

Esercizio guidato

Settimana 3
18/10/2022

Si ringrazia il Dott. Giacomo Baruzzo per il materiale

While (a contatore)

definisco il contatore

while **verifico il contatore** :

faccio cose

aggiorno contatore

```
balance = 20000
rate = 5
year = 0 # contatore
while year < 20 : # verifico il contatore
    balance = balance + balance * rate / 100
    year = year + 1 # aggiorno il contatore
```

Stringhe

Una stringa è una sequenza di caratteri; in Python deve essere racchiusa tra virgolette

```
x = "Ciao"
```

La lunghezza di una stringa si calcola con il comando **len()**

```
len(x)
```

Si può accedere a ciascun carattere di una stringa attraverso le parentesi quadre

```
x[0] è "C"  
x[1] è "i"  
x[2] è "a"  
x[3] è "o"
```

Le posizioni vanno **da 0 a len(x)-1**

Stringa palindroma

Def: Il palindromo (dal greco antico πάλιν "di nuovo" e δρόμος "percorso", col significato "che può essere percorso in entrambi i sensi") è una sequenza di caratteri che, letta al contrario, rimane invariata.

Esempi: "anna" , "otto" , "1asDsa1" , "itopinonavevanonipoti" ecc

Def alternativa: Una stringa X è palindroma se e solo se il suo **inverso** $\text{rev}(X)$ è uguale a X stessa

Esempi:

x = "otto"
rev(x) = "otto"

➡ E' palindroma

x = "Ciao"
rev(x) = "oaiC"

➡ NON è palindroma

Stringa palindroma

```
x = "Ciao"
```

```
... codice per invertire x e memorizzare il  
risultato in y...
```

```
if x == y :  
    print("Palindromo!")  
else :  
    print ("Non palindromo!")
```

Problema di questa soluzione:

- Se x è molto lunga (esempio: genoma umano 3,2 miliardi di caratteri), invertire la stringa richiede **molta memoria**
- mentre magari se la stringa NON è un palindromo la verifica potrebbe terminare molto rapidamente...
sprecando meno memoria...

