagono e' "<< perimetroEsagono(lato)<< endl;
    return 0;
}
int perimetroEsagono (int latoPar)
{
    return latoPar*6;
}
```

59.C++, calcolare il perimetro di un poligono

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 22/01/2021
Numero progressivo dell'esercizio: es59
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: uso di una funzione
Obiettivo del programma: Scrivere a video il perimetro di un poligono utilizzando la
funzione perimetroPoligono parametrizzata
*/
#include <iostream>
using namespace std;
int lato,numLati;
int perimetroPoligono(int latoPar, int numLatiPar);
int main()
{
    cout<<" inserire il numero dei lati del poligono: ";
    cin>>numLati; cout<<" inserire il valore del lato: ";
    cin>>lato; cout <<"il perimetro del poligono e' "<< perimetroPoligono(lato,numLati)<< endl;
    return 0;
}
int perimetroPoligono (int latoPar, int numLatiPar)
{
    return latoPar*numLatiPar;
}
```

60.C++, dati n caratteri stampare a video quante volte viene inserito il carattere a

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 29/01/2021
Numero progressivo dell'esercizio: es60
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: uso degli array a una dimensione detti vettori
Obiettivo del programma: leggere un numero n di caratteri e stampare a video il numero di 'a' inserite
*/
#include <iostream>
using namespace std;
char vet[100];
int i=-1;
int contaA=0;
int main()
{
    do
    {
        i++;
        cout<<"inserisci un carattere: ";
        cin>>vet[i];
    }
    while(vet[i]!='*');
    int dimLog=i;
    for(int i=0; i<dimLog; i++)
    {
        if(vet[i]=='a')
        {
            contaA++;
        }
    }
    cout << "Hai scritto "<<contaA<<" volte il carattere a" << endl;
    return 0;
}
```

61. C++, Somma degli elementi di un vettore

```
/**  
ITIS-LS "Francesco Giordani" Caserta  
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni  
Data: 31/01/2021  
Numero progressivo dell'esercizio: es61  
Versione: 1.0  
Programmatore/i: Alessandro Esposito  
Sistema Operativo: Windows 10  
Compilatore/Interprete: code::blocks ver 20.03  
Obiettivo didattico: uso dei vettori  
Obiettivo del programma: caricare un vettore di interi nella dichiarazione e sommare i suoi elementi  
*/  
#include <iostream>  
using namespace std;  
int vetNum[100]={1,150,25,7};  
int num, somma;  

```

62. C++, caricare un vettore di interi e sommare i suoi elementi

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 31/01/2021
Numero progressivo dell'esercizio: es62
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: uso dei vettori, tecnica della somma
Obiettivo del programma: caricare un vettore di interi e sommare i suoi elementi
*/
#include <iostream>
using namespace std;
int vetNum[100];
int num, somma;
int main()
{
    for(int i=0;i<4;i++)
    {
        cout<<"inserire un numero: ";
        cin>>vetNum[i];
    }
    for(int i=0;i<4;i++)
    {
        somma=somma+vetNum[i];
    }
    cout <<somma<<endl;
    return 0;
}
```

63. C++, caricare un vettore e contare quanti dei suoi elementi sono maggiori o uguali a 10

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 02/01/2021
Numero progressivo dell'esercizio: es63
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: uso dei vettori, tecnica del contatore
Obiettivo del programma: caricare un vettore di interi e contare quanti dei suoi elementi sono maggiori o
uguali a 10
*/
#include <iostream>
using namespace std;
int vetNum[100];
int somma=0, contatore=0;
int main()
{
    for(int i=0;i<4;i++)
    {
        cout<<"inserire un numero: ";
        cin>>vetNum[i];
    }
    for(int i=0;i<4;i++)
    {
        if(vetNum[i]>=10) contatore++;
    }
    cout << "i numeri maggiori di 10 sono " << contatore << endl;
    return 0;
}
```


64. C++, caricare un vettore di 5 numeri interi e sommare i negativi

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 06/01/2021
Numero progressivo dell'esercizio: es64
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: l'alunno è in grado di utilizzare i vettori e la tecnica della somma
Obiettivo del programma: caricare un vettore di 5 numeri interi e sommare i negativi
*/
#include <iostream>
using namespace std;
int vetNum[200];
int somma=0;
int main()
{
    for(int i=0;i<5;i++)
    {
        cout<<"inserire un numero: ";
        cin>>vetNum[i];
    }
    for(int i=0;i<5;i++)
    {
        if(vetNum[i]<=0) somma = vetNum[i]+somma;
    }
    cout << "la somma dei numeri negativi inseriti e' " << somma << endl;
    return 0;
}
```

65. C++, caricare un vettore di interi finchè non viene inserito 0, poi sommare i suoi elementi

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 06/01/2021
Numero progressivo dell'esercizio: es65
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: l'alunno è in grado di utilizzare i vettori e la tecnica della somma
Obiettivo del programma: caricare un vettore di interi finchè l'utente inserisce 0 e sommare i negativi
*/
#include <iostream>
using namespace std;
int vetNum[200];
int i=-1, somma=0;
int main()
{
    do
    {
        i++;
        cout<<"inserire un numero (0 per finire): ";
        cin>>vetNum[i];
    }while(vetNum[i]!=0);
    int dimLog=i;
    for(int i=0;i<dimLog;i++)
    {
        if(vetNum[i]<=0) somma = vetNum[i]+somma;
    }
    cout << "la somma dei numeri negativi inseriti e' " << somma << endl;
    return 0;
}
```

66. C++, dato un vettore assegnare al terzo posto un valore inserito da tastiera

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 06/01/2021
Numero progressivo dell'esercizio: es66
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: l'alunno è in grado di utilizzare i vettori
Obiettivo del programma: Dato un vettore assegnare al terzo
elemento un valore inserito da tastiera
*/
#include <iostream>
using namespace std;
char vetCar[75];
char car, risp;
int i=-1;
int main()
{
    do
    {
        i++;
        cout<<"inserire un carattere: ";
        cin>>vetCar[i];
        cout<<"vuoi continuare(s/n): ";
        cin>>risp;
    }while(risp == 's' || risp=='S');
    int dimLog=i;
    for(int i=0; i<=dimLog; i++)
    {
        cout<<vetCar[i]<<" ";
    }
    cout<<endl;
    cout << "inserisci il carattere da inserire nella terza posizione: ";
    cin>>car;
    vetCar[2]=car;
    for(int i=0; i<=dimLog; i++)
    {
        cout<<vetCar[i]<<" ";
    }
    cout<<endl;
    return 0;
}
```

67. C++, caricare e leggere un vettore con valori interi casuali compresi tra 0 e 300

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 08/01/2021
Numero progressivo dell'esercizio: es67
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: l'alunno è in grado di utilizzare i vettori e la generazione di numeri casuali
Obiettivo del programma: Caricare e leggere un vettore con valori interi casuali compresi tra 0 e 300
*/
#include <iostream>
#include <time.h>
#include <stdlib.h>
using namespace std;
int vetNum[10];
int numRand;
int main()
{
    srand(time(NULL));
    for (int i=0;i<10;i++)
    {
        vetNum[i]=rand()%301;
    }
    cout << "I numeri casuali generati sono: " << endl;
    for(int i=0;i<10;i++)
    {
        cout<<vetNum[i]<<" - ";
    }
    return 0;
}
```

68. C++, caricare un vettore con valori interi casuali compresi tra 0 e 300 e raddoppiare i valori

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 08/01/2021
Numero progressivo dell'esercizio: es68
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: l'alunno è in grado di utilizzare i vettori e la generazione di numeri casuali
Obiettivo del programma: Caricare e leggere un vettore con valori interi casuali compresi tra 0 e 300,
raddoppiare il valore del vettore caricato in modo casuale
*/
#include <iostream>
#include <time.h>
#include <stdlib.h>
using namespace std;
int vetNum[10];
int numRand;
int main()
{
    srand(time(NULL));
    for (int i=0;i<10;i++)
    {
        vetNum[i]=rand()%301;
    }
    cout << "I numeri casuali generati sono: " << endl;
    for(int i=0;i<10;i++)
    {
        cout<<vetNum[i]<<" - ";
    }
    cout<<endl;
    for (int i=0;i<10;i++)
    {
        vetNum[i]=vetNum[i]*2;
    }
    cout << "I numeri casuali generati raddoppiati sono: " << endl;
    for(int i=0;i<10;i++)
    {
        cout<<vetNum[i]<<" - ";
    }
    cout<<endl;
    return 0;
}
```

69. C++, caricare un vettore con interi casuali tra 0 e 300 e raddoppiare i compresi tra 10 e 100

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 08/01/2021
Numero progressivo dell'esercizio: es69
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: l'alunno è in grado di utilizzare i vettori e la generazione di numeri casuali
Obiettivo del programma: Caricare e leggere un vettore con valori interi casuali compresi tra 0 e 300,
raddoppiare solo i numeri casuali compresi tra 10 e 100
*/
#include <iostream>
#include <time.h>
#include <stdlib.h>
using namespace std;
int vetNum[10];
int numRand;
int main()
{
    srand(time(NULL));
    for (int i=0;i<10;i++)
    {
        vetNum[i]=rand()%301;
    }
    cout << "I numeri casuali generati sono: " << endl;
    for(int i=0;i<10;i++)
    {
        cout<<vetNum[i]<<" - ";
    }
    cout<<endl;
    for (int i=0;i<10;i++)
    {
        If(vetNum[i]>10 && vetNum[i]<100) vetNum[i]=vetNum[i]*2;
    }
    cout << "I numeri casuali generati raddoppiati sono: " << endl;
    for(int i=0;i<10;i++)
    {
        cout<<vetNum[i]<<" - ";
    }
    cout<<endl;
    return 0;
}
```

70. C++, Caricare un vettore di caratteri utilizzando le procedure caricaVet e stampaVet. Contare le "F" di un vettore utilizzando la funzione contaF

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 12/01/2021
Numero progressivo dell'esercizio: es70
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: l'alunno
Obiettivo del programma: Caricare e leggere un vettore di caratteri utilizzando le
procedure caricaVet e stampaVet. Contare le "F" di un vettore utilizzando la funzione contaF.
*/
#include <iostream>
using namespace std;
void caricaVet();
void stampaVet();
int contaF();
char vetCar[150];
int main()
{
    caricaVet();
    stampaVet();
    cout << endl;
    cout << contaF() << endl;
    return 0;
}
void caricaVet()
{
    for(int i=0; i<3; i++)
    {
        cout << "inserisci un carattere: ";
        cin >> vetCar[i];
    }
}
void stampaVet()
{
    for(int i=0; i<3; i++)
    {
        cout << vetCar[i] << " ";
    }
}
int contaF()
{
    int contatore=0;
    for(int i=0; i<3; i++)
    {
        if(vetCar[i]=='F' || vetCar[i]=='f')
```

```
{  
    contatore++;  
}  
}  
return contatore;  
}
```


71. C++, Scrivere la procedura che stampa le posizioni del carattere F nel vettore di caratteri

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 12/01/2021
Numero progressivo dell'esercizio: es71
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: l'alunno
Obiettivo del programma: Caricare e leggere un vettore di caratteri utilizzando le
procedure caricaVet e stampaVet. Contare le "F" di un vettore utilizzando la funzione contaF. Scrivere la
procedura che stampa le posizioni del carattere F nel vettore di caratteri
*/
#include <iostream>
using namespace std;
void caricaVet();
void stampaVet();
int contaF();
void PosizioniF();
char vetCar[150];
int main()
{
    caricaVet();
    stampaVet();
    cout << endl;
    cout << "Hai inserito " << contaF() << " volte il carattere F" << endl;
    cout << "Hai inserito il carattere F nelle posizioni: " << endl;
    PosizioniF();
    return 0;
}
void caricaVet()
{
    for(int i=0; i<3; i++)
    {
        cout << "inserisci un carattere: ";
        cin >> vetCar[i];
    }
}
void stampaVet()
{
    for(int i=0; i<3; i++)
    {
        cout << vetCar[i] << " ";
    }
}
int contaF()
{
    int contatore=0;
```

```
for(int i=0; i<3; i++)
{
    if(vetCar[i]=='F' || vetCar[i]=='f')
    {
        contatore++;
    }
}
return contatore;
}
void PosizioniF()
{
    for(int i=0; i<3; i++)
    {
        if(vetCar[i]=='F' || vetCar[i]=='f')
        {
            cout << "Posizione: " << i << endl;
        }
    }
}
```

73. C++, Contare le F in un vettore e stampare le posizioni con i parametri

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 19/01/2021
Numero progressivo dell'esercizio: es73
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: operazioni sugli elementi del vettore
Obiettivo del programma: Caricare e leggere un vettore di caratteri utilizzando le procedure.
Contare le "F" di un vettore utilizzando una funzione. Scrivere la
procedura che stampa le posizioni del carattere F nel vettore di caratteri
*/
#include <iostream>
using namespace std;
//dati
char vetCar[150];
char vetCar2[50];
int contatore;
int cont=10;
//prototipi o signature o firme
void caricaVet(char vetPar[], int dimLogPar);
void stampaVet(char vetPar[], int dimLogPar);
int contaF(char vetPar[], int dimLogPar);
void PosizioniF(char vetPar[], int dimLogPar);
int main()
{
    cout<<cont<<endl;
    caricaVet(vetCar,4);
    stampaVet(vetCar, 4);
    contatore = contaF();
    cout << "Hai inserito " << contatore << " volte il carattere F" << endl;
    cout <<"Hai inserito il carattere F nelle posizioni: ";
    PosizioniF(char vetPar[], int dimLogPar);
    cout << endl;
    caricaVet(vetCar2, 5);
    stampaVet(vetCar2, 5);
    cout << endl;

    cout<<endl;
    cout<<cont;
    return 0;
}
void caricaVet(char vetPar[], int dimLogPar)
{
    for(int i=0; i<dimLogPar; i++)
    {
        cout << "inserisci un carattere: ";
```

```
        cin >> vetPar[i];
    }
}
void stampaVet(char vetPar[], int dimLogPar)
{
    for(int i=0; i<dimLogPar; i++)
    {
        cout << vetPar[i] << " - ";
    }
}
int contaF(char vetPar[], int dimLogPar)
{
    int cont=0;
    for(int i=0; i<dimLogPar; i++)
    {
        if(vetPar[i]=='f')
        {
            cont++;
        }
    }
    return cont;
}
void PosizioniF(char vetPar[], int dimLogPar)
{
    for(int i=0; i<dimLogPar; i++)
    {
        if(vetPar[i]=='f')
        {
            cout << i << " - ";
        }
    }
}
}
```

74. C++, Data la lunghezza e il numero di lati calcolare il perimetro

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 22/01/2021
Numero progressivo dell'esercizio: es74
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: operazioni sugli elementi del vettore
Obiettivo del programma: Dato in input il lato di un poligono calcolare e visualizzare il suo perimetro
*/
#include <iostream>
using namespace std;
int lungLato,numLati;
int perimetro(int lungLatoPar,int numLatiPar);
int main()
{
    //Perimetro=perimetro(lungLato,numLati)
    cout<<"inserire la lunghezza del lato: ";
    cin>>lungLato;
    cout<<"inserire il numero dei lati: ";
    cin>>numLati;
    cout<<"perimetro= "<<perimetro(lungLato,10)<<endl;
    return 0;
}
int perimetro(int lungLatoPar,int numLatiPar)
{
    return lungLatoPar*numLatiPar;
}
```

75. C++, Scrivere la procedura per scambiare il valore di due variabili per indirizzo

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 23/01/2021
Numero progressivo dell'esercizio: es75
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: passaggio di parametri per valore e per indirizzo
Obiettivo del programma: Scrivere la procedura per scambiare il valore di due variabili (per indirizzo)
*/
#include<iostream>
using namespace std;
int n1, n2;
void scambia(int &n1Par, &n2Par);
int main()
{
n1=10;
n2=20;
scambia(n1, n2);
cout<<endl;
cout<<"valori delle variabili n1 e n2: ";
cout << n1 << " " << n2 << endl;
return 0;
}
void scambia(int &n1Par, &n2Par)
{
int app;
app=n1Par;
n1Par=n2Par;
n2Par=app;
cout << "parametri n1Par e n2Par: ";
cout << n1Par << " " << n2Par;
}
```

