

ITIS-LS “Francesco Giordani” Caserta

prof. Ennio Ranucci

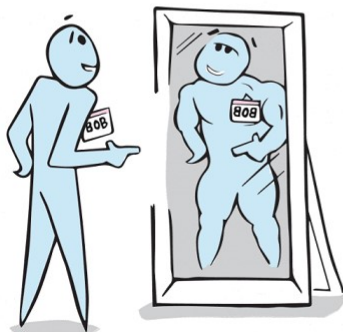
a.s. 2019-2020

## Palindrome

Java, C++, Python, Javascript, Perl, Php, MdT



02 02 2020  
0202 2020  
Palindrome day



**php** Palindrome in PHP

```
<?php
//input string is MADAM
$input = "MADAM";
echo '<br> Input String ' . $input;
//reverse of input string - MADAM - using strrev
$reverse = strrev($input);
echo '<br> Output String ' . $reverse;
//condition to check if the input and the reverse of
the string is equal or not
if($input == $reverse) {
echo '<br> '$input.' is a palindrome';
}
else {
echo '<br> '$input.' is not a palindrome';
}
?>
```

Output

MADAM

Il Palindromo, letteralmente dal greco “che corre all’indietro”, è una sequenza di caratteri che può essere letta in entrambi i versi mantenendo la sua identità. E’ di fatto bifronte, simmetrico e per questa sua virtù è stato spesso venerato e contemplato dagli antichi per la sua capacità di rappresentare il divino. Nell’antichità gli alchimisti amavano dissimulare verità sacre sotto forma di artifici matematici e letterari.

**Algoritmo ricorsivo scritto in Java in grado di riconoscere le stringhe PALINDROME.**

*Acquisisce in input una stringa e restituisce true se essa (o una sua sottostringa delimitata dai parametri i e j) è palindroma, altrimenti restituisce false.*

```
public class Palindrome
{
    public static void main(String[] args)
    {
        String parola="osso";
        if (isPalindroma(parola, 0, parola.length()-1))
            System.out.println("true");
        else
            System.out.println("false");
    }

    public static boolean isPalindroma(String str, int i, int j)
    {
        if (i>=j) return true;
        if (str.charAt(i)==str.charAt(j))
            return isPalindroma(str, i+1, j-1);
        return false;
    }
}
```

## Palindrome C++

```
#include <iostream>
using namespace std;
bool IsPalindrome(const string& str);
int main()
{
    cout << IsPalindrome("osso") << endl;
    return 0;
}

bool IsPalindrome(const string& str)
{
    if (str.empty())
        return false;
    int i = 0;
    int j = str.length() - 1;
    while (i < j)
    {
        if (str[i] != str[j])
            { return false; }
        i++;
        j--;
    }
    return true;
}
```

## Palindrome PYTHON

```
myStr = "osso"
j = myStr
rev = ""
while len(j) > 0:
    if len(j) > 0:
        a = j[-1]
        j = j[:-1]
        rev = rev + a
if rev == myStr:
    print("La stringa è Palindroma.")
else:
    print("La stringa non è Palindroma.")
```

## Palindrome JAVASCRIPT

```
<html>
<head>
<script>
function isPalindrome(string)
{
  string = string.toLocaleLowerCase();
  for (i = 0; i < string.length; i++)
  {
    if (string[i] != string[string.length - i - 1])
    {
      return alert("Non e' palindroma");
    }
  }
  return alert("E' palindroma");
}
</script>
</head>
<body>
<input type="button" value="Esegui Palindrome" onClick="isPalindrome('osso')" />
</body>
</html>
```

## Ricorsiva

```
function isPalindrome(string)
{
  string = string.toLocaleLowerCase();
  if (string.length==0 || string.length== 1)
  {
    return alert("E' palindroma");
  }
  if (string[0] == string[string.length - 1])
  {
    return isPalindrome(string.slice(1, string.length - 1) );
  }
  return alert("Non e' palindroma");
};
```