Esempio di stringa palindroma

- $S = \text{"a321ca123a"}$
- $\text{len}(S) = 10$

a	3	2	1	c	a	1	2	3	a
<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>

← caratteri

← posizioni

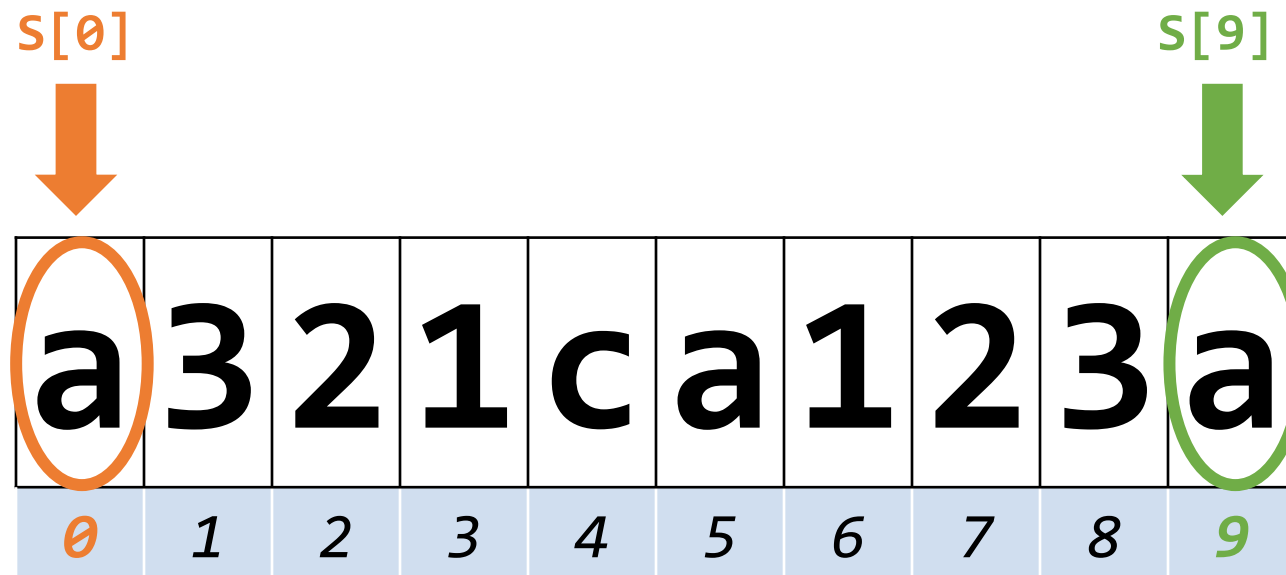
Esempio di stringa palindroma

- $S = \text{"a321ca123a"}$
- $\text{len}(S) = 10$

Step 1

le	S[le]
0	a

ri	S[ri]
9 (N-1)	a



← caratteri

← posizioni

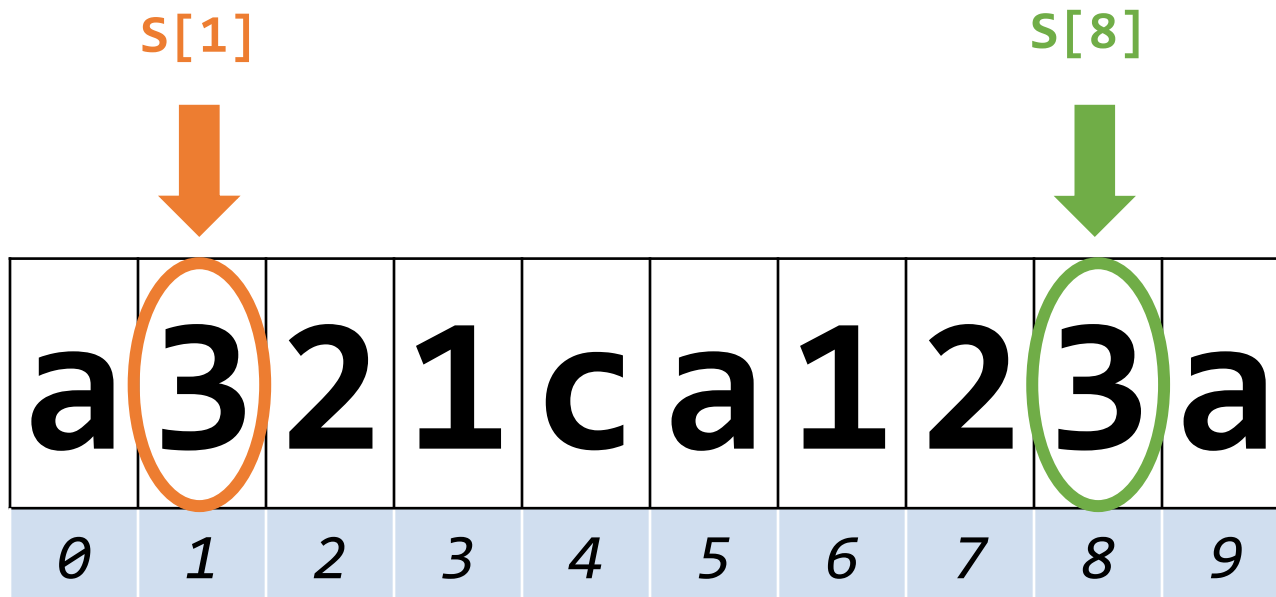
Esempio di stringa palindroma

- $S = \text{"a321ca123a"}$
- $\text{len}(S) = 10$

Step 2

le	S[le]
0	a
1	3

ri	S[ri]
9 (N-1)	a
8 (N-2)	3



← caratteri

← posizioni

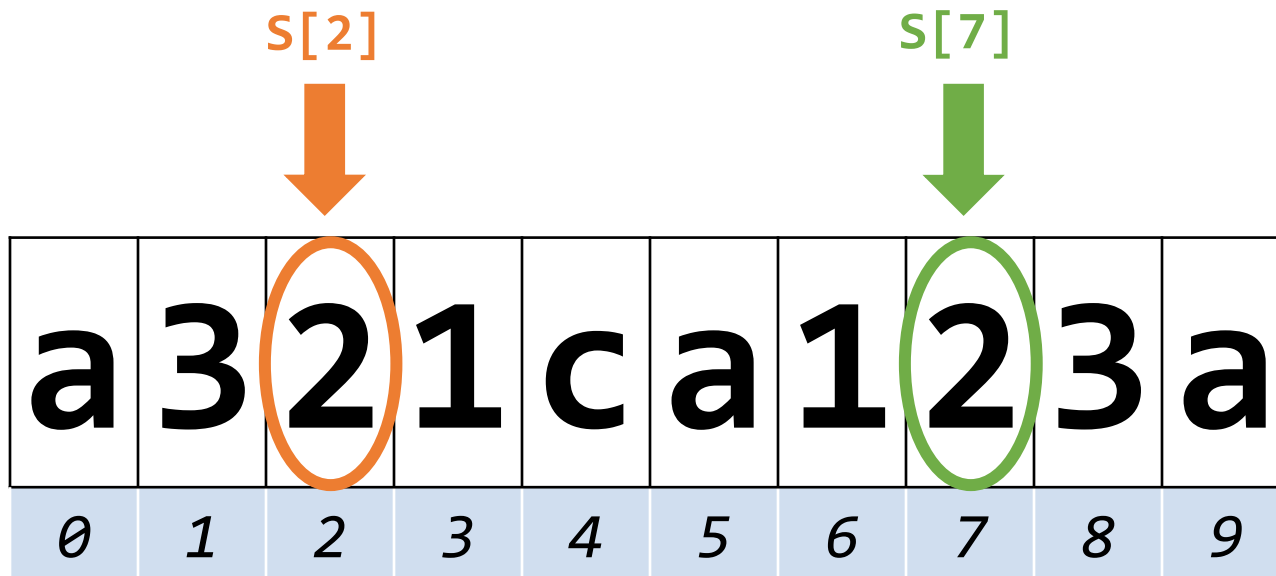
Esempio di stringa palindroma

- $S = \text{"a321ca123a"}$
- $\text{len}(S) = 10$

Step 3

le	S[le]
0	a
1	3
2	2

ri	S[ri]
9 (N-1)	a
8 (N-2)	3
7 (N-3)	2



← caratteri

← posizioni

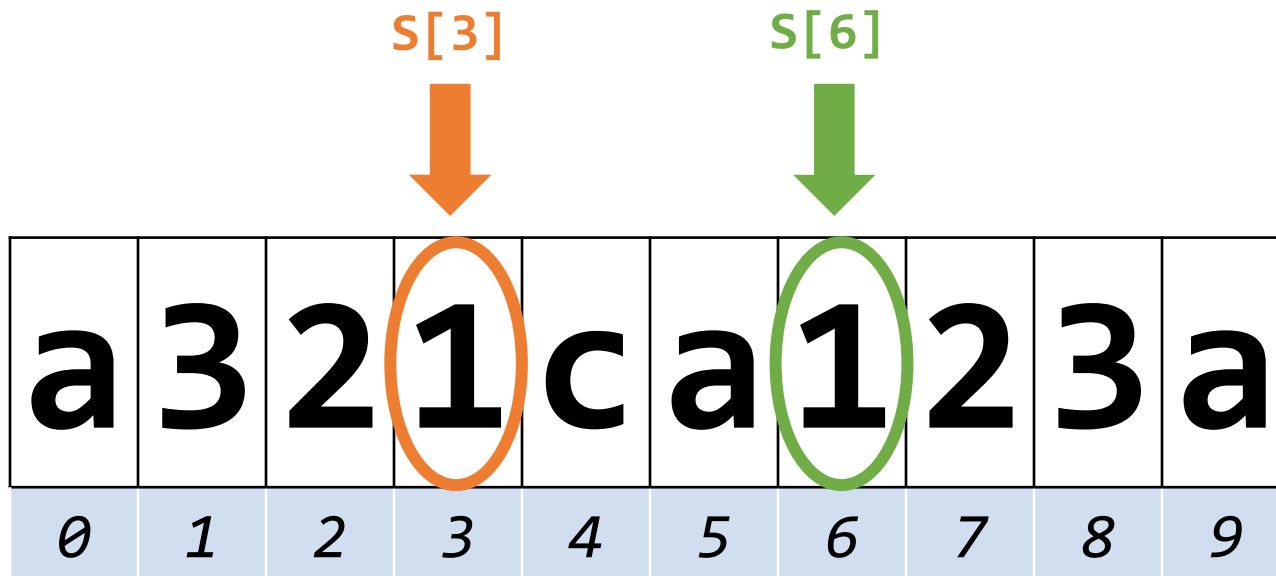
Esempio di stringa palindroma

- $S = \text{"a321ca123a"}$
- $\text{len}(S) = 10$

Step 4

le	S[le]
0	a
1	3
2	2
3	1

ri	S[ri]
9 (N-1)	a
8 (N-2)	3
7 (N-3)	2
6 (N-4)	1



← caratteri

← posizioni

Esempio di stringa palindroma

- $S = \text{"a321ca123a"}$
- $\text{len}(S) = 10$

le	S[le]
0	a
1	3
2	2