76. C++, Dato un vettore di interi scambiare il primo elemento con l'ultimo

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 25/02/2021
Numero progressivo dell'esercizio: es76
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: passaggio di parametri per valore
Obiettivo del programma: Dato un vettore di interi scambiare il primo elemento con l'ultimo
*/
#include<iostream>
using namespace std;
int vetNum[100];
int dimLog = 5;
void caricaVet(int vetNumPar[], int dimLogPar);
void stampaVet(int vetNumPar[], int dimLogPar);
void scambia(int vetNumPar[], int dimLogPar);
int main()
{
caricaVet(vetNum, dimLog);
stampaVet(vetNum, dimLog);
cout<<endl;
scambia(vetNum, dimLog);
stampaVet(vetNum, dimLog);
cout<<endl;
return 0;
}
void caricaVet(int vetNumPar[], int dimLogPar)
{
for(int i=0; i<dimLog; i++)
{
cout << "inserisci l'elemento numero " << i << " del vettore: ";
cin>> vetNumPar[i];
}
}
void stampaVet(int vetNumPar[], int dimLogPar)
{
cout << "gli elementi del vettore di interi sono: "
for(int i=0; i<dimLog; i++)
{
cout << vetNumPar[i] << " - ";
}
}
void scambia(int vetNumPar[], int dimLogPar)
{
int app;
app=vetNumPar[0];
```

```
vetNum[0]=vetNumPar[dimLogPar-1];  
vetNum[dimLogPar-1]=app;  
}
```


77. C++, Caricare 3 vettori di interi e stamparli a video

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 26/02/2021
Numero progressivo dell'esercizio: es77
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo del programma: caricare 3 vettori di interi e stamparli a video
*/
#include <iostream>
using namespace std;
int vetNum1[100];
int vetNum2[100];
int vetNum3[100];
void caricaVet(int vetNumPar[]);
void stampaVet(int vetNumPar[]);
int main()
{
    cout << "-Primo Vettore-" << endl;
    caricaVet(int vetNum1);
    cout << "-Secondo Vettore-" << endl;
    caricaVet(int vetNum2);
    cout << "-Terzo Vettore-" << endl;
    caricaVet(int vetNum3);
    cout << endl;
    cout << "-Primo Vettore-" << endl;
    stampaVet(int vetNum1);
    cout << "-Secondo Vettore-" << endl;
    stampaVet(int vetNum2);
    cout << "-Terzo Vettore-" << endl;
    stampaVet(int vetNum3);
    return 0;
}
void caricaVet(int vetNumPar[])
{
    int dimLog;
    cout << "Quanti elementi vuoi inserire? " << endl;
    cin >> dimLog;
    for(int i=0; i<dimLog; i++)
    {
        cout << "inserire l'elemento numero " << i << " del vettore: ";
        cin >> vetNumPar[i]
    }
}
void stampaVet(int vetNumPar[]);
{
    cout << "gli elementi del vettore di interi sono: "
    for(int i=0; i<dimLog; i++)
    {
```

```
    cout << vetNumPar[i] << " - ";  
  }  
}
```

78. C++, Copiare gli elementi di un vettore in un altro vettore

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 28/02/2021
Numero progressivo dell'esercizio: es78
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo del programma: copiare gli elementi di un vettore in un altro vettore
*/
#include <iostream>
using namespace std;
int vet1[100];
int vet2[100];
int dimLog;
void caricaVet(int vetNumPar[], int dimLogPar);
void stampaVet(int vetNumPar[], int dimLogPar);
void copiaVet(int vetNumPar[], int vetCopia[], int dimLogPar);
int main()
{
    cout << "Quanti valori vuoi inserire? ";
    cin >> dimLog;
    caricaVet(vet1, dimLog);
    cout << endl;
    cout << "Stampa Primo Vettore" << endl;
    stampaVet(vet1, dimLog);
    copiaVet(vet1, vet2, dimLog);
    cout << endl;
    cout << "Stampa Secondo Vettore" << endl;
    stampaVet(vet2, dimLog);
    return 0;
}
void caricaVet(int vetNumPar[], int dimLogPar)
{
    for(int i=0; i<dimLog; i++)
    {
        cout << "inserire l'elemento numero " << i << " del vettore: ";
        cin >> vetNumPar[i];
    }
}
void stampaVet(int vetNumPar[], int dimLogPar)
{
    for(int i=0; i<dimLog; i++)
    {
        cout << vetNumPar[i] << " - ";
    }
}
void copiaVet(int vetNumPar[], int vetCopia[], int dimLogPar)
{
    for(int i=0; i<dimLog; i++)
```

```
{  
    vetCopia[i] = vetNumPar[i];  
}  
}
```

79. C++, Raddoppiare gli elementi di un vettore e memorizzarli in un altro vettore

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 28/02/2021
Numero progressivo dell'esercizio: es79
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo del programma: raddoppiare gli elementi di un vettore e memorizzarli in un altro vettore
*/
#include <iostream>
using namespace std;
int vet1[100];
int vet2[100];
int dimLog;
void caricaVet(int vetNumPar[], int dimLogPar);
void stampaVet(int vetNumPar[], int dimLogPar);
void copiaVet(int vetNumPar[], int vetCopia[], int dimLogPar);
int main()
{
    cout << "Quanti valori vuoi inserire? ";
    cin >> dimLog;
    caricaVet(vet1, dimLog);
    cout << endl;
    cout << "Stampa Primo Vettore" << endl;
    stampaVet(vet1, dimLog);
    copiaVet(vet1, vet2, dimLog);
    cout << endl;
    cout << "Stampa Secondo Vettore" << endl;
    stampaVet(vet2, dimLog);
    return 0;
}
void caricaVet(int vetNumPar[], int dimLogPar)
{
    for(int i=0; i<dimLog; i++)
    {
        cout << "inserire l'elemento numero " << i << " del vettore: ";
        cin >> vetNumPar[i];
    }
}
void stampaVet(int vetNumPar[], int dimLogPar)
{
    for(int i=0; i<dimLog; i++)
    {
        cout << vetNumPar[i] << " - ";
    }
}
void copiaVet(int vetNumPar[], int vetCopia[], int dimLogPar)
```

```
{  
  for(int i=0; i<dimLog; i++)  
  {  
    vetCopia[i] = vetNumPar[i]*2;  
  }  
}
```

80. C++, Sommare due vettori

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 28/02/2021
Numero progressivo dell'esercizio: es80
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo del programma: Sommare due vettori
*/
#include <iostream>
using namespace std;
int vet1[100];
int vet2[100];
int vetSomma[100];
int dimLog;
void caricaVet(int vetNumPar[], int dimLogPar);
void stampaVet(int vetNumPar[], int dimLogPar);
void sommaVet(int vetNumPar1[], int vetNumPar[], int vetSommaPar[], int dimLogPar);
int main()
{
    cout << "Quanti valori vuoi inserire? ";
    cin >> dimLog;
    caricaVet(vet1, dimLog);
    cout << endl;
    caricaVet(vet2, dimLog);
    cout << "Stampa Primo Vettore" << endl;
    stampaVet(vet1, dimLog);
    cout << endl;
    cout << "Stampa Secondo Vettore" << endl;
    stampaVet(vet2, dimLog);
    cout << endl;
    sommaVet(vet1, vet2, vetSomma, dimLog);
    cout << "Somma dei due vettori" << endl;
    stampaVet(vetSomma, dimLog);
    return 0;
}
void caricaVet(int vetNumPar[], int dimLogPar)
{
    for(int i=0; i<dimLog; i++)
    {
        cout << "inserire l'elemento numero " << i << " del vettore: ";
        cin >> vetNumPar[i];
    }
}
void stampaVet(int vetNumPar[], int dimLogPar)
{
    for(int i=0; i<dimLog; i++)
    {
        cout << vetNumPar[i] << " - ";
    }
}
```

```
    }  
  }  
  void sommaVet(int vetNumPar1[], int vetNumPar2[], int vetSommaPar[], int dimLogPar)  
  {  
    int somma;  
    for(int i=0; i<dimLog; i++)  
    {  
      vetSomma[i] = vetNumPar1[i] + vetNumPar2[i];  
    }  
  }  
}
```


81. C++, Cicli annidati

```
/**
```

```
ITIS-LS "Francesco Giordani" Caserta
```

```
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
```

```
Data: 01/03/2021
```

```
Numero progressivo dell'esercizio: es81
```

```
Versione: 1.0
```

```
Programmatore/i: Alessandro Esposito
```

```
Sistema Operativo: Windows 10
```

```
Compilatore/Interprete: code::blocks ver 20.03
```

```
Obiettivo didattico Cicli annidati
```

```
*/
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    for(int i=0; i<2; i++)
```

```
    {
```

```
        for(int j=0; j<3; j++)
```

```
        {
```

```
            cout << i << " ";
```

```
            cout << j << endl;
```

```
        }
```

```
        cout << "Stampa seconda linea" << endl;
```

```
    }
```

```
    return 0;
```

```
}
```

82. C++, stampare a video la tavola pitagorica

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 02/03/2021
Numero progressivo dell'esercizio: es82
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: Cicli annidati
Obiettivo del programma: Stampare a video la tavola pitagorica
*/
#include <iostream>
using namespace std;
const int dim=10;
int main()
{
    for(int i=1; i<=dim; i++)
    {
        cout << "Tabellina Del " << i << endl;
        for(int j=1; j<=dim; j++)
        {
            cout << i*j << '\t';
        }
        cout << endl;
    }
    return 0;
}
```

83. C++, caricare un vettore di interi e ordinarlo

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 05/03/2021
Numero progressivo dell'esercizio: es83
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo del programma: caricare un vettore di interi, ordinare i suoi elementi in ordine crescente e
stamparlo a video
*/
#include <iostream>
using namespace std;
int vet[100]={5,18,12,8};
int i,j;
int main()
{
    for(int i=0;i<4;i++) cout<<vet[i]<<" ";
    cout<<endl;
    for(int i=0;i<3;i++)
        for(int j=i+1;j<4;j++)
            {
                if(vet[i]>vet[j])
                {
                    int app=vet[j];
                    vet[j]=vet[i];
                    vet[i]=app;
                }
            }
    for(int i=0;i<4;i++) cout<<vet[i]<<" ";
    return 0;
}
```

84. C++, Caricare un vettore di interi e ordinarlo usando le procedure

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 05/03/2021
Numero progressivo dell'esercizio: es84
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo del programma: caricare un vettore di interi, ordinare i suoi elementi in ordine crescente e
stamparlo a video. Utilizzando le procedure caricaVet, stampaVet e ordinaVet
*/
#include <iostream>
using namespace std;
void caricaVet(int vetPar[], int dimLogPar);
void stampaVet(int vetPar[], int dimLogPar);
void ordinaVet(int vetPar[], int dimLogPar);
void scambia(int &num1Par, int &num2Par);
int dimLog;
int vet[100];
int main()
{
    cout << "Quanti elementi vuoi inserire? ";
    cin >> dimLog;
    cout << "INSERIMENTO DATI"<<endl;
    caricaVet(vet, dimLog);
    cout << "STAMPA DATI"<<endl;
    ordinaVet(vet, dimLog);
    cout << "ORDINAMENTO DATI"<<endl;
    stampaVet(vet, dimLog);
    return 0;
}
void caricaVet(int vetPar[], int dimLogPar)
{
    for(int i=0; i<dimLog; i++)
    {
        cout << "inserisci il numero nella posizione " << i << ": ";
        cin >> vetPar[i];
    }
}
void stampaVet(int vetPar[], int dimLogPar)
{
    cout << "Stampa del Vettore: " << endl;
    for(int i=0; i<dimLogPar; i++)
    {
        cout << vetPar[i] << " ";
    }
    cout << endl;
}
```

```
}  
void ordinaVet(int vetPar[], int dimLogPar)  
{  
    for(int i=0;i<dimLogPar-1;i++)  
    {  
        for(int j=i+1;j<dimLogPar;j++)  
        {  
            if(vetPar[i]>vetPar[j])  
            {  
                scambia(vetPar[i], vetPar[j]);  
            }  
        }  
    }  
}  
void scambia(int &num1Par, int &num2Par)  
{  
    int app;  
    app=num2Par;  
    num2Par=num1Par;  
    num1Par=app;  
}
```

85. C++, Caricare un vettore di interi e ordinarlo usando le procedure e i parametri

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 05/03/2021
Numero progressivo dell'esercizio: es85
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo del programma: caricare un vettore di interi, ordinare i suoi elementi in ordine crescente e
stamparlo a video. Utilizzando le procedure ottieniDim, caricaVet, stampaVet e ordinaVet
*/
#include <iostream>
using namespace std;
int const dimFis=100;
int dimLog;
int vet[100];
int ottieniDimLog(int dimFisPar);
void caricaVet(int vetPar[], int dimLogPar);
void stampaVet(int vetPar[], int dimLogPar);
void ordinaVet(int vetPar[], int dimLogPar);
void scambia(int &num1Par, int &num2Par);
int main()
{
    dimLog = ottieniDimLog(dimFis);
    caricaVet(vet, dimLog);
    cout << "STAMPA DEL VETTORE DOPO IL CARICAMENTO"<<endl;
    stampaVet(vet, dimLog);
    ordinaVet(vet, dimLog);
    cout<< "STAMPA DEL VETTORE DOPO L'ORDINAMENTO" <<endl;
    stampaVet(vet, dimLog);
    return 0;
}
int ottieniDimLog(int dimFisPar)
{
    int dimensioneL;
    do
    {
        cout << "Quanti elementi vuoi inserire: (MAX: "<<dimFisPar<<") ";
        cin >> dimensioneL;
    }
    while(dimensioneL>dimFisPar || dimensioneL<0);
    return dimensioneL;
}
void caricaVet(int vetPar[], int dimLogPar)
{
    for(int i=0; i<dimLog; i++)
    {
```

```

    cout << "inserisci il numero nella posizione " << i << ": ";
    cin >> vetPar[i];
}
}
void stampaVet(int vetPar[], int dimLogPar)
{
    for(int i=0;i<dimLogPar;i++)
    {
        cout << vetPar[i] << " ";
    }
    cout << endl;
}
void ordinaVet(int vetPar[], int dimLogPar)
{
    for(int i=0;i<dimLogPar-1;i++)
    {
        for(int j=i+1;j<dimLogPar;j++)
        {
            if(vetPar[i]>vetPar[j])
            {
                scambia(vetPar[i], vetPar[j]);
            }
        }
    }
}
void scambia(int &num1Par, int &num2Par)
{
    int app;
    app=num2Par;
    num2Par=num1Par;
    num1Par=app;
}

```

86. C++, Ricerca Lineare

/**

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 09/03/2021

Numero progressivo dell'esercizio: es86

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: code::blocks ver 20.03

Obiettivo didattico: Ricerca Lineare

Obiettivo del programma: Scrivere la procedura per ricercare un elemento all'interno di un vettore e stampare a video la posizione del elemento ricercato. Altrimenti stampare ricerca senza successo.

*/

```
#include <iostream>
```

```
using namespace std;
```

```
int vet[50]={5, 44, 62, 96, 100};
```

```
int dimLog=5;
```

```
int ricercaLineare(int vetPar[], int dimLogPar, int kPar);
```

```
int main()
```

```
{
```

```
    int k=96;
```

```
    if (ricercaLineare(vet, 4, k) == -1)
```

```
        cout << "ricerca senza successo";
```

```
    else
```

```
        cout << "ricerca con successo nella posizione " << ricercaLineare(vet, 4, k);
```