### Palindrome PERL

```
$string= "ossoe";
$reverse = reverse($string);
if($string eq $reverse)
{
    print "E'palindroma";
}
else
{
    print "Non è Palindroma";
}
```

### Palindrome PHP

```
<?php
function isPalindrome($string)
{
    $string = str_replace( ' ', "", $string );
    return $string == strrev( $string );
}
$string = 'osso';
if( isPalindrome($string ) == true )
{
    echo "E' Palindroma";
}
else
{
    echo "Non e' Palindroma";
}
?>
```

**Controllare se una stringa binaria è palindroma utilizzando il simulatore della macchina di Turing:**

<http://morphett.info/turing/turing.html>

<b>0 0 _ r 1o</b>	Se la MdT è nello stato "0" legge 0 cambia stato in "1o" ( cancella scrivendo "spazio")
<b>0 1 _ r 1i</b>	Se la MdT è nello stato "0" legge 1 cambia stato in "1i" ( cancella scrivendo "spazio")
<b>0 _ _ * accept</b>	Se la MdT è nello stato "0" legge "spazio" passa allo stato accept(fine del programma con esito positivo)
<b>1o _ _ l 2o</b>	Se la MdT è nello stato "1o" legge "spazio" torna indietro
<b>1o * * r 1o</b>	Se la MdT è nello stato "1o" legge "qualsiasi valore" passa avanti
<b>1i _ _ l 2i</b>	Se la MdT è nello stato "1i" legge "spazio" e torna indietro
<b>1i * * r 1i</b>	Se la MdT è nello stato "1i" legge "qualsiasi valore" passa avanti
<b>2o 0 _ l 3</b>	Se la MdT è nello stato "2o" legge 0 cambia stato in "3" ( cancella scrivendo "spazio")
<b>2o _ _ * accept</b>	Se la MdT è nello stato "2o" legge "spazio" passa allo stato accept (fine del programma con esito positivo)
<b>2o * * * reject</b>	Se la MdT è nello stato "2o" legge "qualsiasi valore" che è diverso da 0 cambia stato in "reject"(fine del programma con esito negativo)
<b>2i 1 _ l 3</b>	Se la MdT è nello stato "2i" legge 1 cambia stato in "3" ( cancella scrivendo "spazio")
<b>2i _ _ * accept</b>	Se la MdT è nello stato "2i" legge "spazio" passa allo stato accept (fine del programma con esito positivo)
<b>2i * * * reject</b>	Se la MdT è nello stato "2i" legge "qualsiasi valore" che è diverso da 0 cambia stato in "reject"(fine del programma con esito negativo)
<b>3 _ _ * accept</b>	Se la MdT è nello stato "3" legge "spazio" passa allo stato accept (fine del programma con esito positivo)
<b>3 * * l 4</b>	Se la MdT è nello stato "3" legge "qualsiasi valore" torna indietro, andando nello stato "4"
<b>4 * * l 4</b>	Se la MdT è nello stato "4" legge "qualsiasi valore" torna indietro
<b>4 _ _ r 0</b>	Se la MdT è nello stato "4" legge "spazio" passa avanti, tornando allo stato "0"
<b>accept * : r accept2</b>	Se la MdT è nello stato "accept" legge "qualsiasi valore" passa avanti, scrivendo il carattere ":"
<b>accept2 * ) * halt-accept</b>	Se la MdT è nello stato "accept2" legge "qualsiasi valore" si ferma, scrivendo il carattere ")"(chiude il programma)
<b>reject _ : r reject2</b>	Se la MdT è nello stato "reject" legge "spazio" passa avanti, scrivendo il carattere ":"
<b>reject * _ l reject</b>	Se la MdT è nello stato "reject" legge "qualsiasi valore" torna indietro
<b>reject2 * ( * halt-reject</b>	Se la MdT è nello stato "reject2" legge "qualsiasi valore" si ferma, scrivendo il carattere "("(chiude il programma)