3	1
4	c

ri	S[ri]
9 (N-1)	a
8 (N-2)	3
7 (N-3)	2
6 (N-4)	1
5 (N-5)	a

Step 5

$S[4]$ $S[5]$



a	3	2	1	c	a	1	2	3	a
0	1	2	3	4	5	6	7	8	9

← caratteri

← posizioni

Osservazioni

- Con al più **$\text{len}(S)/2$** step (confronti tra coppie di caratteri) è possibile stabilire se la stringa è palindroma
- Se la stringa è di **lunghezza dispari**, il carattere centrale può essere ignorato
- La **stringa vuota** è per definizione palindroma

Algoritmo

- Leggo la stringa S da input
- Scorro (la prima metà) della stringa S
- Ad ogni step
 - Seleziono il carattere di sinistra ($S[l_e]$)
 - Seleziono il carattere di destra ($S[r_i]$)
 - Confronto i due caratteri
 - Salvo il risultato del confronto
- Comunico a video se S è palindromo o meno

Dall'algoritmo al codice

```
S = input("Inserire la stringa da
verificare: ")

isPalindrome = True
i = 0
while i < len(S) // 2 :
    le = i
    ri = len(S) - 1 - le
    if S[le] != S[ri] :
        isPalindrome = False
    i += 1
if isPalindrome :
    print(S, "è un palindromo")
else :
    print(S, "non è un palindromo")
```

- Leggo la stringa S da input
- Scorro (la prima metà) della stringa S
- Ad ogni step