```
}
```

```
int ricercaLineare(int vetPar[], int dimLogPar, int kPar)
```

```
{
```

```
    int i=0, trovato=-1;
```

```
    while(i<dimLogPar && trovato == -1)
```

```
    {
```

```
        if(vetPar[i]==kPar) trovato=i;
```

```
        else i++;
```

```
    }
```

```
    return trovato;
```

```
}
```


87. Python, sintesi vocale

'''

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021

Classe 3^ sez.B spec. Informatica e telecomunicazioni

Data: 12/03/2021

Numero progressivo dell'esercizio: es87

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Python 3.8 repl.it

Obiettivo didattico: Sintesi Vocale

Obiettivo del programma: generare un file audio che legga il contenuto della variabile text con la voce di google

'''

```
from gtts import gTTS
```

```
text="Ciao a tutti. Oggi il professore di informatica vi fa ascoltare la voce di google"
```

```
tts=gTTS(text=text,lang='it')
```

```
tts.save("tts_output_audio.mp3")
```

88. Python, sintesi vocale e conteggio parole

'''

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021

Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 12/03/2021

Numero progressivo dell'esercizio: es88

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Python 3.8 repl.it

Obiettivo didattico: Sintesi Vocale

Obiettivo del programma: generare un file audio che legga il contenuto della variabile text e conti il numero di parole con la voce di Google

'''

```
from gtts import gTTS
```

```
text="Ciao ragazzi abbiamo scritto un programma python che conta le parole del nostro testo."
```

```
text2="Questo testo contiene"
```

```
text3=str(len(text.split()))
```

```
text4="parole"
```

```
text=text+text2+text3+text4
```

```
tts=gTTS(text=text,lang='it')
```

```
tts.save("tts_output_audio.mp3")
```

89. Python, sintesi vocale e conteggio parole e vocali

'''

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021

Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 12/03/2021

Numero progressivo dell'esercizio: es88

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Python 3.8 repl.it

Obiettivo didattico: Sintesi Vocale

Obiettivo del programma: generare un file audio che legga il contenuto della variabile text e conti il numero di parole con la voce di Google

'''

```
from gtts import gTTS
```

```
text="Ciao ragazzi abbiamo scritto un programma python che conta le parole del nostro testo."
```

```
text2="Questo testo contiene"
```

```
text3=str(len(text.split()))
```

```
text4="parole"
```

```
text=text+text2+text3+text4
```

```
tts=gTTS(text=text,lang='it')
```

```
tts.save("tts_output_audio.mp3")
```

90. Python, sintesi vocale e conteggio parole e vocali di un file esterno

'''

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021

Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 12/03/2021

Numero progressivo dell'esercizio: es90

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Python 3.8 repl.it

Obiettivo didattico: Sintesi Vocale

Obiettivo del programma: generare un file audio che legga il contenuto del file esempio.txt , conti il numero di parole e il numero delle vocali con la voce di Google

'''

```
from gtts import gTTS
text=open("esempio.txt", "r").read()
text2="Questo testo contiene"
text3=str(len(text.split()))
text4="parole."
vocali= 'A,a,E,e,I,i,O,o,U,u'
contaVocali = 0
for each_char in text:
if each_char in vocali:
    contaVocali+= 1
text5=" e "+str(contaVocali) + "vocali"
text=text+text2+text3+text4+text5
tts=gTTS(text=text,lang='it')
tts.save("tts_output_audio.mp3")
```

Principali differenze tra python e c++

Le principali differenze tra il Python e il C++ sono la mancanza di una funzione main obbligatoria nel Python. Nella dichiarazione di variabili e procedure/funzioni, in C++ bisogna specificare ogni cosa mentre nel Python non è necessario. Nel utilizzo delle parentesi, in C++ sono obbligatorie per definire funzioni, procedure, cicli, selezione ecc. mentre in Python servono solo in rari casi. L'indentazione che nel C++ non è necessaria al compilatore mentre nel Python è importantissima. E le librerie che nel Python sono moltissime per vari utilizzi, IA, numeri casuali, riproduzione suoni, operazioni sul PC ecc. e facili da scaricare mentre nel C++ sono di meno e con più difficoltà di scaricamento. Infine il Python lascia molte cose sottointese anche se questo non è un bene dato che lascia poca personalizzazione al utente e per raggiungere una personalizzazione come il C++ diventa più difficile.

C++, scambio di elementi al interno del vettore

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 14/03/2021
Numero progressivo dell'esercizio: es92
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: scambio di elementi nel vettore
Obiettivo del programma: caricare un vettore di interi, scambiare il primo elemento con l'ultimo e
stamparlo a video
*/
#include <iostream>
using namespace std;
char vet[100];
int dimLog;
int ottieniDim(int dimFisPar);
void caricaVet(char vetPar[], int dimLogPar);
void stampaVet(char vetPar[], int dimLogPar);
void scambiaPrimoUlt(char vetPar[], int dimLogPar);
int main()
{
    dimLog = ottieniDim(100);
    caricaVet(vet, dimLog);
    cout << "stampa prima dello scambio" << endl;
    stampaVet(vet, dimLog);
    cout << endl;
    scambiaPrimoUlt(vet, dimLog);
    cout << "stampa dopo lo scambio" << endl;
    stampaVet(vet, dimLog);
    return 0;
}
int ottieniDim(int dimFisPar)
{
    int dim;
    do
    {
        cout << "quanti caratteri vuoi inserire? ";
        cin >> dim;
    }while(dim<0 || dim>dimFisPar);
    return dim;
}
void caricaVet(char vetPar[], int dimLogPar)
{
    for(int i=0; i<dimLogPar; i++)
    {
        cout << "inserisci un carattere: ";
        cin >> vetPar[i];
    }
}
```

```
void stampaVet(char vetPar[], int dimLogPar)
{
    for(int i=0;i<dimLog;i++)
    {
        cout << vet[i];
    }
}
void scambiaPrimoUlt(char vetPar[], int dimLogPar)
{
    char app;
    app = vetPar[0];
    vetPar[0] = vetPar[dimLogPar-1];
    vetPar[dimLogPar-1] = app;
}
```

C++, procedura update

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 15/03/2021
Numero progressivo dell'esercizio: es93
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: procedura update
Obiettivo del programma: caricare un vettore di caratteri, stamparlo a video, eseguire l'update di un
carattere in una posizione scelta e ristamparlo a video
*/
#include <iostream>
using namespace std;
char vet[100];
int dimLog;
int const dimFis=100;
int ottieniDimLog(int dimFisPar);
void caricaVet(char vetPar[], int dimLogPar);
void stampaVet(char vetPar[], int dimLogPar);
void update(char vetPar[], int posPar, char carPar);
int main()
{
    int pos;
    char car;
    dimLog = ottieniDimLog(dimFis);
    caricaVet(vet, dimLog);
    stampaVet(vet, dimLog);
    cout << endl;
    cout << "in che posizione vuoi effettuare l'update? ";
    cin >> pos;
    cout << "quale carattere vuoi inserire? ";
    cin >> car;
    update(vet, pos, car);
    stampaVet(vet, dimLog);
    return 0;
}
int ottieniDimLog(int dimFisPar)
{
    int dim;
    do{
        cout<<"quanti caratteri vuoi inserire? ";
        cin >> dim;
    }while(dim<0 || dim>dimFisPar);
    return dim;
}
void caricaVet(char vetPar[], int dimLogPar)
{
    for(int i=0;i<dimLog;i++)
    {
```



```
    cout << "inserisci un carattere: ";
    cin >> vetPar[i];
}
}
void stampaVet(char vetPar[], int dimLogPar)
{
    for(int i=0;i<dimLog;i++)
    {
        cout<<vetPar[i]<<" ";
    }
}
void update(char vetPar[], int posPar, char carPar)
{
    vetPar[posPar] = carPar;
}
```

C++, procedura update con due vettori

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 16/03/2021
Numero progressivo dell'esercizio: es94
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: procedura update
Obiettivo del programma: caricare due vettori di caratteri, stamparli a video, eseguire l'update di un
carattere in una posizione scelta in entrambi i vettori e ristamparli a video
*/
#include <iostream>
using namespace std;
char vet[100];
char vet2[50];
int dimLog;
int const dimFis=100;
int ottieniDimLog(int dimFisPar);
void caricaVet(char vetPar[], int dimLogPar);
void stampaVet(char vetPar[], int dimLogPar);
void update(char vetPar[],int dimLogPar, int posPar, char carPar);
int main()
{
    int pos;
    char car;
    dimLog = ottieniDimLog(dimFis);
    caricaVet(vet, dimLog);
    stampaVet(vet, dimLog);
    cout << endl;
    caricaVet(vet2, dimLog);
    stampaVet(vet2, dimLog);
    cout << endl;
    cout << "in che posizione vuoi effettuare l'update? ";
    cin >> pos;
    cout << "quale carattere vuoi inserire? ";
    cin >> car;
    update(vet, dimLog, pos, car);
    stampaVet(vet, dimLog);
    cout << endl;
    update(vet2, dimLog, pos, car);
    stampaVet(vet2, dimLog);
    return 0;
}
int ottieniDimLog(int dimFisPar)
{
    int dim;
    do{
        cout<<"quanti caratteri vuoi inserire? ";
        cin >> dim;
    }
}
```

```
    }while(dim<0 || dim>dimFisPar);
    return dim;
}
void caricaVet(char vetPar[], int dimLogPar)
{
    for(int i=0;i<dimLog;i++)
    {
        cout << "inserisci un carattere: ";
        cin >> vetPar[i];
    }
}
void stampaVet(char vetPar[], int dimLogPar)
{
    for(int i=0;i<dimLog;i++)
    {
        cout<<vetPar[i]<<" ";
    }
}
void update(char vetPar[], int dimLogPar, int posPar, char carPar)
{
    if(posPar<=dimLogPar)
        vetPar[posPar] = carPar;
}
}
```

Python e C++, conto alla rovescia

```
/**  
ITIS-LS "Francesco Giordani" Caserta  
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni  
Data: 18/03/2021  
Numero progressivo dell'esercizio: es95  
Versione: 1.0  
Programmatore/i: Alessandro Esposito  
Sistema Operativo: Windows 10  
Compilatore/Interprete: code::blocks ver 20.03  
Obiettivo didattico: conto alla rovescia  
Obiettivo del programma: eseguire il conto alla rovescia da 10 a 1 in C++  
*/
```

```
#include <iostream>  
#include <thread>  
#include <chrono>  
using namespace std;  
int main()  
{  
    for (int i=10; i>0; i=i-1)  
    {  
        cout << i << endl;  
        this_thread::sleep_for (chrono::seconds(1));  
    }  
    return 0;  
}
```

'''

```
ITIS-LS "Francesco Giordani" Caserta  
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni  
Data: 18/03/2021  
Numero progressivo dell'esercizio: es95  
Versione: 1.0  
Programmatore/i: Alessandro Esposito  
Sistema Operativo: Windows 10  
Compilatore/Interprete: Python 3.8.1  
Obiettivo didattico: procedura update  
Obiettivo del programma: eseguire il conto alla rovescia da 10 a 1 inPython  
'''
```

```
import time as t  
for r in range(10,0,-1):  
    t.sleep(1)  
    print(r)
```

Python e C++, conto in avanti e conto alla rovescia

/**

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 18/03/2021

Numero progressivo dell'esercizio: es96

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: code::blocks ver 20.03

Obiettivo didattico: conto in avanti e conto alla rovescia

Obiettivo del programma: con un menu l'utente può scegliere di contare da 1 a 50 oppure decrementare da 50 a 1 in C++

*/

```
#include <iostream>
#include <cstdlib>
#include <thread>
#include <chrono>
using namespace std;
int main()
{
    int risp;
    do{
        system("clear");
        cout << "MENU"<<endl;
        cout << "1) Conto alla rovescia"<<endl;
        cout << "2) Conto in avanti"<<endl;
        cout << "3) Exit"<<endl;
        cout << "inserisci il numero dell'operazione che vuoi svolgere:";
        cin >> risp;
        switch(risp)
        {
            case 1:
                for (int i=50; i>0; i=i-1)
                {
                    cout << i << endl;
                    this_thread::sleep_for (chrono::milliseconds(1000));
                }
                break;
            case 2:
                for (int i=1; i<51; i++)
                {
                    cout << i << endl;
                    this_thread::sleep_for (chrono::seconds(1));
                }
                break;
        }
        cin.get()
    }while(risp!=3);
    return 0;
}
```

'''

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni

Data: 18/03/2021

Numero progressivo dell'esercizio: es96

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: code::blocks ver 20.03

Obiettivo didattico: procedura update

Obiettivo del programma: con un menu l'utente può scegliere di contare da 1 a 50 oppure decrementare da 50 a 1 in Python

'''

```
import time as t
import os
import replit
while True:
    replit.clear()
    print("MENU")
    print("1) Conto alla rovescia")
    print("2) Conto normale")
    print("3) Exit")
    risp = input("inserisci il numero della tua scelta: ")
    if risp=='1':
        for r in range(50,0,-1):
            t.sleep(1)
            print(r)
    elif risp=='2':
        for r in range(1,51):
            t.sleep(1)
            print(r)
    elif risp=='3':
        break
    os.system("pause")
```

Python, selezione a una via

'''

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 22/03/2021

Numero progressivo dell'esercizio: es97

Versione: 1.1

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Python 3.8

Obiettivo didattico: la selezione a una via

Obiettivo del programma: acquisire 5 numeri da tastiera e stampare a video quelli maggiori di 10

'''

```
for r in range(5):
    num = int(input(str(r)+" inserisci un numero intero: "))
    if num>10:
        print(num)
```

Python e C++, somma di due vettori

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 23/03/2021
Numero progressivo dell'esercizio: es98
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: somma dei vettori
Obiettivo del programma: Sommare due vettori
*/
#include <iostream>
using namespace std;
int vet1[100];
int vet2[100];
int vetSomma[100];
int dimLog=6;
void caricaVet(int vetNumPar[], int dimLogPar);
void stampaVet(int vetNumPar[], int dimLogPar);
void sommaVet(int vetNumPar1[], int vetNumPar2[], int vetSommaPar[], int dimLogPar);
int main()
{
    caricaVet(vet1, dimLog);
    cout << endl;
    caricaVet(vet2, dimLog);
    cout << "Stampa Primo Vettore" << endl;
    stampaVet(vet1, dimLog);
    cout << endl;
    cout << "Stampa Secondo Vettore" << endl;
    stampaVet(vet2, dimLog);
    cout << endl;
    sommaVet(vet1, vet2, vetSomma, dimLog);
    cout << "Somma dei due vettori" << endl;
    stampaVet(vetSomma, dimLog);
    return 0;
}
void caricaVet(int vetNumPar[], int dimLogPar)
{
    for(int i=0; i<dimLog; i++)
    {
        cout << "inserire l'elemento numero " << i << " del vettore: ";
        cin >> vetNumPar[i];
    }
}
void stampaVet(int vetNumPar[], int dimLogPar)
{
    for(int i=0; i<dimLog; i++)
    {
        cout << vetNumPar[i] << " - ";
    }
}
void sommaVet(int vetNumPar1[], int vetNumPar2[], int vetSommaPar[], int dimLogPar)
{
```



```

int somma;
for(int i=0; i<dimLog; i++)
{
    vetSomma[i] = vetNumPar1[i] + vetNumPar2[i];
}
}
"""

```

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 23/03/2021

Numero progressivo dell'esercizio: es98

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Python 3.8

Obiettivo didattico: somma dei vettori

Obiettivo del programma: Sommare due vettori

"""

