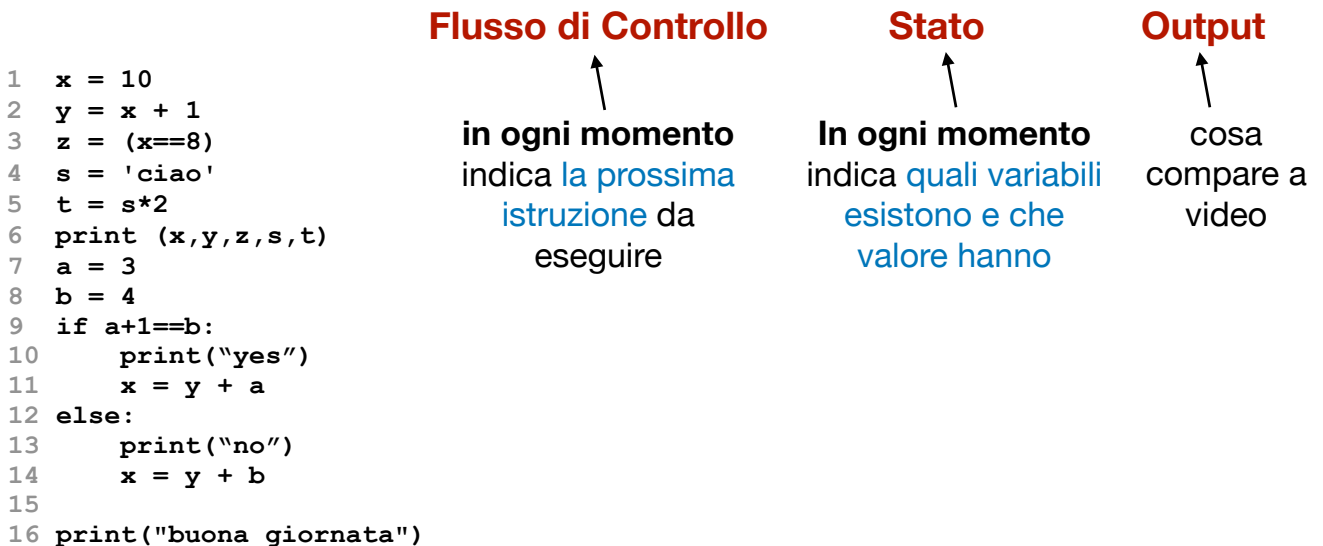