 - Seleziono il carattere di sinistra (S[le])
 - Seleziono il carattere di destra (S[ri])
 - Confronto i due caratteri
 - Salvo il risultato del confronto
- Comunico a video se S è palindromo o meno

Dall'algoritmo al codice

```
S = input("Inserire la stringa da
verificare: ")

isPalindrome = True
i = 0
while i < len(S) // 2 :
    le = i
    ri = len(S) - 1 - le
    if S[le] != S[ri] :
        isPalindrome = False
    i += 1
if isPalindrome :
    print(S, "è un palindromo")
else :
    print(S, "non è un palindromo")
```

- Leggo la stringa S da input
- Scorro (la prima metà) della stringa S
- Ad ogni step
 - Seleziono il carattere di sinistra (S[le])
 - Seleziono il carattere di destra (S[ri])
 - Confronto i due caratteri
 - Salvo il risultato del confronto
- Comunico a video se S è palindromo o meno

Dall'algoritmo al codice

```
S = input("Inserire la stringa da
verificare: ")

isPalindrome = True
i = 0
while i < len(S) // 2 :
    le = i
    ri = len(S) - 1 - le
    if S[le] != S[ri] :
        isPalindrome = False
    i += 1
if isPalindrome :
    print(S, "è un palindromo")
else :
    print(S, "non è un palindromo")
```