```

vet1=[0, 1, 2, 3, 4, 5, 6,]
vet2=[7, 8, 9, 10, 11, 12, 13]
vetSomma=[0, 0, 0, 0, 0, 0, 0]
dimLog = 6
def stampaVet(vetPar, dimLog):
    for r in range(dimLog):
        print(vetPar[r])
def sommaVet(vetPar1, vetPar2, vetSommaPar, dimLog):
    for r in range(dimLog):
        vetSommaPar[r]= int(vetPar1[r]) + int(vetPar2[r])
vet1=[0, 1, 2, 3, 4, 5, 6,]
dimLog = 6
sommaVet(vet1, vet2, vetSomma, dimLog)
stampaVet(vetSomma, dimLog)

```

Python, cambiare il colore di sfondo della tartaruga

'''

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 25/03/2021

Numero progressivo dell'esercizio: es99

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Python 3.8.1

Obiettivo didattico: la tartaruga

Obiettivo del programma: cambiare il colore di sfondo

'''

```
from turtle import Turtle, Screen
```

```
tartaruga = Turtle()
```

```
sfondo = Screen()
```

```
sfondo.colormode(255)
```

```
R = 255
```

```
G = 255
```

```
B = 0
```

```
sfondo.bgcolor((R, G, B))
```

Python, disegnare un cerchio in loop infinito con la tartaruga

'''

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2020/2021 Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 25/03/2021

Numero progressivo dell'esercizio: es100

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Python 3.8.1

Obiettivo didattico: la tartaruga

Obiettivo del programma: disegnare un cerchio in loop infinito

'''

```
from turtle import Turtle, Screen
```

```
turtle = Turtle()
```

```
while True:
```

```
turtle.forward(1) turtle.right(1)
```

Python e C++, data una città stampare a video il numero degli abitanti

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 26/03/2021
Numero progressivo dell'esercizio: es101
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: vettori paralleli
Obiettivo del programma: data una città scrivere a video il numero degli abitanti
*\
#include <iostream>
using namespace std;
int const dimFis=50;
char vetCitta[dimFis]= {'N','C','S','A','B'};
long int vetAbitanti[dimFis]={1000000, 80000, 130000, 53000, 57000};
char citta;
int ricercaLineare(char vetPar[], int dimLogPar, char kPar);
int main() {
    cout << "inserisci il nome della citta: ";
    cin >> citta;
    if (ricercaLineare(vetCitta, 5, citta) == -1)
        cout<<"la citta' non e' presente nel vettore";
    else
        cout << "la citta' "<<citta<<" ha: "<<
            vetAbitanti[ricercaLineare(vetCitta, 5, citta)] << " abitanti";
}
int ricercaLineare(char vetPar[], int dimLogPar, char kPar)
{
    int i=0, trovato=-1;
    while(i<dimLogPar && trovato == -1)
    {
        if(vetPar[i]==kPar) trovato=i;
        else i++;
    }
    return trovato;
}

'''
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 26/03/2021
Numero progressivo dell'esercizio: es101
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: Python 3.8.1
Obiettivo didattico: vettori paralleli
Obiettivo del programma: data una città scrivere a video il numero degli abitanti
```

```
'''
vetCitta = ['N', 'C', 'S', 'A', 'B']
vetAbitanti = [1000000, 80000, 130000, 53000, 57000]
def ricercaLineare(vetPar, dimLogPar, kPar):
    i = 0
    trovato = -1
    while(i < dimLogPar and trovato == -1):
        if vetPar[i] == kPar:
            trovato = i
        else:
            i += 1
    return trovato;
citta = input("inserisci il nome della città: ")
if ricercaLineare(vetCitta, 5, citta) == -1:
    print("La città non è presente nel vettore")
else:
    print("La città "+citta+" ha: "+str(vetAbitanti[ricercaLineare(vetCitta, 5, citta)])+" abitanti")
```

Python e C++, data una città stampare a video il numero degli abitanti (stringhe)

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 27/03/2021
Numero progressivo dell'esercizio: es102
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: code::blocks ver 20.03
Obiettivo didattico: vettori paralleli
Obiettivo del programma: data una città scrivere a video il numero degli abitanti con le stringhe
*\
#include <iostream>
#include <string>
using namespace std;
int const dimFis=50;
string vetCitta[dimFis]= {"Napoli","Caserta","Salerno","Avellino","Benevento"};
long int vetAbitanti[dimFis]={1000000, 80000, 130000, 53000, 57000};
string citta;
int ricercaLineare(string vetPar[], int dimLogPar, string kPar);
int main() {
    cout << "inserisci il nome della citta': ";
    cin >> citta;
    if (ricercaLineare(vetCitta, 5, citta) == -1)
        cout<<"la citta' non e' presente nel vettore ";
    else
        cout << "la citta': "<<citta<<" ha: "<<
            vetAbitanti[ricercaLineare(vetCitta, 5, citta)] << " abitanti";
}
int ricercaLineare(string vetPar[], int dimLogPar, string kPar)
{
    int i=0, trovato=-1;
    while(i<dimLogPar && trovato == -1)
    {
        if(vetPar[i]==kPar) trovato=i;
        else i++;
    }
    return trovato;
}

'''
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 27/03/2021
Numero progressivo dell'esercizio: es102
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
```

Compilatore/Interprete: python 3.8.1

Obiettivo didattico: vettori paralleli

Obiettivo del programma: data una città scrivere a video il numero degli abitanti con le stringhe

"""

```
vetCitta = ['Napoli', 'Caserta', 'Salerno', 'Avellino', 'Benevento']
```

```
vetAbitanti = [1000000, 80000, 130000, 53000, 57000]
```

```
def ricercaLineare(vetPar, dimLogPar, kPar):
```

```
    i = 0
```

```
    trovato = -1
```

```
    while(i < dimLogPar and trovato == -1):
```

```
        if vetPar[i] == kPar:
```

```
            trovato = i
```

```
        else:
```

```
            i += 1
```

```
    return trovato;
```

```
citta = input("inserisci il nome della città: ")
```

```
if ricercaLineare(vetCitta, 5, citta) == -1:
```

```
    print("La città non è presente nel vettore")
```

```
else:
```

```
    print("La città "+citta+" ha: "+str(vetAbitanti[ricercaLineare(vetCitta, 5, citta)])+" abitanti")
```

C++, operazioni con le temperature delle città (vettori paralleli)

```
/**
```

```
ITIS-LS "Francesco Giordani" Caserta
```

```
Anno scolastico: 2020/2021 Classe 3^ sez.B spec. Informatica e telecomunicazioni
```

```
Data: 29/03/2021
```

```
Numero progressivo dell'esercizio: es103
```

```
Versione: 1.0
```

```
Programmatore/i: Alessandro Esposito
```

```
Sistema Operativo: Windows 10
```

```
Compilatore/Interprete: code::blocks ver 20.03
```

```
Obiettivo didattico: vettori paralleli
```

```
Obiettivo del programma: data una città stampare a video la temperatura minima, la temperatura massima,
```

```
la temperatura media, l'escursione termica, la temperatura media delle temperature minime e la temperatura
```

```
media delle temperature massime. Facoltativo: mettere in ordine crescente le città e le relative temperature.
```

```
*/
```

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
int const dimFis = 50;
```

```
int dimLog = 5;
```

```
string vetCitta[dimFis] = {"Napoli", "Caserta", "Salerno", "Avellino", "Benevento"};
```

```
int vetTempMin[dimFis] = {8,7,11,8,9};
```

```
int vetTempMax[dimFis] = {19,22,19,22,22};
```

```
string citta;
```

```
int tempM(int vetPar[], int dimLogPar);
```

```
int ricercaLineare(string vetPar[], int dimLogPar, string kPar);
```

```
int tempMedia(int num1Par, int num2Par);
```

```
void ordinaCitta(string vetCittaPar[], int vetMinPar[], int vetMaxPar[], int dimLogPar);
```

```
void scambiaCitta(string &citta1, string &citta2);
```

```
void scambiaTemp(int &temp1, int &temp2);
```

```
void stampaC(string cittaPar[], int dimLogPar);
```

```
void stampaT(int tempPar[], int dimLogPar);
```

```
int main()
```

```
{
```

```
    int escTer, pos;
```

```
    cout << "inserisci il nome della citta' ";
```

```
    cin >> citta;
```

```
    if(ricercaLineare(vetCitta, dimLog,citta)==-1)
```

```
    {
```

```
        cout << "la citta' inserita non e' presente nel vettore" << endl;
```

```
    }
```

```
    else
```

```
    {
```

```
        pos = ricercaLineare(vetCitta, dimLog,citta);
```

```
        escTer = vetTempMax[pos]-vetTempMin[pos];
```

```
        cout << "Escursione termica di "<< citta << ": " << escTer << " C" << endl;
```

```
        cout << "Temperatura minima di "<<citta<<": "<<vetTempMin[pos] << " C" <<endl;
```

```
        cout << "Temperatura massima di "<<citta<<": "<<vetTempMax[pos] << " C" <<endl;
```



```

cout << "Temperatura media di "<<citta<<": "<<tempMedia(vetTempMin[pos], vetTempMax[pos]) << " C"
<<endl;
cout << "Temperatura minima media Campana: "<<tempM(vetTempMin, dimLog) << " C" << endl;
cout << "Temperatura massima media Campana: "<<tempM(vetTempMax, dimLog) << " C" << endl;
ordinaCitta(vetCitta, vetTempMin, vetTempMax, dimLog);
cout << "Stampa citta' in ordine crescente: " << endl;
stampaC(vetCitta, dimLog);
cout << endl;
/*
cout << "Stampa temperature minime: " << endl;
stampaT(vetTempMin, dimLog);
cout << endl;
    cout << "Stampa temperature massime: " << endl;
stampaT(vetTempMax, dimLog);
    cout << endl;
*/
}
return 0;
}
int ricercaLineare(string vetPar[], int dimLogPar, string kPar)
{
    int i=0, trovato=-1;
    while(i<dimLogPar && trovato == -1)
    {
        if(vetPar[i]==kPar) trovato=i;
        else i++;
    }
    return trovato;
}
int tempM(int vetPar[], int dimLogPar)
{
    int somma=0;
    for(int i=0;i<dimLogPar;i++)
    {
        somma=somma+vetPar[i];
    }
    return somma/dimLogPar ;
}
int tempMedia(int num1Par, int num2Par)
{
    int somma;
    somma = num1Par + num2Par;
    return somma/2;
}
void ordinaCitta(string vetCittaPar[], int vetMinPar[], int vetMaxPar[], int dimLogPar)
{
    for(int i=0;i<dimLogPar-1;i++)
    {
        for(int j=i+1;j<dimLogPar;j++)
        {
            if(vetCittaPar[i]>vetCittaPar[j])
            {
                scambiaCitta(vetCittaPar[i], vetCittaPar[j]);
            }
        }
    }
}

```

```

        scambiaTemp(vetMinPar[i], vetMinPar[j]);
        scambiaTemp(vetMaxPar[i], vetMaxPar[j]);
    }
}
}
void scambiaCitta(string &citta1, string &citta2)
{
    string app;
    app = citta1;
    citta1 = citta2;
    citta2 = app;
}
void scambiaTemp(int &temp1, int &temp2)
{
    int app;
    app = temp1;
    temp1 = temp2;
    temp2 = app;
}
void stampaC(string cittaPar[], int dimLogPar)
{
    for(int i=0;i<dimLogPar;i++)
    {
        cout << cittaPar[i] << " ";
    }
}
void stampaT(int tempPar[], int dimLogPar)
{
    for(int i=0;i<dimLogPar;i++)
    {
        cout << tempPar[i] << " ";
    }
}

```

HTML, realizzare una pagina contenente un messaggio

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 31/03/2021

Numero progressivo dell'esercizio: es104

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Google Chrome

Obiettivo didattico: Costruire la prima pagina web(due sezioni: head e body)

Obiettivo del programma: Realizzare una pagina contenente un messaggio

```
<HTML><HEAD></HEAD>
```

```
<BODY>
```

E' davvero semplice !!!

```
</BODY>
```

```
</HTML>
```

HTML, realizzare una pagina web contenente un titolo

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 31/03/2021

Numero progressivo dell'esercizio: es105

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Microsoft Edge

Obiettivo didattico: Il tag Title

Obiettivo del programma: Realizzare una pagina web con un titolo

```
<HTML><HEAD><TITLE>INTESTAZIONE</TITLE></HEAD>
```

```
<BODY>
```

È davvero semplice !!!

```
</BODY>
```

```
</HTML>
```

HTML, realizzare una pagina web con i tag h1...h6 e center

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3^ sez.B spec. Informatica e telecomunicazioni

Data: 31/03/2021

Numero progressivo dell'esercizio: es106

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Microsoft Edge

Obiettivo didattico: I tag h1...h6, center

Obiettivo del programma: Realizzare una pagina web con i tag h1...h6, center

```
<HTML><HEAD><TITLE>ESERCIZIO n°3 </TITLE></HEAD>
```

```
<BODY>
```

```
<H1><CENTER>CASERTA </CENTER></H1>
```

```
<H2>REGGIA DI CASERTA </H2>
```

```
<H3>CASERTAVECCHIA</H3>
```

```
<H4>SAN LEUCIO </H4>
```

```
<H5>SAN PIETRO AD MONTES DI CASOLLA </H5>
```

```
<H6>SAN RUFO DI CASOLLA </H6>
```

```
</BODY>
```

```
</HTML>
```

HTML, realizzare una pagina web con i tag strong, p e l'attributo text

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 31/03/2021

Numero progressivo dell'esercizio: es107

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Microsoft Edge

Obiettivo didattico: I tag strong p e l'attributo text

Obiettivo del programma: Realizzare una pagina web con i tag strong p e l'attributo text (costruire il colore con la tavolozza PAINT)

```
<HTML>
```

```
<HEAD></HEAD>
```

```
<BODY TEXT="4169E1">
```

```
<STRONG>LA REGGIA DI CASERTA</STRONG>
```

```
<P>Occupa una superficie di 44.000 metri quadrati,<BR>  
con 1200 vani, di cui 134 destinato agli alloggi reali.<BR>
```

```
Concepita come struttura polifunzionale ...</P>
```

```
</BODY>
```

```
</HTML>
```

HTML, realizzare una pagina con un collegamento ipertestuale

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 31/03/2021

Numero progressivo dell'esercizio: es108

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Microsoft Edge

Obiettivo didattico: Collegamenti ipertestuali

Obiettivo del programma: Realizzare una pagina contenente un collegamento ipertestuale

```
<HTML><HEAD>
```

```
<TITLE>Collegamento con un documento locale</TITLE></HEAD>
```

```
<BODY >
```

```
<H1><CENTER>CASERTA </CENTER></H1>
```

```
<H2><A HREF ="esercizio4.html"> REGGIA DI CASERTA </A></H2>
```

```
<H3>CASERTAVECCHIA</H3>
```

```
<H4>SAN LEUCIO </H4>
```

```
<H5>SAN PIETRO AD MONTES DI CASOLLA </H5>
```

```
<H6>SAN RUFO DI CASOLLA </H6>
```

```
</BODY>
```

```
</HTML>
```

HTML, realizzare una pagina con un collegamento ipertestuale ed un segnalibro

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 07/04/2021

Numero progressivo dell'esercizio: es109

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Google Chrome

Obiettivo didattico: Collegamenti ipertestuali e segnalibro

Obiettivo del programma: Realizzare una pagina contenente un collegamento ipertestuale ed un segnalibro

```
<HTML><HEAD><TITLE>Collegamento con una particolare posizione del documento  
corrente</TITLE></HEAD>
```

```
<BODY >
```

```
<H1><CENTER>CASERTA </CENTER></H1>
```

```
<H2><A HREF ="es4.htm"> REGGIA DI CASERTA </A></H2>
```

```
<H3>CASERTAVECCHIA</H3>
```

```
<H4>SAN LEUCIO </H4>
```

```
<H5>SAN PIETRO AD MONTES DI CASOLLA </H5>
```

```
<H6>SAN RUFO DI CASOLLA </H6>
```

```
<P>
```

```
<A HREF="#SOTTOMENU"> SOTTOMENU </A><BR>
```

```
<P>
```

```
<BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR><BR>
```

```
<A NAME="SOTTOMENU">
```

```
<H3>La mappa della città<BR>
```

```
i numeri di telefono utili<BR>
```

```
i servizi<BR></H3>
```

```
</BODY>
```

```
</HTML>
```