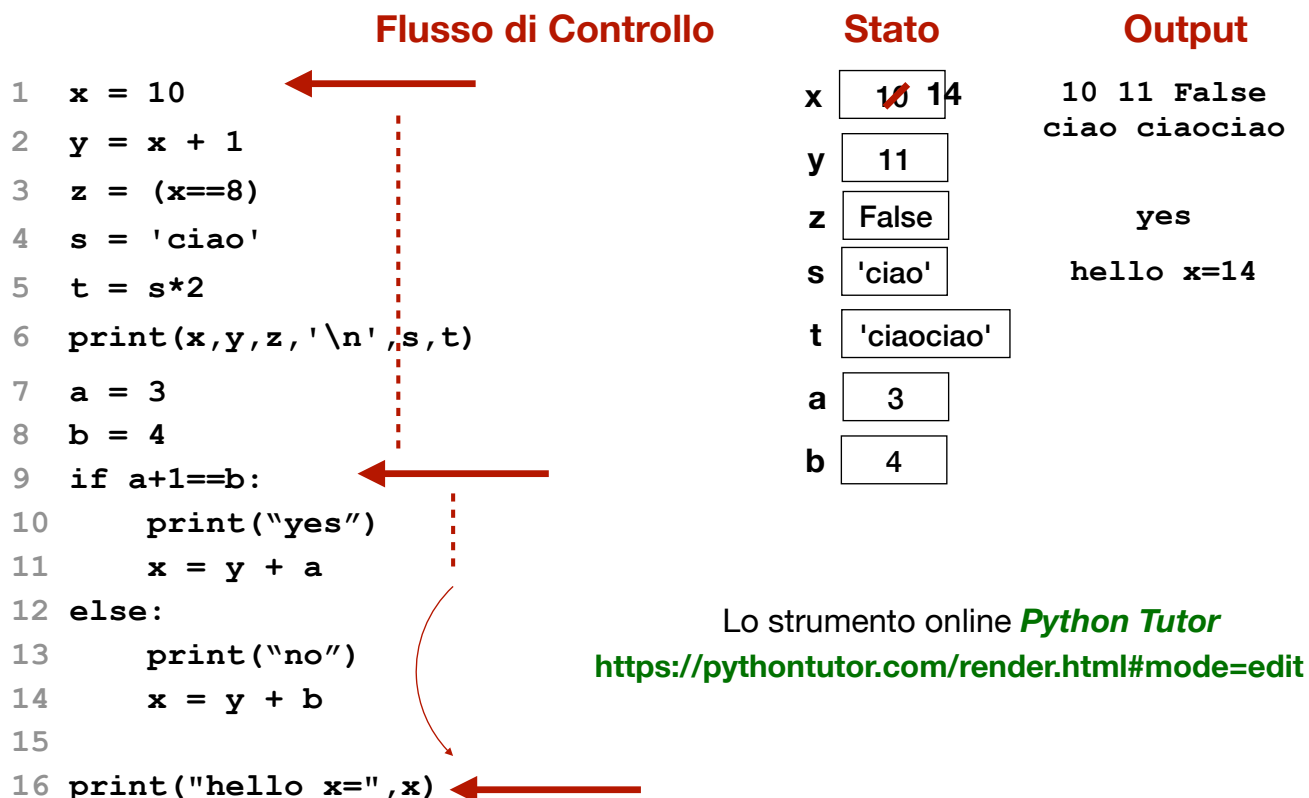