- Leggo la stringa S da input
- Scorro (la prima metà) della stringa S
- Ad ogni step
 - Seleziono il carattere di sinistra (S[le])
 - Seleziono il carattere di destra (S[ri])
 - Confronto i due caratteri
 - Salvo il risultato del confronto
- Comunico a video se S è palindromo o meno

Individuare $S[l_e]$ e $S[r_i]$

- Per trovare $S[l_e]$ e $S[r_i]$ servono gli indici l_e e r_i
- Ad ogni step
 - l_e è noto (coincide con i , l'indice per scorrere la stringa)
 - Come trovo il valore di r_i corrispondente?

Individuare $S[le]$ e $S[ri]$

- Per trovare $S[le]$ e $S[ri]$ servono gli indici le e ri
- Ad ogni step
 - le è noto (coincide con i , l'indice per scorrere la stringa)
 - Come trovo il valore di ri corrispondente?

- Esempio:

- $S = \text{"a321cc123a"}$

le	S[le]
0	a
1	3
2	2
3	1
4	c

ri	S[ri]
9 (N-1)	a
8 (N-2)	3
7 (N-3)	2
6 (N-4)	1
5 (N-5)	c

Individuare $S[le]$ e $S[ri]$

- Per trovare $S[le]$ e $S[ri]$ servono gli indici le e ri
- Ad ogni step
 - le è noto (coincide con i , l'indice per scorrere la stringa)
 - Come trovo il valore di ri corrispondente?

- Esempio:
 - $S = \text{"a321cc123a"}$

- $le = i$
- $ri = \text{len}(S) - 1 - le$

le	S[le]
0	a
1	3
2	2
3	1
4	c

ri	S[ri]
9 (N-1)	a
8 (N-2)	3
7 (N-3)	2
6 (N-4)	1
5 (N-5)	c

Dall'algoritmo al codice

```
S = input("Inserire la stringa da
verificare: ")

isPalindrome = True
i = 0
while i < len(S) // 2 :
    le = i
    ri = len(S) - 1 - le
    if S[le] != S[ri] :
        isPalindrome = False
    i += 1
if isPalindrome :
    print(S, "è un palindromo")
else :
    print(S, "non è un palindromo")
```

- Leggo la stringa S da input
- Scorro (la prima metà) della stringa S
- Ad ogni step
 - Seleziono il carattere di sinistra (S[le])
 - Seleziono il carattere di destra (S[ri])
 - Confronto i due caratteri
 - Salvo il risultato del confronto
- Comunico a video se S è palindromo o meno

Dall'algoritmo al codice

```
S = input("Inserire la stringa da
verificare: ")

isPalindrome = True
i = 0
while i < len(S) // 2 :
    le = i
    ri = len(S) - 1 - le
    if S[le] != S[ri] :
        isPalindrome = False
    i += 1
if isPalindrome :
    print(S, "è un palindromo")
else :
    print(S, "non è un palindromo")
```

- Leggo la stringa S da input
- Scorro (la prima metà) della stringa S
- Ad ogni step
 - Seleziono il carattere di sinistra (S[le])
 - Seleziono il carattere di destra (S[ri])
 - Confronto i due caratteri
 - Salvo il risultato del confronto
- Comunico a video se S è palindromo o meno

Dall'algoritmo al codice

```
S = input("Inserire la stringa da
verificare: ")

isPalindrome = True
i = 0
while i < len(S) // 2 :
    le = i
    ri = len(S) - 1 - le
    if S[le] != S[ri] :
        isPalindrome = False
    i += 1
if isPalindrome :
    print(S, "è un palindromo");
else :
    print(S, "non è un palindromo");
```

- Leggo la stringa S da input
- Scorro (la prima metà) della stringa S
- Ad ogni step
 - Seleziono il carattere di sinistra (S[le])
 - Seleziono il carattere di destra (S[ri])
 - Confronto i due caratteri
 - Salvo il risultato del confronto
- Comunico a video se S è palindromo o meno