HTML, realizzare una pagina con un collegamento ipertestuale ed un segnalibro esterno

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 07/4/2021

Numero progressivo dell'esercizio: es110

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Google Chrome

Obiettivo didattico: Collegamenti ipertestuali ed un segnalibro esterno

Obiettivo del programma: Realizzare una pagina contenente un collegamento ipertestuale ed un segnalibro esterno

```
<HTML><HEAD><TITLE>Collegamento con una particolare posizione del documento  
corrente</TITLE></HEAD>
```

```
<BODY >
```

```
  <H1><CENTER>CASERTA </CENTER></H1>
```

```
  <H2><A HREF ="REGGIA.htm#PARCO"> REGGIA DI CASERTA </A></H2>
```

```
  <H3>CASERTAVECCHIA</H3>
```

```
  <H4>SAN LEUCIO </H4>
```

```
  <H5>SAN PIETRO AD MONTES DI CASOLLA </H5>
```

```
  <H6>SAN RUFO DI CASOLLA </H6>
```

```
</BODY>
```

```
</HTML>
```

HTML, realizzare una pagina contenente elenchi puntati annidati

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 07/04/2021

Numero progressivo dell'esercizio: es111

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Google Chrome

Obiettivo didattico: Elenchi puntati

Obiettivo del programma: Realizzare una pagina contenente elenchi puntati annidati

```
<HTML><HEAD><TITLE>Elenchi puntati</TITLE></HEAD>
```

```
<BODY >
```

```
<LH> TITOLO LISTA</LH>
```

```
<UL>
```

```
<LI> PRIMO ELEMENTO PUNTATO </LI>
```

```
<LI> SECONDO ELEMENTO PUNTATO </LI>
```

```
<UL>
```

```
<LI> PRIMO ELEMENTO PUNTATO LISTA ANNIDATA
```

```
<LI> SECONDO ELEMENTO PUNTATO LISTA ANNIDATA
```

```
</UL>
```

```
<LI> TERZO ELEMENTO PUNTATO
```

```
</UL>
```

```
</BODY>
```

```
</HTML>
```

HTML, realizzare una pagina contenente un elenco numerato

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 07/04/2021

Numero progressivo dell'esercizio: es112

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Google Chrome

Obiettivo didattico: Elenchi numerati

Obiettivo del programma: Realizzare una pagina contenente un elenco numerato

```
<HTML><HEAD><TITLE>Elenchi puntati numerati</TITLE></HEAD>
```

```
<BODY >
```

```
  <OL TYPE="I" START="2">
```

```
    <LI> PRIMO ELEMENTO PUNTATO
```

```
    <LI> SECONDO ELEMENTO PUNTATO
```

```
  </OL>
```

```
</BODY>
```

```
</HTML>
```

HTML, realizzare una pagina contenente un elenco misto (numerato/puntato)

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 09/04/2021

Numero progressivo dell'esercizio: es113

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Google Chrome

Obiettivo didattico: Elenchi numerati e puntati

Obiettivo del programma: Realizzare una pagina contenente un elenco misto(numerato/puntato)

```
<HTML><HEAD><TITLE>Lista mista</TITLE></HEAD>
```

```
<BODY >
```

```
  <OL TYPE="1" START="1"><LI> Riviste</LI>
```

```
  <UL><LI>informatica</LI>
```

```
  <LI>medicina</LI></UL>
```

```
  <LI> Libri</LI>
```

```
  <UL><LI>informatica</LI>
```

```
  <LI>medicina</LI></UL></OL>
```

```
</BODY>
```

```
</HTML>
```

HTML, realizzare una pagina contenente una tabella

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 09/04/2021

Numero progressivo dell'esercizio: es114

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Google Chrome

Obiettivo didattico: Le tabelle

Obiettivo del programma: Realizzare una pagina contenente una tabella

```
<HTML><HEAD><TITLE>tabelle</TITLE></HEAD>
```

```
<BODY >
```

```
<TABLE border=2>
```

```
<TR>
```

```
<TD> A11 </TD>
```

```
<TD> A12 </TD>
```

```
<TD> A13 </TD>
```

```
</TR>
```

```
<TR>
```

```
<TD> A21 </TD>
```

```
<TD> A22 </TD>
```

```
<TD> A23 </TD>
```

```
</TR>
```

```
</TABLE>
```

```
</BODY>
```

```
</HTML>
```

115. HTML, realizzare una pagina contenente una tabella

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 12/04/2021

Numero progressivo dell'esercizio: es115

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Google Chrome

Obiettivo didattico: Le tabelle

Obiettivo del programma: Realizzare una pagina contenente una tabella

```
<HTML><HEAD><TITLE>tabelle</TITLE></HEAD>
```

```
<BODY >
```

```
<H2>ITIS "F.GIORDANI" CASERTA</H2>
```

```
<H3>CONSUNTIVO DIDATTICO 1998 (reale e previsione) DELLA III B  
INFORMATICA</H3>
```

```
<TABLE><TH>I TRIMESTRE</TH>
```

```
<TH>II TRIMESTRE</TH><TH>III TRIMESTRE</TH><TR>
```

```
<TD> 40 % (< 6) reale </TD>
```

```
<TD> 35 % (< 6) reale </TD>
```

```
<TD> 30 % (< 6) reale </TD><TR>
```

```
<TD> 35 % (< 6) previsto </TD>
```

```
<TD> 30 % (< 6) previsto </TD>
```

```
<TD> 25 % (< 6) previsto </TD><TR>
```

```
</TABLE>
```

```
</BODY>
```

```
</HTML>
```

116. HTML, realizzare una pagina contenente una tabella dei voti

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 12/04/2021

Numero progressivo dell'esercizio: es116

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Google Chrome

Obiettivo didattico: Le tabelle (caption TH ALIGN)

Obiettivo del programma: Realizzare una pagina contenente una tabella

```
<HTML><HEAD><TITLE>tabelle</TITLE></HEAD>
<BODY >
<H3 ALIGN="CENTER">ITIS "F.GIORDANI" CASERTA</H3>
<TABLE BORDER=9>
<CAPTION>Prospetto voti di informatica della VBinf. a.s. 98/99</CAPTION>
<TH ALIGN="LEFT">ALLIEVI</TH>
<TH> ORALE</TH>
<TH>SCRITTO</TH>
<TH>PRATICO</TH >
<TR>
<TD ALIGN="LEFT">ROSSI ALBERTO </TD>
<TD ALIGN="CENTER"> 5 </TD>
<TD ALIGN="CENTER"> 6 </TD>
<TD ALIGN="CENTER"> 5 </TD>
</TR>
<TR>
<TD ALIGN="LEFT">ZORZI ENNIO </TD>
<TD ALIGN="CENTER"> 8 </TD>
<TD ALIGN="CENTER"> 9 </TD>
<TD ALIGN="CENTER"> 9 </TD>
```

```
</TR>
</TABLE>
</BODY>
</HTML>
```

117. HTML, realizzare una pagina contenente una tabella dei voti

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 12/04/2021

Numero progressivo dell'esercizio: es117

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Google Chrome

Obiettivo didattico: Le tabelle

Obiettivo del programma: Realizzare una pagina contenente una tabella

```
<HTML><HEAD><TITLE>tabelle</TITLE></HEAD>
<BODY >
<H3 ALIGN="CENTER">ITIS "F.GIORDANI" CASERTA</H3>
<TABLE BORDER>
<CAPTION>Prospetto voti della VBinf. a.s. 98/99</CAPTION>
<TH ALIGN="LEFT">ALLIEVI</TH>
<TH> MATERIA</TH>
<TH> ORALE</TH>
<TH>SCRITTO</TH>
<TH>PRATICO</TH >
<TR>
<TD ROWSPAN =2 ALIGN="LEFT">ROSSI ALBERTO </TD>
<TD ALIGN="CENTER">systemi</TD>
<TD ALIGN="CENTER"> 5 </TD>
<TD ALIGN="CENTER"> 6 </TD>
```



```
<TD ALIGN="CENTER"> 5 </TD>
</TR>
<TR>
<TD ALIGN="CENTER">informatica</TD>
<TD ALIGN="CENTER"> 4 </TD>
<TD ALIGN="CENTER"> 5 </TD>
<TD ALIGN="CENTER"> 5 </TD>
</TR>
<TR>
<TD ROWSPAN=2 ALIGN="LEFT">ZORZI ENNIO </TD>
<TD ALIGN="CENTER">sistemi</TD>
<TD ALIGN="CENTER"> 8 </TD>
<TD ALIGN="CENTER"> 8 </TD>
<TD ALIGN="CENTER"> 7 </TD>
</TR>
<TR>
<TD ALIGN="CENTER">informatica</TD>
<TD ALIGN="CENTER"> 9 </TD>
<TD ALIGN="CENTER"> 7 </TD>
<TD ALIGN="CENTER"> 9 </TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

118. HTML, realizzare una pagina contenente una CheckBox

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 12/04/2021

Numero progressivo dell'esercizio: es118

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Google Chrome

Obiettivo didattico: CheckBox-Radio

Obiettivo del programma: Realizzare una pagina contenente CheckBox-Radio

```
<HTML><HEAD><TITLE>FORM</TITLE></HEAD>
```

```
<BODY >
```

```
<FONT SIZE=1>
```

```
<FORM NAME="PRIMO">(SERIE DI CASELLE DI SELEZIONE)
```

```
  Scegli la destinazione : <BR>
```

```
<INPUT TYPE="CHECKBOX" NAME=LUOGO VALUE="Appartamenti Reali" CHECKED >
```

```
  Appartamenti Reali<BR>
```

```
<INPUT TYPE="CHECKBOX" NAME=LUOGO VALUE="Parco Reale" CHECKED > Parco  
Reale<BR>
```

```
<INPUT TYPE="CHECKBOX" NAME=LUOGO VALUE="Casertavecchia" CHECKED  
>Casertavecchia<BR>
```

```
<INPUT TYPE="CHECKBOX" NAME=LUOGO VALUE="San Leucio" CHECKED > San  
Leucio<BR>
```

```
<INPUT TYPE="CHECKBOX" NAME=LUOGO VALUE="San Pietro ad Montes Casolla"  
CHECKED >
```

```
  San Pietro ad Montes Casolla<BR>
```

```
(SERIE DI CASELLE DI OPZIONE) <BR>
```

```
  Scegli la destinazione (é possibile seglierne una soltanto): <BR>
```

```
<INPUT TYPE="RADIO" NAME=LUOGO VALUE="Appartamenti Reali"> Appartamenti Reali<BR>
```

```
<INPUT TYPE="RADIO" NAME=LUOGO VALUE="Parco Reale"> Parco Reale<BR>
```

```
<INPUT TYPE="RADIO" NAME=LUOGO VALUE="Casertavecchia"> Casertavecchia<BR>
```

```
<INPUT TYPE="RADIO" NAME=LUOGO VALUE="San Leucio" > San Leucio<BR>
```

```
<INPUT TYPE="RADIO" NAME=LUOGO VALUE="San Pietro ad Montes Casolla" > San Pietro ad  
  Montes Casolla<BR>
```

```
<INPUT TYPE="SUBMIT" VALUE="PAGA IL BIGLIETTO">
```

```
<INPUT TYPE="RESET">
```

```
</FORM>
```

```
</BODY>
```

```
</HTML>
```

119. HTML, realizzare una pagina contenente una ComboBox

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 12/04/2021

Numero progressivo dell'esercizio: es119

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Google Chrome

Obiettivo didattico: ComboBox

Obiettivo del programma: Realizzare una pagina contenente una ComboBox

```
<HTML><HEAD><TITLE>FORM</TITLE></HEAD>
```

```
<BODY >
```

```
<FONT SIZE=1>
```

```
<FORM NAME="PRIMO">Scegli la destinazione (ctrl e click per sceglierne piú di una): <BR>
```

```
<SELECT NAME=LUOGO MULTIPLE style="width:500px;">
```

```
<OPTION SELECTED> Appartamenti Reali
```

```
<OPTION> Parco Reale
```

```
<OPTION> Casertavecchia
```

```
<OPTION> San Leucio
```

```
<OPTION> San Pietro ad Montes Casolla
```

```
</SELECT>
```

```
<BR>
```

```
    Scegli la destinazione (é possibile selezionarne una sola): <BR>
```

```
<SELECT NAME=LUOGO >
```

```
<OPTION SELECTED> Appartamenti Reali
```

```
<OPTION> Parco Reale
```

```
<OPTION> Casertavecchia
```

```

<OPTION> San Leucio
<OPTION> San Pietro ad Montes Casolla
</SELECT>
<BR>NOTE: <BR>
<TEXTAREA NAME="note" col=20 rows=5></TEXTAREA>
<BR><BR><BR>
<INPUT TYPE="SUBMIT" VALUE="PAGA IL BIGLIETTO">
<INPUT TYPE="RESET">
</FORM>
</BODY>
</HTML>

```

120. C++, caricare e stampare una matrice

```

/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2019/2020
Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 13/04/2021
Numero progressivo dell'esercizio: es120
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: Google Crome
Obiettivo didattico: Matrici
Obiettivo del programma: Realizzare un programma che carica e stampa una matrice
*/
#include <iostream>
using namespace std;
int dimRig = 2;
int dimCol=3;
int mat[100] [100]; //matrice con due righe e tre colonne di numeri interi
void caricaMat(int matPar[100][100], int dimRigPar, int dimColPar);
void stampaMat(int matPar[100][100], int dimRigPar, int dimColPar);
int main()
{
    caricaMat(mat, dimRig, dimCol);
    stampaMat(mat, dimRig, dimCol);
    return 0;
}
void caricaMat(int matPar[100][100], int dimRigPar, int dimColPar)
{
    for(int i=0;i<dimRigPar;i++)

```

```

    {
        for(int j=0;j<dimColPar;j++)
        {
            cout << "inserisci un numero: ";
            cin >> mat[i][j];
        }
    }
}
void stampaMat(int matPar[100][100], int dimRigPar, int dimColPar)
{
    for(int i=0;i<dimRigPar;i++)
    {
        cout << endl;
        for(int j=0;j<dimColPar;j++)
        {
            cout << mat[i][j] << "\t";
        }
    }
}

```

121. C++, caricare, calcolare la media degli elementi e stampare una matrice

/**

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 13/04/2021

Numero progressivo dell'esercizio: es121

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Google Crome

Obiettivo didattico: Le Matrici

Obiettivo del programma: Realizzare un programma che carica, calcola la media aritmetica tra gli elementi della matrice e la stampa

*/

#include <iostream>

using namespace std;

int dimRig = 2;

int dimCol=3;

int mat[100] [100]; //matrice con due righe e tre colonne di numeri interi

void caricaMat(int matPar[100][100], int dimRigPar, int dimColPar);

void stampaMat(int matPar[100][100], int dimRigPar, int dimColPar);

float media(int matPar[100][100], int dimRigPar, int dimColPar);

int main()

{

caricaMat(mat, dimRig, dimCol);

stampaMat(mat, dimRig, dimCol);

cout << endl;

cout << "media aritmetica dei valori: " << media(mat, dimRig, dimCol);

cout << endl;

```

    return 0;
}
void caricaMat(int matPar[100][100], int dimRigPar, int dimColPar)
{
    for(int i=0;i<dimRigPar;i++)
    {
        for(int j=0;j<dimColPar;j++)
        {
            cout << "inserisci un valore: ";
            cin >> mat[i][j];
        }
    }
}
void stampaMat(int matPar[100][100], int dimRigPar, int dimColPar)
{
    cout << "stampa dei valori: ";
    for(int i=0;i<dimRigPar;i++)
    {
        cout << endl;
        for(int j=0;j<dimColPar;j++)
        {
            cout << mat[i][j] << "\t";
        }
    }
}
float media(int matPar[100][100], int dimRigPar, int dimColPar)
{
    float somma=0,dim=0;
    for(int i=0;i<dimRigPar;i++)
    {
        for(int j=0;j<dimColPar;j++)
        {
            somma = somma + mat[i][j];
            dim++;
        }
    }
    return somma/dim;
}

```

122. C++, data una matrice 3x4 calcolare la somma degli elementi della seconda riga