Esecuzione di un programma

Per tracciare con precisione **cosa accade durante l'esecuzione** di un programma, serve tenere traccia di 3 cose:



Esecuzione di un programma




cicli e iterazione

- **ripetere** un gruppo di istruzioni un certo numero di volte:

```
for variabile in elenco:  
    blocco_istruzioni # body
```

la variabile *itera* sui valori nell'elenco e
per ogni valore esegue il corpo del for



```
for i in [1,2,3,4,5]:  
    print("Stay hungry,")  
    print(" stay foolish!")
```

5 ripetizioni


```
body  
con i=1 [ Stay hungry,  
         stay foolish!  
body  
con i=2 [ Stay hungry,  
         stay foolish!  
         Stay hungry,  
         stay foolish!  
         Stay hungry,  
         stay foolish!  
body  
con i=5 [ Stay hungry,  
         stay foolish!
```

cicli e iterazione

- **ripetere** un gruppo di istruzioni un certo numero di volte:

```
for variabile in elenco:  
    blocco_istruzioni # body
```

la variabile *itera* sui valori nell'elenco e
per ogni valore esegue il corpo del for



```
for i in range(5):  
    print("Stay hungry,")  
    print(" stay foolish!")
```

[0,5)=[0,1,2,3,4]

5 ripetizioni

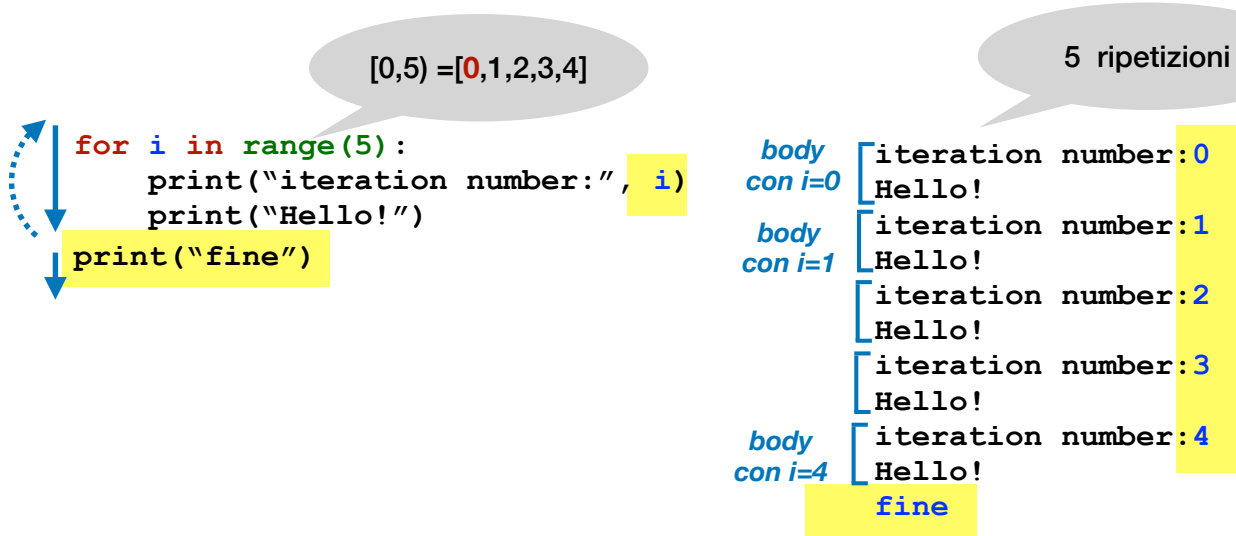
```
body  
con i=0 [ Stay hungry,  
         stay foolish!  
body  
con i=1 [ Stay hungry,  
         stay foolish!  
         Stay hungry,  
         stay foolish!  
         Stay hungry,  
         stay foolish!  
body  
con i=4 [ Stay hungry,  
         stay foolish!
```

cicli e iterazione

- **ripetere** un gruppo di istruzioni un certo numero di volte:

```
for variabile in elenco:  
    blocco_istruzioni # body
```

la variabile *itera* sui valori nell'elenco e
per ogni valore esegue il corpo del for



Esercizio

- Come si comportano i seguenti programmi?

```
for i in range(5):  
    print("iteration:", i)  
    print("Hello!")  
print("fine")
```

```
for i in range(5):  
    print("iteration:", i)  
    print("Hello!")  
print("fine")
```

fare molta attenzione
all'indentazione.
Questi errori (logici) sono
difficili da trovare

cicli e iterazioni

in generale la funzione `range` ha 3 argomenti:

`range(begin, end, step)` = *numeri da begin a end escluso, di step in step*

```
for i in range(1,8,2):  
    print("hello iteration:" i)
```

numeri da 1 a 8 escluso
di 2 in 2:
[1,3,5,7]

```
for i in range(6,2,-1):  
    print("hello iteration:" i)
```

numeri da 6 a 2 escluso
di -1 in -1
[6,5,4,3]

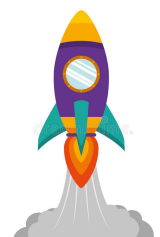
`range(5)` sta per `range(begin=0, end=5, step=1)` = [0,1,2,3,4]

Esempio

Sapendo che `range(10,0,-1)` corrisponde all'elenco [10,9,8,7,...,2,1]
stampare un conto alla rovescia

```
print("Countdown:")  
for i in range(10,0,-1):  
    print(i)  
print("Ignition!")
```

Countdown:
10
9
8
7
6
5
4
3
2
1
Ignition!