Dall'algoritmo al codice

```
S = input("Inserire la stringa da
verificare: ")

isPalindrome = True
i = 0
while i < len(S) // 2 :
    le = i
    ri = len(S) - 1 - le
    if S[le] != S[ri] :
        isPalindrome = False
    i += 1
if isPalindrome :
    print(S, "è un palindromo")
else :
    print(S, "non è un palindromo")
```

- Leggo la stringa S da input
- Scorro (la prima metà) della stringa S
- Ad ogni step
 - Seleziono il carattere di sinistra (S[le])
 - Seleziono il carattere di destra (S[ri])
 - Confronto i due caratteri
 - Salvo il risultato del confronto
- Comunico a video se S è palindromo o meno

Esempi di input – Stringa vuota

- $S = ""$

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = 0
```

```
while i < len(S) // 2 :
```

```
    le = i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i += 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```


Esempi di input – Stringa vuota

- $S = ""$

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = 0
```

```
while i < len(S) // 2 :
```

```
    le = i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i += 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

Esempi di input – Stringa vuota

- $S = ""$

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = 0
```

```
while i < len(S) // 2 :
```

```
    le = i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i += 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

$\text{len}(S)$ è 0

quindi $\text{len}(S)//2$ è 0

$i < 0$? No, quindi non entro nel ciclo

Esempi di input – Stringa vuota

- $S = ""$

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = 0
```

```
while i < len(S) // 2 :
```

```
    le = i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i += 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

Esempi di input – Non palindromo

- S = “abzheba”

0 1 2 3 4 5 6

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = 0
```

```
while i < len(S) // 2 :
```

```
    le = i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i += 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

Esempi di input – Non palindromo

- S = “abzheba”

0 1 2 3 4 5 6

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = 0
```

```
while i < len(S) // 2 :
```

```
    le = i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i += 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

Esempi di input – Non palindromo

- $S = \text{"abzheba"}$

0 1 2 3 4 5 6

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = 0
```

```
while i < len(S) // 2 :
```

```
    le = i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i += 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

$\text{len}(S)$ è 7

quindi $\text{len}(S) // 2$ è 3

$i < 3$? Sì, quindi entro nel ciclo

Esempi di input – Non palindromo

- $S = \text{"abzheba"}$

0 1 2 3 4 5 6

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = 0
```

```
while i < len(S) // 2 :
```

```
    le = i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i += 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

$\text{len}(S)$ è 7

quindi $\text{len}(S) // 2$ è 3

$i < 3$? Sì, quindi entro nel ciclo

i	$i < \text{len}(S) // 2$	le	ri	S[le]	S[ri]	isPalindrome
0	True	0	$7-1-0 = 6$	a	a	True
1	True	1	$7-1-1 = 5$	b	b	True
2	True	2	$7-1-2 = 4$	z	e	False
3	False					

Esempi di input – Non palindromo

- $S = \text{"abzheba"}$

0 1 2 3 4 5 6

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = 0
```

```
while i < len(S) // 2 :
```

```
    le = i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i += 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

$\text{len}(S)$ è 7

quindi $\text{len}(S) // 2$ è 3

$i < 3$? Sì, quindi entro nel ciclo

i	$i < \text{len}(S) // 2$	le	ri	S[le]	S[ri]	isPalindrome
0	True	0	$7-1-0 = 6$	a	a	True
1	True	1	$7-1-1 = 5$	b	b	True
2	True	2	$7-1-2 = 4$	z	e	False
3	False					

Esempi di input - Palindromo

- S = “abzhzba”

0 1 2 3 4 5 6

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = 0
```

```
while i < len(S) // 2 :
```

```
    le = i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i += 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

Esempi di input - Palindromo

- S = "abzhzba"

0 1 2 3 4 5 6

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = 0
```

```
while i < len(S) // 2 :
```

```
    le = i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i += 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