```

/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2019/2020
Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 16/04/2021
Numero progressivo dell'esercizio: es122
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10

```

Compilatore/Interprete: Code::Blocks 20.03

Obiettivo didattico: Le Matrici

Obiettivo del programma: Data una matrice 3x4 di interi calcolare la somma degli elementi della seconda riga

```
*/  
#include <iostream>  
using namespace std;  
int mat[3][4] = {9,3,1,5,  
                8,4,10,2};  
int dimCol = 4;  
int main()  
{  
    int somma = 0;  
    for(int i=0;i<dimCol;i++)  
    {  
        somma = somma + mat[1][i];  
    }  
    cout << "La somma e' " << somma << endl;  
    return 0;  
}
```

123. C++, data una matrice MxN calcolare la media della colonna scelta

/**

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 16/04/2021

Numero progressivo dell'esercizio: es123

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Code::Blocks 20.03

Obiettivo didattico: Le Matrici

Obiettivo del programma: Data una matrice MxN di interi e un numero di colonna, calcolare la media aritmetica della colonna indicata

```
*/  
#include <iostream>  
using namespace std;  
int const M=100,N=100;  
int dimCol = 3;  
int dimRig = 3;  
int mat[M][N];  
void caricaMat(int matPar[M][N],int dimColPar,int dimRigPar);  
void stampaMat(int matPar[M][N],int dimColPar,int dimRigPar);  
int main()  
{  
    int col,somma=0,media;  
    caricaMat(mat,dimCol,dimRig);  
    stampaMat(mat,dimCol,dimRig);  
}
```

```

cout << endl;
do{
cout << "inserire il numero della colonna di cui si vuole calcolare la media aritmetica: ";
cin >> col;
}while(col>=dimCol || col<0);
for(int i=0;i<dimRig;i++)
{
    somma=somma+mat[i][col];
}
media = somma/(col+1);
cout << "la media aritmetica della colonna numero "<<col<<" e' "<<media<<endl;
return 0;
}
void caricaMat(int matPar[M][N], int dimColPar, int dimRigPar)
{
    for(int i=0;i<dimRigPar;i++)
    {
        for(int j=0;j<dimColPar;j++)
        {
            cout << "inserisci il numero nella posizione " << i << "x" << j << ": ";
            cin >> mat[i][j];
        }
    }
}
void stampaMat(int matPar[M][N], int dimColPar, int dimRigPar)
{
    for(int i=0;i<dimRigPar;i++)
    {
        cout << endl;
        for(int j=0;j<dimColPar;j++)
        {
            cout << mat[i][j] << "\t";
        }
    }
}
}

```

124. C++, dato un vettore trovare il numero massimo all'interno di esso

/**

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 17/04/2021

Numero progressivo dell'esercizio: es124

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Code::Blocks 20.03

Obiettivo didattico: Tecnica del massimo

Obiettivo del programma: Dato un vettore, trovare il numero massimo all'interno di esso


```

*/
#include <iostream>
using namespace std;
int const dimFis=100;
int dimLog;
int vet[dimFis];
void caricaVet(int vetPar[], int dimLogPar);
void stampaVet(int vetPar[], int dimLogPar);
int ottieniDim(int dimFisPar);
int trovaNumeroMax(int vetPar[], int dimLogPar);
int main()
{
    dimLog = ottieniDim(dimFis);
    cout << "caricamento del vettore" << endl;
    caricaVet(vet, dimLog);
    cout << endl;
    cout << "stampa del vettore" << endl;
    stampaVet(vet, dimLog);
    cout << endl;
    cout << endl;
    cout << "il numero massimo inserito e' " << trovaNumeroMax(vet, dimLog) << endl;
    return 0;
}
int ottieniDim(int dimFisPar)
{
    int dim;
    do{
        cout << "Quanti numeri vuoi inserire: ";
        cin >> dim;
    }while(dim > dimFisPar || dim < 0);
    return dim;
}
void caricaVet(int vetPar[], int dimLogPar)
{
    for(int i=0;i<dimLogPar;i++)
    {
        cout << i+1<<" inserisci un numero: ";
        cin >> vetPar[i];
    }
}
void stampaVet(int vetPar[], int dimLogPar)
{
    for(int i=0;i<dimLogPar;i++)
    {
        cout << vetPar[i] << " ";
    }
}
int trovaNumeroMax(int vetPar[], int dimLogPar)
{
    int Max = vetPar[0];
    for(int i=1;i<dimLogPar;i++)
    {
        if(Max < vetPar[i]) Max = vetPar[i];
    }
}

```

```
}  
return Max;  
}
```

125. C++, dati due vettori paralleli trovare la città con temperatura minima

```
/**  
ITIS-LS "Francesco Giordani" Caserta  
Anno scolastico: 2019/2020  
Classe 3^ sez.B spec. Informatica e telecomunicazioni  
Data: 19/04/2021  
Numero progressivo dell'esercizio: es125  
Versione: 1.0  
Programmatore/i: Alessandro Esposito  
Sistema Operativo: Windows 10  
Compilatore/Interprete: Code::Blocks 20.03  
Obiettivo didattico: Tecnica del minimo  
Obiettivo del programma: Dati due vettori paralleli (citta, temperature) trovare la città con temperatura minima  
*/  
#include <iostream>  
#include <string>  
using namespace std;  
int const dimFis = 100;  

```

```

}
int posTempMin(int tempPar[], int dimLogPar)
{
    int min=tempPar[0];
    int pos;
    for(int i=1; i<dimLogPar; i++)
    {
        if(min>tempPar[i])
        {
            min=tempPar[i];
            pos = i;
        }
    }
    return pos;
}

```

126. C++, gestire i voti si una classe

```

/*
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2020/2021
Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 26/4/2021
Numero progressivo dell'esercizio: es126
Versione: 1.0
Programmatore/i: Lezione collettiva
Sistema Operativo: Windows 10
Compilatore/Interprete: C++
Obiettivo didattico: costruire un modello dei dati e
                    scegliere opportunamente le relative
                    strutture dati
Obiettivo del programma: gestire i voti di una classe
Modello dei dati: nomiAlunni, materie, voti
Strutture dati: vettoreAlunni, vettoreMaterie, matriceVoti
Sottoprogrammi: caricaAlunni, caricaMaterie, caricaVoti, ricerca,
                stampaVotiDellaClasse, stampaVotiDiUnAlunno,
                stampaVotiDiUnaMateria,
                stampaVotoDiUnAlunnoInUnaMateria
*/
#include <iostream>
#include <string>
using namespace std;
int const numMaxAlunni=30;
int const numMaxMaterie=15;
//Strutture dati
string vetAlunni[numMaxAlunni];
string vetMaterie[numMaxMaterie];
int matVoti[numMaxAlunni][numMaxMaterie];
//Sottoprogrammi

```

```

void caricaAlunni(string vetAlunniPar[],int numAlunniPar);
void caricaMaterie(string vetMateriePar[],int numMateriePar);
void caricaVoti(int matVotiPar[numMaxAlunni][numMaxMaterie],
               int numAlunniPar,int numMateriePar);
int ricerca(string vetPar[],int dimLogPar,string kPar);
void stampaVotiClasse(string vetAlunniPar[],string vetMateriePat[],
                    int matVotiPar[numMaxAlunni][numMaxMaterie],
                    int numAlunniPar,int numMateriePar);
void stampaVotiUnaMateria(string vetMateriePar[],string vetAlunniPar[],
                        int matVotiPar[numMaxAlunni][numMaxMaterie],
                        int numAlunniPar,int posPar);
void stampaVotiUnAlunno(string vetMateriePar[],string vetAlunniPar[],
                      int numMateriePar,
                      int matVotiPar[numMaxAlunni][numMaxMaterie],int posPar);
int stampaVotiUnAlunnoUnaMateria(int matVotiPar[numMaxAlunni][numMaxMaterie],
                                int posPar, int pos2Par);

int main()
{
    int numAlunni, numMaterie,pos,pos2;
    string k,k2;
    do{
        cout << "inserire numero degli alunni(compreso tra 0 e 30): ";
        cin >> numAlunni;
    }while(numAlunni<0 || numAlunni>numMaxAlunni);
    do{
        cout << "inserire numero delle materie: ";
        cin >> numMaterie;
    }while(numMaterie<0 || numMaterie>numMaxMaterie);
    cout << "CARICA ALUNNI" << endl;
    caricaAlunni(vetAlunni, numAlunni);
    cout << "CARICA MATERIE" << endl;
    caricaMaterie(vetMaterie,numMaterie);
    cout << "CARICA VOTI" << endl;
    caricaVoti(matVoti, numAlunni, numMaterie);
    cout << "STAMPA DELLA TABELLA DEI VOTI" << endl;
    stampaVotiClasse(vetAlunni,vetMaterie,
                    matVoti,numAlunni,numMaterie);
    cout << endl;
    cout << "inserire il nome dell'alunno di cui si vuole stampare i voti: ";
    cin >> k;
    pos=ricerca(vetAlunni,numAlunni,k);
    if(pos==-1) cout << "L'alunno e' inesistente";
    else stampaVotiUnAlunno(vetMaterie,vetAlunni,
                            numMaterie,matVoti,pos);
    cout << endl;
    cout << "inserire il nome della materia di cui si vuole stampare i voti: ";
    cin >> k;
    pos=ricerca(vetMaterie,numMaterie,k);
    if(pos==-1) cout << "la materia è inesistente"
    else stampaVotiUnaMateria(vetMaterie, vetAlunni, matVoti, numAlunni,pos);
    cout << endl;
    cout << "inserire il nome dell'alunno di cui si vuole stampare i voti: ";
    cin >> k;
}

```

```

cout << "inserire il nome della materia di cui si vuole stampare i voti: ";
cin >> k2;
pos = ricerca(vetAlunni,numAlunni,k);
pos2 = ricerca(vetMaterie,numMaterie,k2);
if(pos==-1 || pos2==-1) cout << "L'alunno o la materia inserita sono inesistenti";
else cout << "voto dell'alunno " <<k<<" nella materia " <<k2<<":
"<<stampaVotiUnAlunnoUnaMateria(matVoti,pos,pos2);
return 0;
}
void caricaAlunni(string vetAlunniPar[],int numAlunniPar)
{
for(int i=0;i<numAlunniPar;i++)
{
cout << "inserire il nome del " <<i+1<<"^ alunno: ";
cin >> vetAlunniPar[i];
}
}
void caricaMaterie(string vetMateriePar[],int numMateriePar)
{
for(int i=0;i<numMateriePar;i++)
{
cout << "inserire il nome della " <<i+1<<"^ materia: ";
cin >> vetMateriePar[i];
}
}
void caricaVoti(int matVotiPar[numMaxAlunni][numMaxMaterie],int numAlunniPar,int numMateriePar)
{
for(int i=0;i<=numAlunniPar;i++)
{
for(int j=0;j<numMateriePar;j++)
{
cout << "inserire il voto del " <<i+1<<"^ alunno sulla " <<j+1<<"^ materia: ";
cin >> matVotiPar[i][j];
}
}
}
void stampaVotiClasse(string vetAlunniPar[],string vetMateriePar[],
int matVotiPar[numMaxAlunni][numMaxMaterie],
int numAlunniPar,int numMateriePar)
{
cout << " ";
for(int i=0;i<=numAlunniPar;i++)
{
for(int j=0;j<numMateriePar;j++)
{
if(i==0) cout << vetMateriePar[j] << " ";
if(i!= 0) cout << matVotiPar[i][j] << " ";
}
cout << endl;
cout << vetAlunniPar[i] << "\t";
}
}
int ricerca(string vetPar[],int dimLogPar,string kPar)

```

```

{
    int trovato = -1,i=0;
    do{
        if(vetPar[i]==kPar) trovato=i;
        else i++;
    }while(i<dimLogPar && trovato==-1);
    return trovato;
}
void stampaVotiUnAlunno(string vetMateriePar[],string vetAlunniPar[],
                        int numMateriePar,
                        int matVotiPar[numMaxAlunni][numMaxMaterie],int posPar)
{
    cout << "VOTI DELL'ALUNNO: "<<vetAlunniPar[posPar]<<endl;
    for(int i=0;i<numMateriePar;i++)
    {
        cout <<vetMateriePar[i]<<": "<<matVotiPar[posPar][i];
    }
}
void stampaVotiUnaMateria(string vetMateriePar[],string vetAlunniPar[],
                        int matVotiPar[numMaxAlunni][numMaxMaterie],
                        int numAlunniPar,int posPar)
{
    cout << "VOTI DELLA MATERIA: "<<vetMateriePar[posPar]<<endl;
    for(int i=0;i<=numAlunniPar;i++)
    {
        cout <<vetAlunniPar[i]<<": "<<matVotiPar[posPar][i];
    }
}
int stampaVotiUnAlunnoUnaMateria(int matVotiPar[numMaxAlunni][numMaxMaterie],int posPar, int pos2
Par)
{
    return matVotiPar[posPar][pos2Par];
}

```

127. JavaScript, scrivere Ciao Mondo in una pagina web

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 23/04/2021

Numero progressivo dell'esercizio: es127

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Google Chrome

Obiettivo didattico: Conoscere l'ambiente di programmazione

Obiettivo del programma: Scrivere Ciao Mondo in una pagina web

<html>

<head>

```
<script type="text/javascript">
{
alert("Ciao Mondo");
}
</script>
</head>
<body></body>
</html>
```

128. JavaScript, scrivere nella pagina web la scelta effettuata dall'utente

Anno scolastico: 2019/2020

Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 23/04/2021

Numero progressivo dell'esercizio: es128

Versione: 1.0

Programmatore/i:

Sistema Operativo: Windows 10

Compilatore/Interprete: Google Chrome

Obiettivo didattico: Alert, Confirm, document.write. La selezione: if else

Obiettivo del programma: Scrivere nella pagina web la scelta effettuata dall'utente

```
<html>
```

```
<head>
```

```
<script type="text/javascript">
```

```
{
```

```
alert("Benvenuto");
```

```
var dato=confirm("scegli ok/annulla ");
```

```
if (dato==true) // controllo della conferma
```

```
{document.write("Hai cliccato OK")}
```

```
else {document.write("Hai cancellato")}
```

```
}
```

```
</script>
```

```
</head>
```

```
<body></body>
```

```
</html>
```

129. JavaScript, scrivere nella pagina web la stringa inserita dall'utente

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 23/04/2021

Numero progressivo dell'esercizio: es129

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Google Chrome

Obiettivo didattico: Prompt La selezione: if

Obiettivo del programma: Scrivere nella pagina web la stringa inserita dall'utente