cicli e iterazione

- **ripetere** un gruppo di istruzioni un certo numero di volte:

```
for variabile in elenco:
    blocco_istruzioni # body
```

qualsiasi elenco, non solo elenco di numeri

una stringa è un elenco di caratteri

```
for lettera in "banana":
    print(lettera)
```

b ← body con lettera='b'
a ← body con lettera='a'
n
a
n
a

```
frutto = input("scegli un frutto")
for lettera in frutto:
    print(lettera*2)
```

in Python ci sono tanti tipi di elenchi:
range(...), *stringhe*,
liste, *dizionari*, *tuple...*

cicli e iterazione

- **ripetere** un gruppo di istruzioni un certo numero di volte:

```
for variabile in elenco:
    blocco_istruzioni # body
```

qualsiasi istruzione:
es. assegnamenti, condizionali,
anche altri for

```
for i in range(10):
    if i%2==0:
        print(i)
```

stampa le cifre pari 0 2 4 6 8

Esercizi:

1. Scrivi un programma che usa il comando for per stampare le cifre dispari
2. completa il codice seguente in modo che stampi i primi 10 numeri pari

```
for i in range(???) :
    print(i)
```

cicli e iterazione

```
for i in ["cane", "gatto", "oca"]:  
    for j in range(1,4):  
        print(i, j, end=' ')  
    print()
```

i="cane"

| | | |
|---------|------------|--------|
| j=1 | stampa i,j | cane 1 |
| j=2 | stampa i,j | cane 2 |
| j=3 | stampa i,j | cane 3 |
| print() | va a capo | |

i="gatto"

| | | |
|---------|------------|---------|
| j=1 | stampa i,j | gatto 1 |
| j=2 | stampa i,j | gatto 2 |
| j=3 | stampa i,j | gatto 3 |
| print() | va a capo | |

i="oca"

| | | |
|---------|------------|-------|
| j=1 | stampa i,j | oca 1 |
| j=2 | stampa i,j | oca 2 |
| j=3 | stampa i,j | oca 3 |
| print() | va a capo | |

```
cane 1 cane 2 cane 3  
gatto 1 gatto 2 gatto 3  
oca 1 oca 2 oca 3
```

cicli e iterazione

```
for i in ["Mary's", "Bob's", "Joe's"]:  
    for j in ["cat", "dog", "duck"]:  
        print(i, j, end=' ')  
    print()
```

Cosa stampa ???

ESERCIZIO: modificare il programma in modo da rendere più leggibile l'output stampando delle virgole tra gli animali dello stesso padrone. Fare altre modifiche in modo da rendere più simpatico l'output

cicli e iterazione

```
for i in range(1,4):  
    for j in range(1,4):  
        print(i*j, end=' ')  
    print()
```

Cosa stampa?

....sono le tabelline!

```
i=1  
j=1 stampa 1*1  
j=2 stampa 1*2  
j=3 stampa 1*3  
print()  
1 2 3  
2 4 6  
3 6 9
```

```
i=2  
j=1 stampa 2*1  
j=2 stampa 2*2  
j=3 stampa 2*3  
print()
```

```
i=3  
j=1 stampa 3*1  
j=2 stampa 3*2  
j=3 stampa 3*3
```

Esempio

Cosa stampa?

....sono le tavole di verità!

```
elenco = [True, False] #lista di 2 valori  
for a in elenco:  
    print("a=", a, " not a=", not a)
```

stampe

```
a= True    not a= False  
a= False   not a= True
```

```
elenco = [True, False]  
for a in elenco:  
    for b in elenco:  
        print("a=", a, "b=", b)
```

iterazioni

stampe

a=True

```
b=True    a=True  b=True  
b=False   a=True  b=False
```

a=False

```
b=True    a=False b=True  
b=False   a=False b=False
```

```
elenco = [True, False]  
for a in elenco:  
    for b in elenco:  
        print(a, "or", b, "=", a or b)
```