Esempi di input - Palindromo

- $S = \text{"abzhzba"}$

0 1 2 3 4 5 6

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = 0
```

```
while i < len(S) // 2 :
```

```
    le = i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i += 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

$\text{len}(S)$ è 7

quindi $\text{len}(S)//2$ è 3

$i < 3$? Sì, quindi entro nel ciclo

Esempi di input - Palindromo

- $S = \text{"abzhzba"}$

0 1 2 3 4 5 6

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = 0
```

```
while i < len(S) // 2 :
```

```
    le = i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i += 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

$\text{len}(S)$ è 7

quindi $\text{len}(S) // 2$ è 3

$i < 3$? Sì, quindi entro nel ciclo

i	$i < \text{len}(S) // 2$	le	ri	S[le]	S[ri]	isPalindrome
0	True	0	$7-1-0 = 6$	a	a	True
1	True	1	$7-1-1 = 5$	b	b	True
2	True	2	$7-1-2 = 4$	z	z	True
3	False					

Esempi di input - Palindromo

- $S = \text{"abzhzba"}$

0 1 2 3 4 5 6

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = 0
```

```
while i < len(S) // 2 :
```

```
    le = i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i += 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

$\text{len}(S)$ è 7

quindi $\text{len}(S) // 2$ è 3

$i < 3$? Sì, quindi entro nel ciclo

i	$i < \text{len}(S) // 2$	le	ri	S[le]	S[ri]	isPalindrome
0	True	0	$7-1-0 = 6$	a	a	True
1	True	1	$7-1-1 = 5$	b	b	True
2	True	2	$7-1-2 = 4$	z	z	True
3	False					

Versione alternativa 1

- Abbiamo percorso la stringa dai 2 estremi verso il centro...
- possiamo fare anche l'opposto: partiamo dal centro e andiamo verso i 2 estremi
- Problema:
 - Come “muovo” gli indici *le* e *ri* ?

Versione alternativa 1

```
S = input("Inserire la stringa da verificare: ")
isPalindrome = True
i = 0
while i < len(S) // 2 :
    le = len(S) // 2 - 1 - i
    ri = len(S) - 1 - le
    if S[le] != S[ri] :
        isPalindrome = False
    i += 1
if isPalindrome :
    print(S, "è un palindromo")
else :
    print(S, "non è un palindromo")
```

Versione alternativa 1

- S = “abzhzba”

0 1 2 3 4 5 6

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = 0
```

```
while i < len(S) // 2 :  
    le = len(S) // 2 - 1 - i  
    ri = len(S) - 1 - le  
    if S[le] != S[ri] :  
        isPalindrome = False  
    i += 1
```

```
if isPalindrome :  
    print(S, "è un palindromo")
```

```
else :  
    print(S, "non è un palindromo")
```


Versione alternativa 1

- $S = \text{"abzhzba"}$

0 1 2 3 4 5 6

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = 0
```

```
while i < len(S) // 2 :  
    le = len(S) // 2 - 1 - i  
    ri = len(S) - 1 - le  
    if S[le] != S[ri] :  
        isPalindrome = False  
    i += 1
```

```
if isPalindrome :  
    print(S, "è un palindromo")
```

```
else :  
    print(S, "non è un palindromo")
```

$\text{len}(S)$ è 7

quindi $\text{len}(S) // 2$ è 3

$i < 3$? Sì, quindi entro nel ciclo

Versione alternativa 1

- $S = \text{"abzhzba"}$

0 1 2 3 4 5 6

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = 0
```

```
while i < len(S) // 2 :
```

```
    le = len(S) // 2 - 1 - i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i += 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

$\text{len}(S)$ è 7

quindi $\text{len}(S) // 2$ è 3

$i < 3$? Sì, quindi entro nel ciclo

i	$i < \text{len}(S) // 2$	le	ri	S[le]	S[ri]	isPalindrome
0	True	2	4	z	z	True
1	True	1	5	b	b	True
2	True	0	6	a	a	True
3	False					

Versione alternativa 1

- $S = \text{"abzhzba"}$

0 1 2 3 4 5 6

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = 0
```

```
while i < len(S) // 2 :
```

```
    le = len(S) // 2 - 1 - i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i += 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