```
<html>
<head>
<script type="text/javascript">
var dato = prompt("inserisci");
if (dato!=null && dato !="") // controllo del dato
{document.write("Hai inserito "+dato)}
</script>
</head>
<body></body>
</html>
```

130. C++, Sommare gli elementi della diagonale principale di una matrice quadrata

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2019/2020
Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 24/04/2021
Numero progressivo dell'esercizio: es130
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: Code::Blocks 20.03
Obiettivo didattico: Le Matrici
Obiettivo del programma: Sommare gli elementi della diagonale principale di una matrice quadrata
*/
#include <iostream>
using namespace std;;
int const dimFis=100;
int dimRig=3;
int dimCol=3;
int mat[dimFis][dimFis];
void caricaMat(int matPar[dimFis][dimFis], int dimRigPar, int dimColPar);
void stampaMat(int matPar[dimFis][dimFis], int dimRigPar, int dimColPar);
int sommaDiagPrinc(int matPar[dimFis][dimFis], int dimRigPar, int dimColPar);
int main()
{
    caricaMat(mat, dimRig, dimCol);
    stampaMat(mat, dimRig, dimCol);
    cout << endl;
    cout << "Somma della diagonale principale: " << sommaDiagPrinc(mat, dimRig, dimCol);
    cout << endl;
    return 0;
}
void caricaMat(int matPar[dimFis][dimFis], int dimRigPar, int dimColPar)
{
```



```

for(int i=0;i<dimRigPar;i++)
{
    for(int j=0;j<dimColPar;j++)
    {
        cout << "inserisci un valore nella posizione " << i << "x" << j << ": ";
        cin >> mat[i][j];
    }
}
}
void stampaMat(int matPar[dimFis][dimFis], int dimRigPar, int dimColPar)
{
    cout << "stampa dei valori: ";
    for(int i=0;i<dimRigPar;i++)
    {
        cout << endl;
        for(int j=0;j<dimColPar;j++)
        {
            cout << mat[i][j] << "\t";
        }
    }
}
int sommaDiagPrinc(int matPar[dimFis][dimFis], int dimRigPar, int dimColPar)
{
    int somma=0;
    for(int i=0;i<dimRigPar;i++)
    {
        for(int j=0;j<dimColPar;j++)
        {
            if(i==j) somma = matPar[i][j]+somma;
        }
    }
    return somma;
}

```

131. C++, Sommare gli elementi della diagonale secondaria di una matrice quadrata

```

/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2019/2020
Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 24/04/2021
Numero progressivo dell'esercizio: es131
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: Code::Blocks 20.03
Obiettivo didattico: Le Matrici
Obiettivo del programma: Sommare gli elementi della diagonale secondaria di una matrice quadrata
*/
#include <iostream>

```

```

using namespace std;;
int const dimFis=100;
int dimRig=3;
int dimCol=3;
int mat[dimFis][dimFis];
void caricaMat(int matPar[dimFis][dimFis], int dimRigPar, int dimColPar);
void stampaMat(int matPar[dimFis][dimFis], int dimRigPar, int dimColPar);
int sommaDiagSec(int matPar[dimFis][dimFis], int dimRigPar, int dimColPar);
int main()
{
    caricaMat(mat, dimRig, dimCol);
    stampaMat(mat, dimRig, dimCol);
    cout << endl;
    cout << "Somma della diagonale secondaria: " << sommaDiagSec(mat, dimRig, dimCol);
    cout << endl;
    return 0;
}
void caricaMat(int matPar[dimFis][dimFis], int dimRigPar, int dimColPar)
{
    for(int i=0;i<dimRigPar;i++)
    {
        for(int j=0;j<dimColPar;j++)
        {
            cout << "inserisci un valore nella posizione " << i << "x" << j << ": ";
            cin >> mat[i][j];
        }
    }
}
void stampaMat(int matPar[dimFis][dimFis], int dimRigPar, int dimColPar)
{
    cout << "stampa dei valori: ";
    for(int i=0;i<dimRigPar;i++)
    {
        cout << endl;
        for(int j=0;j<dimColPar;j++)
        {
            cout << mat[i][j] << "\t";
        }
    }
}
int sommaDiagSec(int matPar[dimFis][dimFis], int dimRigPar, int dimColPar)
{
    int somma=0,j=dimColPar;
    for(int i=0;i<dimRigPar;i++)
    {
        j--;
        somma = somma + matPar[i][j];
    }
    return somma;
}

```

132. C++, dati due record visualizzare chi è la persona con età maggiore

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2019/2020
Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 03/05/2021
Numero progressivo dell'esercizio: es132
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: Code::Blocks 20.03
Obiettivo didattico: I Record
Obiettivo del programma: Dati due record con formato nome cognome età visualizzare chi è la persona con
età maggiore
*/
#include <iostream>
using namespace std;
struct personaType
{
    string nome;
    string cognome;
    int eta;
};
void caricaRec();
void stampaRec();
void comparaEta(int eta1,int eta2);
personaType persona1, persona2;
int main()
{
    caricaRec();
    cout << endl;
    stampaRec();
    cout << endl;
    comparaEta(persona1.eta,persona2.eta);
    cout << endl;
    return 0;
}
void caricaRec()
{
    cout << "inserisci il nome della prima persona: ";
    cin >> persona1.nome;
    cout << "inserisci il cognome della prima persona: ";
    cin >> persona1.cognome;
    cout << "inserisci l'eta' della prima persona: ";
    cin >> persona1.eta;
    cout << "inserisci il nome della seconda persona: ";
    cin >> persona2.nome;
    cout << "inserisci il cognome della seconda persona: ";
    cin >> persona2.cognome;
    cout << "inserisci l'eta' della seconda persona: ";
```

```

    cin >> persona2.eta;
}
void stampaRec()
{
    cout << "Nome della prima persona: " << persona1.nome << endl;
    cout << "Cognome della prima persona: " << persona1.cognome << endl;
    cout << "Eta' della prima persona: " << persona1.eta << endl;
    cout << "Nome della seconda persona: " << persona2.nome << endl;
    cout << "Cognome della seconda persona: " << persona2.cognome << endl;
    cout << "Eta' della seconda persona: " << persona2.eta << endl;
}
void comparaEta(int eta1,int eta2)
{
    if(eta1<eta2) cout << persona2.nome <<" e' piu' grande di "<< persona1.nome;
    else cout << persona1.nome <<" e' piu' grande di "<< persona2.nome;
}

```

133. C++, dati tre record visualizzare in ordine crescente rispetto l'anno e auto che costa meno

```

/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2019/2020
Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 03/05/2021
Numero progressivo dell'esercizio: es133
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: Code::Blocks 20.03
Obiettivo didattico: I Record
Obiettivo del programma: Dati tre records con il seguente formato: modello, marca, anno di
immatricolazione,
prezzo. Visualizzare le tre auto in ordine crescente rispetto all'anno di immatricolazione e auto che costa
di meno
*/
#include <iostream>
using namespace std;
struct autoType
{
    string modello;
    string marca;
    int anno;
    float prezzo;
};
int main()
{
    autoType auto1, auto2, auto3;
    auto1.modello = "500";
    auto1.marca = "Fiat";

```

```
auto1.anno = 2012;
auto1.prezzo = 8000;
```

```
auto2.modello = "Model S";
auto2.marca = "Tesla";
auto2.anno = 2021;
auto2.prezzo = 100000;
```

```
auto3.modello = "Yaris";
auto3.marca = "Toyota";
auto3.anno = 2020;
auto3.prezzo = 16000;
```

```
int prezzoMin = auto1.prezzo;
string autoMenoCostosa = auto1.modello;
if(auto2.prezzo < prezzoMin)
{
    prezzoMin = auto2.prezzo;
    autoMenoCostosa = auto2.modello;
}
if(auto3.prezzo < prezzoMin)
{
    prezzoMin = auto3.prezzo;
    autoMenoCostosa = auto3.modello;
}
```

```
cout << "L'auto meno costosa risulta: " << autoMenoCostosa << " con il prezzo: "
<< prezzoMin << endl;
```

```
int a1=auto1.anno;
int a2=auto2.anno;
int a3=auto3.anno;
string a1Model=auto1.modello;
string a2Model=auto2.modello;
string a3Model=auto3.modello;
int app;
int vet[3]={a1,a2,a3};
string vetModel[3]={a1Model,a2Model,a3Model};
string app2;
for(int i=0;i<2;i++)
    for(int j=i+1;j<3;j++)
        if(vet[i]>vet[j])
        {
            app = vet[i];
            vet[i] = vet[j];
            vet[j] = app;
            app2 = vetModel[i];
            vetModel[i] = vetModel[j];
            vetModel[j] = app2;
        }
for(int i=0;i<3;i++)
    cout << vetModel[i] << " " << vet[i] << "\t";
```

```
    return 0;
}
```

134. C++, caricare e stampare a tabella un vettore di array con le procedure

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2019/2020
Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 07/05/2021
Numero progressivo dell'esercizio: es134
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: Code::Blocks 20.03
Obiettivo didattico: i vettori di array
Obiettivo del programma: Caricare e stampare a tabella un vettore di array con le procedure
*/
#include <iostream>
using namespace std;
struct personaType
{
    string nome;
    string cognome;
    int eta;
};
personaType vet[10];
void caricaVet(int dimLogPar);
void stampaVet(int dimLogPar);
int main()
{
    int dimLog = 3;
    caricaVet(dimLog);
    cout << endl;
    stampaVet(dimLog);
    return 0;
}
void caricaVet(int dimLogPar)
{
    for(int i=0;i<dimLogPar;i++)
    {
        cout << "inserire il nome della "<<i+1<<"^ persona: ";
        cin >> vet[i].nome;
        cout << "inserire il cognome della "<<i+1<<"^ persona: ";
        cin >> vet[i].cognome;
        cout << "inserire l'eta' della "<<i+1<<"^ persona: ";
        cin >> vet[i].eta;
        cout << " " << endl;
    }
}
```

```

void stampaVet(int dimLogPar)
{
    cout << "    " << " P1 " << " P2" << " P3" << endl;
    for(int i=0;i<dimLogPar;i++)
    {
        cout << "Nome: ";
        for(int j=0;j<dimLogPar;j++)
        {
            cout << vet[j].nome << " ";
        }
        cout << endl;
        cout << "Cognome: ";
        for(int j=0;j<dimLogPar;j++)
        {
            cout << vet[j].cognome << " ";
        }
        cout << endl;
        cout << "Eta': ";
        for(int j=0;j<dimLogPar;j++)
        {
            cout << vet[j].eta << " ";
        }
        cout << endl;
    }
}

```

135. C++, calcolare la media delle altezze di un array di record

```

/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2019/2020
Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 10/05/2021
Numero progressivo dell'esercizio: es135
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: Code::Blocks 20.03
Obiettivo didattico: i vettori di array
Obiettivo del programma: Calcolare la media delle altezze in un array di record con il seguente formato:
nome, altezza, età
*/
#include <iostream>
#include <string>
using namespace std;
struct personaType
{
    string nome;
    float altezza;

```

```

    int eta;
};
personaType personaRec;
personaType personaVet[10];
int dimLog=3;
void caricaPersonaVet(personaType personaVetPar[],int dimLogPar);
void stampaPersonaVet(personaType personaVetPar[],int dimLogPar);
float mediaAltezza(personaType personaVetPar[],int dimLogPar);
int main()
{
    caricaPersonaVet(personaVet,dimLog);
    stampaPersonaVet(personaVet,dimLog);
    cout<<endl<<"La media delle altezze e': "<<mediaAltezza(personaVet,dimLog);
    return 0;
}
void caricaPersonaVet(personaType personaVetPar[],int dimLogPar)
{
    for(int i=0;i<dimLogPar;i++)
    {
        cout<<"Inserisci il nome: ";
        cin>>personaVetPar[i].nome;
        cout<<"Inserisci l'altezza: ";
        cin>>personaVetPar[i].altezza;
        cout<<"Inserisci l'eta': ";
        cin>>personaVetPar[i].eta;
        cout<<endl;
    }
}
void stampaPersonaVet(personaType personaVetPar[],int dimLogPar)
{
    cout<<"Stampa dell'elenco: "<<endl;
    cout<<"NOME"<<"\t"<<"ALTEZZA"<<"\t"<<"ETA"<<endl;
    for(int i=0;i<dimLogPar;i++)
    {
        cout<<personaVetPar[i].nome<<"\t"<<personaVetPar[i].altezza<<"\t"<<personaVetPar[i].eta<<endl;
    }
}
float mediaAltezza(personaType personaVetPar[],int dimLogPar)
{
    somma=0;
    for(int i=0;i<dimLogPar;i++)
    {
        somma=somma+personaVetPar[i].altezza;
    }
    return somma/dimLogPar;
}

```

136. C++, eseguire lo shift a sinistra di un vettore di caratteri


```

/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2019/2020
Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 11/05/2021
Numero progressivo dell'esercizio: es136
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: Code::Blocks 20.03
Obiettivo didattico: shift di un vettore
Obiettivo del programma: eseguire lo shift a sinistra di un vettore di caratteri
*/
#include <iostream>
using namespace std;
int dimlog=5;
char vet[150];
void caricaVet(char vetPar[],int dimlogPar);
void shiftsx(char vetPar[],int dimlogPar);
void stampaVet(char vetPar[],int dimlogPar);
int main()
{
    cout << "caricamento vettore:" << endl;
    caricaVet(vet,dimlog);
    cout << "stampa vettore:" << endl;
    stampaVet(vet,dimlog);
    cout<<"stampa vettore dopo lo shift sx:"<<endl;
    shiftsx(vet,dimlog);
    stampaVetchar(vet,dimlog);
    return 0;
}
void caricaVet(char vetPar[],int dimlogPar)
{
    for(int i=0;i<dimlogPar;i++)
    {
        cout<<"inserire il "<<i+1<<" carattere:";
        cin>>vetPar[i];
    }
}
void stampaVet(char vetPar[],int dimlogPar)
{
    for(int i=0;i<dimlogPar;i++)
    {
        cout<<vetPar[i]<<"\t";
    }
}
void shiftsx(char vetPar[],int dimlogPar)
{
    for(int i=0;i<dimlogPar-1;i++)
    {
        vetPar[i]=vetPar[i+1];
    }
}

```

137. C++, eseguire lo shift a destra di un vettore di caratteri

```
/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2019/2020
Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 13/05/2021
Numero progressivo dell'esercizio: es137
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: Code::Blocks 20.03
Obiettivo didattico: shift di un vettore
Obiettivo del programma: eseguire lo shift a destra di un vettore di caratteri
*/
#include <iostream>
using namespace std;
int dimlog=5;
char vet[150];
void caricaVet(char vetPar[],int dimlogPar);
void shiftdx(char vetPar[],int dimlogPar);
void stampaVet(char vetPar[],int dimlogPar);
int main()
{
    cout << "caricamento vettore:" << endl;
    caricaVet(vet,dimlog);
    cout << "stampa vettore:" << endl;
    stampaVet(vet,dimlog);
    cout<<"stampa vettore dopo lo shift dx:"<<endl;
    shifdx(vet,dimlog);
    stampaVetchar(vet,dimlog);
    return 0;
}
void caricaVet(char vetPar[],int dimlogPar)
{
    for(int i=0;i<dimlogPar;i++)
    {
        cout<<"inserire il "<<i+1<<" carattere:";
        cin>>vetPar[i];
    }
}
void stampaVet(char vetPar[],int dimlogPar)
{
    for(int i=0;i<dimlogPar;i++)
    {
        cout<<vetPar[i]<<"\t";
    }
}
void shiftdx(char vetPar[],int dimlogPar)
{
    for(int i=dimlogPar;i>1;i--)
```

```

    {
        vetPar[i]=vetPar[i-1];
    }
}

```

138. C++, contare il numero di vocali per ogni riga di una matrice

```

/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2019/2020
Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 14/05/2021
Numero progressivo dell'esercizio: es138
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: Code::Blocks 20.03
Obiettivo didattico: L'alunno è in grado di utilizzare le matrici
Obiettivo del programma: contare il numero di vocali per ogni riga di una matrice
*/
#include <iostream>
using namespace std;
int const dimRig = 3;
int const dimCol = 2;
char mat[dimRig][dimCol]={'a','b',
                          'c','d',
                          'e','f'};

int main()
{
    int contaVoc=0;
    for(int i=0;i<dimRig;i++)
    {
        for(int j=0;j<dimCol;j++)
        {
            if(mat[i][j]=='a' || mat[i][j]=='e' || mat[i][j]=='i' || mat[i][j]=='o' || mat[i][j]=='u') contaVoc++;
        }
        cout << "nella " << i+1 << " riga ci sono " << contaVoc << " vocali" << endl;
        contaVoc=0;
    }
    return 0;
}