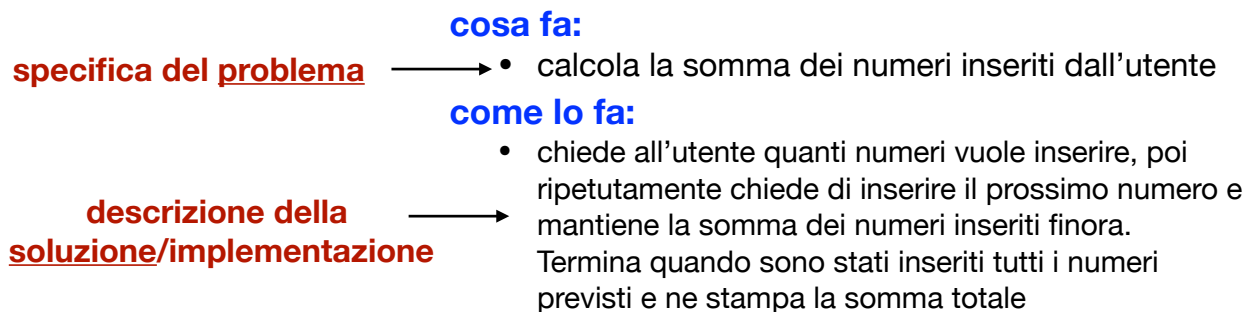
stampe

```
True or True = True  
True or False = True  
False or True = True  
False or False = False
```

cicli e iterazione

Descrivere cosa fa questo programma

```
num_input = input("Quanti numeri vuoi inserire?")
n = int(num_input)
sum = 0
for i in range(n):
    msg = "inserisci il prossimo numero "
    x = int(input(msg))
    sum = sum + x
print("somma: ", sum)
```



cicli e iterazione

Descrivere cosa fa questo programma

```
import random

n = random.randint(1,9) # restituisce un numero a caso in [1,9]
sum = 0
for i in range(n):
    msg = "inserisci il prossimo numero "
    x = int(input(msg))
    sum = sum + x
print("somma: ", sum)
```


Esercizio

Descrivere cosa fa questo programma

```
num_input = input("Quanti numeri vuoi inserire?")
n = int(num_input)
sum = 0
for i in range(n):
    msg = "inserisci il numero " + str(i+1) + " di " + str(n)
    x = int(input(msg))
    sum = sum + x
print("somma: ",sum)
```

cicli e iterazione

Descrivere cosa fa questo programma

```
import random
n = random.randint(1,100)
sum=0
for i in range(n):
    msg = "inserisci il prossimo numero oppure -1 per smettere "
    x = int(input(msg))
    if x == -1:
        break # interrompe l'iterazione e il flusso di controllo
              # salta all'istruzione istruzione che segue il for
    else:
        sum = sum + x
print("somma: ",sum)
```

Esercizio

Scrivere un programma che prende in input un numero n, legge n numeri e calcola e restituisce la somma e la media dei numeri, **tra quelli inseriti, che sono pari e multipli di 3**

```
n = int(input("Quanti numeri vuoi inserire?"))
sum = 0
n_da_considerare = 0

#leggi n numeri
for i in range(n):
    new_num=int(input("inserisci il prossimo:"))
    # controlla se pari e multiplo di 3
    if new_num%2==0 and new_num%3==0:
        sum = sum + new_num
        n_da_considerare = n_da_considerare +1

print("somma:", sum)
print("media:", sum/n_da_considerare)
```

Esercizio:

implementare gli algoritmi scritti in pseudocodice in un software python

- Algoritmo per un programma che verifica se un numero è pari o dispari
- Algoritmo per un programma che verifica se un utente è maggiorenne
- Algoritmo per un programma che verifica se una parola è **palindroma**
- Scrivere un programma che, dato un elenco di prezzi di articoli, dice il totale della spesa
- Scrivere un programma che calcola la media degli esami

cosa riusciamo ad implementare e cosa non sappiamo fare?