$\text{len}(S)$ è 7

quindi $\text{len}(S) // 2$ è 3

$i < 3$? Sì, quindi entro nel ciclo

i	$i < \text{len}(S) // 2$	le	ri	S[le]	S[ri]	isPalindrome
0	True	2	4	z	z	True
1	True	1	5	b	b	True
2	True	0	6	a	a	True
3	False					

Versione alternativa 1

```
S = input("Inserire la stringa da verificare: ")
isPalindrome = True
i = 0
while i < len(S) // 2 :
    le = len(S) // 2 - 1 - i
    ri = len(S) - 1 - le
    if S[le] != S[ri] :
        isPalindrome = False
    i += 1
if isPalindrome :
    print(S, "è un palindromo")
else :
    print(S, "non è un palindromo")
```

Versione alternativa 1

```
S = input("Inserire la stringa da  
verificare: ")

isPalindrome = True
i = 0
while i < len(S) // 2 :
    le = len(S) // 2 - 1 - i
    ri = len(S) - 1 - le
    if S[le] != S[ri] :
        isPalindrome = False
    i += 1
if isPalindrome :
    print(S, "è un palindromo")
else :
    print(S, "non è un palindromo")
```

Versione alternativa 2

```
S = input("Inserire la stringa da  
verificare: ")

isPalindrome = True
i = len(S) // 2 - 1
while i >= 0 :
    le = i
    ri = len(S) - 1 - le
    if S[le] != S[ri] :
        isPalindrome = False
    i -= 1
if isPalindrome :
    print(S, "è un palindromo")
else :
    print(S, "non è un palindromo")
```

Versione alternativa 2

- S = “abzhzba”

0 1 2 3 4 5 6

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = len(S) // 2 - 1
```

```
while i >= 0 :
```

```
    le = i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i -= 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

Versione alternativa 2

- S = “abzhzba”

0 1 2 3 4 5 6

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = len(S) // 2 - 1
```

```
while i >= 0 :
```

```
    le = i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i -= 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

Versione alternativa 2

- $S = \text{"abzhzba"}$

0 1 2 3 4 5 6

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = len(S) // 2 - 1
```

```
while i >= 0 :
```

```
    le = i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i -= 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

$\text{len}(S)$ è 7

quindi $\text{len}(S) // 2 - 1$ è 2

$i \geq 0$? Sì, quindi entro nel ciclo

Versione alternativa 2

- $S = \text{"abzhzba"}$

0 1 2 3 4 5 6

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = len(S) // 2 - 1
```

```
while i >= 0 :
```

```
    le = i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i -= 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

$\text{len}(S)$ è 7

quindi $\text{len}(S) // 2 - 1$ è 2

$i \geq 0$? Sì, quindi entro nel ciclo

i	$i \geq 0$	le	ri	S[le]	S[ri]	isPalindrome
2	True	2	4	z	z	True
1	True	1	5	b	b	True
0	True	0	6	a	a	True
-1	False					

Versione alternativa 2

- $S = \text{"abzhzba"}$

0 1 2 3 4 5 6

```
S = input("Inserire la stringa da  
verificare: ")
```

```
isPalindrome = True
```

```
i = len(S) // 2 - 1
```

```
while i >= 0 :
```

```
    le = i
```

```
    ri = len(S) - 1 - le
```

```
    if S[le] != S[ri] :
```

```
        isPalindrome = False
```

```
    i -= 1
```

```
if isPalindrome :
```

```
    print(S, "è un palindromo")
```

```
else :
```

```
    print(S, "non è un palindromo")
```

$\text{len}(S)$ è 7

quindi $\text{len}(S) // 2 - 1$ è 2

$i \geq 0$? Sì, quindi entro nel ciclo

i	$i \geq 0$	le	ri	S[le]	S[ri]	isPalindrome
2	True	2	4	z	z	True
1	True	1	5	b	b	True
0	True	0	6	a	a	True
-1	False					