```

139. C++, ordinare in modo crescente ogni colonna di una matrice N*M

```

/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2019/2020

```

Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 17/05/2021

Numero progressivo dell'esercizio: es139

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Code::Blocks 20.03

Obiettivo didattico: L'alunno è in grado di utilizzare le matrici

Obiettivo del programma: ordinare in modo crescente ogni colonna di una matrice di interi $N \times M$

```
*/
#include <iostream>
using namespace std;
int const dimRig=3;
int const dimCol=2;
int mat[dimRig][dimCol];
void caricaMat(int matPar[dimRig][dimCol], int dimRigPar, int dimColPar);
void stampaMat(int matPar[dimRig][dimCol], int dimRigPar, int dimColPar);
void ordinaMat(int matPar[dimRig][dimCol], int dimRigPar, int dimColPar);
void scambia(int unoPar, int duePar);
int main()
{
    caricaMat(mat,dimRig,dimCol);
    ordinaMat(mat,dimRig,dimCol);
    stampaMat(mat,dimRig,dimCol);
    return 0;
}
void caricaMat(int matPar[dimRig][dimCol], int dimRigPar, int dimColPar)
{
    for(int i=0;i<dimRigPar;i++)
    {
        for(int j=0;j<dimColPar;j++)
        {
            cout << "inserisci un numero: ";
            cin >> mat[i][j];
        }
    }
}
void stampaMat(int matPar[dimRig][dimCol], int dimRigPar, int dimColPar)
{
    for(int i=0;i<dimRigPar;i++)
    {
        cout << endl;
        for(int j=0;j<dimColPar;j++)
        {
            cout << mat[i][j] << " ";
        }
    }
}
void ordinaMat(int matPar[dimRig][dimCol], int dimRigPar, int dimColPar)
{
    for(int i=0;i<dimRigPar;i++)
    {
        for(int k=0;k<dimColPar-1;k++)
```

```

    {
        for(int j=0;j<dimColPar;j++)
        {
            if(matPar[i][k]>matPar[i][j])
            {
                scambia(matPar[i][k],matPar[i][j]);
            }
        }
    }
}
}
void scambia(int unoPar, int duePar)
{
    int app;
    app=unoPar;
    unoPar=duePar;
    duePar=app;
}

```

140. C++, Sommare gli elementi dispari di ogni riga di una matrice N*M

```

/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2019/2020
Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 17/05/2021
Numero progressivo dell'esercizio: es140
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: Code::Blocks 20.03
Obiettivo didattico: L'alunno è in grado di utilizzare le matrici
Obiettivo del programma: Sommare gli elementi dispari di ogni riga di una matrice N*M
*/
#include <iostream>
using namespace std;
int const dimRig = 3;
int const dimCol = 2;
int mat[dimRig][dimCol];
void caricaMat(int matPar[dimRig][dimCol], int dimRigPar, int dimColPar);
void sommaDispMat(int matPar[dimRig][dimCol], int dimRigPar, int dimColPar);
int main()
{
    caricaMat(mat,dimRig,dimCol);
    sommaDispMat(mat,dimRig,dimCol);
    return 0;
}
void caricaMat(int matPar[dimRig][dimCol], int dimRigPar, int dimColPar)
{
    for(int i=0;i<dimRigPar;i++)

```

```

    {
        for(int j=0;j<dimColPar;j++)
        {
            cout << "inserisci un numero: ";
            cin >> mat[i][j];
        }
    }
}
void sommaDispMat(int matPar[dimRig][dimCol], int dimRigPar, int dimColPar)
{
    int somma=0;
    for(int i=0;i<dimRig;i++)
    {
        for(int j=0;j<dimCol;j++)
        {
            if(mat[i][j]%2==1)
            {
                somma=somma+mat[i][j];
            }
        }
    }
    cout << "nella "<<i+1<<"^ riga la somma dei numeri dispari e' "<< somma << endl;
    somma=0;
}
}

```

141. JavaScript, Stampare una matrice in JavaScript

```

/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2019/2020
Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 17/05/2021
Numero progressivo dell'esercizio: es141
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: Code::Blocks 20.03
Obiettivo didattico: L'alunno è in grado di utilizzare le matrici in JavaScript
Obiettivo del programma: Stampare una matrice in JavaScript
*/
<html>
<head>
<script type="text/javascript">
var matrice = new Array();
matrice[0] = new Array();
matrice[1] = new Array();
matrice[0][0] = 1;
matrice[0][1] = 2;
matrice[0][2] = 3;
matrice[1][0] = 4;
matrice[0][1] = 5;
matrice[0][2] = 6;

```

```

document.write("<table>");
for(var x=0;x<matrice[0].length;x++)
{
for(var y=0;y<matrice[1].length;y++)
{
document.write("<td>" + matrice[x][y] + "<td>");
}
document.write("<tr>");
}
document.write("<table>");
</script>
</head>
<body>
</body>
</html>

```

142. C++, Contare i numeri dispari di una matrice N*M di interi in C++

```

/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2019/2020
Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 21/05/2021
Numero progressivo dell'esercizio: es142
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: Python 3.8.1
Obiettivo didattico: L'alunno è in grado di utilizzare le matrici in C++
Obiettivo del programma: Contare i numeri dispari di una matrice N*M di interi
*/
#include <iostream>
using namespace std;
int const dimRig = 3;
int const dimCol = 2;
int mat[dimRig][dimCol];
void caricaMat(int matPar[dimRig][dimCol], int dimRigPar, int dimColPar);
int contaDisp(int matPar[dimRig][dimCol], int dimRigPar, int dimColPar);
int main()
{
caricaMat(mat,dimRig,dimCol);
cout << "La matrice contiene " << contaDisp << " numeri dispari";
return 0;
}
void caricaMat(int matPar[dimRig][dimCol], int dimRigPar, int dimColPar)
{
for(int i=0;i<dimRigPar;i++)
{
for(int j=0;j<dimColPar;j++)
{

```

```

        cout << "inserisci un numero: ";
        cin >> mat[i][j];
    }
}
}
int contaDisp(int matPar[dimRig][dimCol], int dimRigPar, int dimColPar)
{
    int contatore=0;
    for(int i=0;i<dimRigPar;i++)
    {
        for(int j=0;j<dimColPar;j++)
        {
            if(mat[i][j]%2==1) contatore++;
        }
    }
    return contatore;
}

```

143. Python, Contare i numeri dispari di una matrice N*M di interi in Python

```

'''
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2019/2020
Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 21/05/2021
Numero progressivo dell'esercizio: es143
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: Python 3.8.1
Obiettivo didattico: L'alunno è in grado di utilizzare le matrici in Python
Obiettivo del programma: Contare i numeri dispari di una matrice N*M di interi
'''
mat = [
    [0,0,0],
    [0,0,0],
]
dimRig=2
dimCol=3
def caricaMat(matPar,dimRigPar,dimColPar):
    for i in range(dimRigPar):
        for j in range(dimColPar):
            matPar[i][j]=int(input("inserire un numero: "))
def stampaMat(matPar,dimRigPar,dimColPar):
    for i in range(dimRigPar):
        print("\t")
        for j in range(dimColPar):
            print(str(matPar[i][j])+" ",end="")
def contaDisp(matPar,dimRigPar,dimColPar):
    contatore=0

```



```

for i in range(dimRigPar):
    for j in range(dimColPar):
        if matPar[i][j]%2==1:
            contatore += 1
    return contatore
print("CARICAMENTO DELLA MATRICE: ")
caricaMat(mat,dimRig,dimCol)
print("")
print("STAMPA DELLA MATRICE: ",end="")
stampaMat(mat,dimRig,dimCol)
dispari = contaDisp(mat,dimRig,dimCol)
print("")
print("hai inserito "+str(dispari)+" numeri dispari")

```

144. JavaScript, Contare i numeri dispari di una matrice N*M di interi in JavaScript

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 21/05/2021

Numero progressivo dell'esercizio: es144

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Google Chrome

Obiettivo didattico: L'alunno è in grado di utilizzare le matrici in JavaScript

Obiettivo del programma: Contare i numeri dispari di una matrice N*M di interi in JavaScript

```

<html>
<head>
<script type="text/javascript">
mat = [[1,2,3],[4,5,6]]
var contatore = 0;
for(int i=0;i<dimRig;i++)
{
for(int j=0;j<dimCol;j++)
{
if (mat[i][j]%2==1)
{
Contatore++;
}
}
}
document.write("La matrice contiene ",contatore," numeri dispari");
</script>
</head>
<body></body>
</html>

```

145. C++, Sommare gli elementi la cui somma degli indici è pari

```

/**
ITIS-LS "Francesco Giordani" Caserta
Anno scolastico: 2019/2020
Classe 3^ sez.B spec. Informatica e telecomunicazioni
Data: 24/05/2021
Numero progressivo dell'esercizio: es145
Versione: 1.0
Programmatore/i: Alessandro Esposito
Sistema Operativo: Windows 10
Compilatore/Interprete: Python 3.8.1
Obiettivo didattico: L'alunno è in grado di utilizzare le matrici in C++
Obiettivo del programma: Sommare gli elementi la cui somma degli indici è pari
*/
#include <iostream>
using namespace std;
int const dimFis=50;
int dimRig = 3;
int dimCol = 2;
int mat[dimFis][dimFis];
void caricaMat(int matPar[dimFis][dimFis], int dimRigPar, int dimColPar);
void stampaMat(int matPar[dimFis][dimFis], int dimRigPar, int dimColPar);
int sommaPari(int matPar[dimFis][dimFis], int dimRigPar, int dimColPar);
int main()
{
caricaMat(mat,dimRig,dimCol);
stampaMat(mat,dimRig,dimCol);
cout << "La somma degli elementi la cui somma degli indici è pari e': "<<sommaPari(mat,dimRig,dimCol);
return 0;
}
void caricaMat(int matPar[dimFis][dimFis], int dimRigPar, int dimColPar)
{
for(int i=0;i<dimRigPar;i++)
{
for(int j=0;j<dimColPar;j++)
{
cout << "inserisci un numero: ";
cin >> mat[i][j];
}
}
}
void stampaMat(int matPar[dimFis][dimFis], int dimRigPar, int dimColPar)
{
for(int i=0;i<dimRigPar;i++)
{
for(int j=0;j<dimColPar;j++)
{
cout << mat[i][j] << " ";
}
cout << endl;
}
}
int sommaPari(int matPar[dimFis][dimFis], int dimRigPar, int dimColPar)
{

```

```

int somma=0;
for(int i=0;i<dimRigPar;i++)
{
    for(int j=0;j<dimColPar;j++)
    {
        If(mat[i][j]%2==0) somma = mat[i][j]+somma;
    }
}
return somma;
}

```

146. C++, Data una matrice di record trovare il prezzo massimo della terza riga

/**

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3^a sez.B spec. Informatica e telecomunicazioni

Data: 26/05/2021

Numero progressivo dell'esercizio: es146

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Python 3.8.1

Obiettivo didattico: L'alunno è in grado di utilizzare le matrici in C++

Obiettivo del programma: Data una matrice di record con il seguente formato: codArt, prezzo. Trovare il prezzo massimo della terza riga

*/

```
#include <iostream>
```

```
using namespace std;
```

```
struct articoloType
```

```
{
```

```
    string codArt;
```

```
    float prezzo;
```

```
};
```

```
int dimFis=100;
```

```
int dimRig=3;
```

```

int dimCol=2;
articoloType mat[dimFis][dimFis];
void caricaMat(articoloType matPar[dimFis][dimFis], int dimRigPar, int dimColPar);
float trovaMax(articoloType matPar[dimFis][dimFis], int dimColPar);
int main()
{
    caricaMat(mat[dimFis][dimFis], dimRig, dimCol);
    cout << "il prezzo maggiore e' "<< trovaMax(mat[dimFis][dimFis], dimCol);
    return 0;
}
void caricaMat(articoloType matPar[dimFis][dimFis], int dimRigPar, int dimColPar)
{
    for(int i=0;i<dimColPar;i++)
    {
        for(int j=0;j<dimColPar;j++)
        {
            cout << "inserisci il codice dell'articolo " << i <<": ";
            cin >> matPar[i][j].codArt;
            cout << "inserisci il prezzo dell'articolo " << i <<": ";
            cin >> matPar[i][j].prezzo;
        }
    }
}
float trovaMax(articoloType matPar[dimFis][dimFis], int dimColPar)
{
    float massimo=matPar[3][0].prezzo;
    for(int i=0;i<dimColPar;i++)
    {
        if(massimo<matPar[3][i].prezzo) massimo=matPar[3][i].prezzo;
    }
    return massimo;
}

```

147. C++, rappresentare il voto finale di una materia degli alunni in una scuola media

/**

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3^ sez.B spec. Informatica e telecomunicazioni

Data: 28/05/2021

Numero progressivo dell'esercizio: es147

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Python 3.8.1

Obiettivo didattico: L'alunno è in grado di utilizzare gli array di record in C++

Obiettivo del programma: rappresentare il voto finale di una materia degli alunni in una scuola media

*/

```
#include <iostream>
```

```
using namespace std;
```

```
struct sezioneType
```

```
{
```

```
    string alunno;
```

```
    int voto;
```

```
};
```

```
int dimFis=100;
```

```
int dimRig=3;
```

```
int dimCol=2;
```

```
sezioneType mat[dimFis][dimFis];
```

```
void caricaMat(sezioneType matPar[dimFis][dimFis], int dimRigPar, int dimColPar);
```

```
void stampaMat(sezioneType matPar[dimFis][dimFis], int dimRigPar, int dimColPar);
```

```
int main()
```

```
{
```

```
    caricaMat(mat[dimFis][dimFis], dimRig, dimCol);
```

```

        stampaMat(mat[dimFis][dimFis], dimRig, dimCol);
        return 0;
    }
void caricaMat(sezioneType matPar[dimFis][dimFis], int dimRigPar, int dimColPar)
{
    for(int i=0;i<dimColPar;i++)
    {
        for(int j=0;j<dimColPar;j++)
        {
            cout << "inserisci il nome dell'alunno numero " << i <<": ";
            cin >> matPar[i][j].alunno;
            cout << "inserisci il prezzo dell'articolo " << i <<": ";
            cin >> matPar[i][j].voto;
        }
    }
}
void stampaMat(sezioneType matPar[dimFis][dimFis], int dimRigPar, int dimColPar)
{
    for(int i=0;i<dimColPar;i++)
    {
        for(int j=0;j<dimColPar;j++)
        {
            cout << matPar[i][j].alunno;
            cout << matPar[i][j].voto;
        }
    }
}

```

148. Prolog, rappresentare la base di conoscenza

/**

ITIS-LS "Francesco Giordani" Caserta

Anno scolastico: 2019/2020

Classe 3[^] sez.B spec. Informatica e telecomunicazioni

Data: 31/05/2021

Numero progressivo dell'esercizio: es148

Versione: 1.0

Programmatore/i: Alessandro Esposito

Sistema Operativo: Windows 10

Compilatore/Interprete: Python 3.8.1

Obiettivo didattico: Prolog

Obiettivo del programma: Rappresentare la base di conoscenza

*/

colore(a,rosso).

colore(b,verde).

colore(c,giallo).

colore(d,rosso).

colore(e,verde).

colore(f,giallo).

colore(mare,blu).

confinanti(a,b).

confinanti(a,c).

confinanti(a,e).

confinanti(a,mare).

confinanti(b,a).

confinanti(b,c).

confinanti(b,d).

confinanti(b,mare).

confinanti(c,a).

confinanti(c,d).

confinanti(c,b).

confinanti(c,e).

confinanti(d,b).

confinanti(d,f).

confinanti(d,c).

confinanti(d,e).

confinanti(d,mare).

confinanti(e,a).

confinanti(e,f).

confinanti(e,c).

confinanti(e,mare).

confinanti(f,d).

confinanti(f,e).

confinanti(f,mare).

confinanti(e,d).