Esercizio del calendario

- **Generalizzare il programma** chiedendo in input una data di quest'anno (giorno e mese) e producendo in output il numero di giorni dal 1 gennaio 2023 a quella data (non facile)

Il gioco Mastermind prevede 2 giocatori:

- il giocatore 1 sceglie una sequenza nascosta di **4** pioli colorati, scegliendo tra **6 colori disponibili** (arancio, blu, nero, verde, giallo, rosso);
- il giocatore 2 deve indovinare la sequenza nascosta. Ad ogni tentativo, propone una sequenza di 4 pioli colorati e la sottopone al controllo;
- controllare una sequenza significa dare due informazioni:
 - quanti pioli ci sono del colore giusto al posto giusto
 - quanti pioli ci sono del colore giusto al posto sbagliato
- in ogni momento il giocatore 2 può vedere tutti i suoi tentativi precedenti con le relative informazioni di controllo
- il gioco può prevedere un numero massimo di tentativi



Esercizio (difficile ma da fare pian piano prima dell'esame)

Scrivere un programma che permette a due giocatori di giocare a Mastermind

N.B. **non significa** che il programma indovina automaticamente la soluzione, ma solo che permette di giocare

- Pensare al funzionamento logico del programma: cosa deve fare? come lo fa?
- Provare a scrivere in qualche modo questo funzionamento

cicli e iterazione: while

- **non sempre si sa quante volte ripetere** il ciclo. Ad esempio si deve ripetere **finché è vera/falsa una certa condizione**
- Il costrutto più generale per l'iterazione è **while**

```
while condizione:  
    blocco_istruzioni #body
```

1. si valuta la condizione, che deve essere un'espressione booleana
2. **se la condizione è vera**,
 - si esegue il corpo del ciclo e **poi si torna al passo 1**.
3. **se la condizione è falsa**
 - si **esce dal ciclo** e si esegue l'istruzione che segue il while.

- in generale, le istruzioni del corpo devono cambiare il valore di verità della condizione, altrimenti il ciclo si ripete all'infinito (**loop forever**)

while e for

```
# Countdown  
for n in range(10,0,-1):  
    print(n)  
print("Ignition!")
```

```
# Countdown  
n = 10  
while n > 0:  
    print(n)  
    n = n-1  
print("Ignition!")
```

```
10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
Ignition!
```

se il corpo di
while non modifica n,
la condizione sarà
sempre vera

cicli e iterazione: while

- **non** sempre si sa quante volte ripetere il ciclo. Ad esempio si deve ripetere **finché è vera/falsa una certa condizione**
- Il costrutto più generale per l'iterazione è **while**

```
while condizione:  
    blocco_istruzioni #body
```

```
line = input()  
while line != "stop":  
    print("hai inserito", line)  
    line = input()  
print("finito")
```

Ripete quello che inserisce
l'utente finché l'utente non
inserisce "stop"

Esempio: indovina il numero

Scrivere un programma che **chiede all'utente di indovinare un numero**, **scelto a caso tra 1 a 9**. L'utente è invitato a ritentare finché non indovina il numero

```
import random  
segreto = random.randint(1,9) # un numero a caso in [1,9]  
tenta_ancora = True  
while tenta_ancora :  
    n = int(input("Inserisci un numero tra 1 e 9: "))  
    if n == segreto:  
        tenta_ancora = False  
    else:  
        tenta_ancora = True  
print("Indovinato: ",segreto)
```

Esercizio: indovina il numero

Osserva le differenze tra il codice seguente e quello della slide precedente, e rendersi conto che sono due programmi equivalenti (Il ramo else era inutile).

```
import random
segreto = random.randint(1,9) # un numero a caso in [1,9]
tenta = True
while tenta:
    n = int(input("Inserisci un numero tra 1 e 9: "))
    if n == segreto:
        tenta = False
print("Indovinato: ",segreto)
```

Esercizio: Arricchire il comportamento del programma nei modi seguenti

1. Se il tentativo è errato, avvisa l'utente dell'errore prima di chiedere il nuovo numero.
2. Se il tentativo è errato, avvisa se il numero segreto è maggiore o minore di quello inserito.
3. Invece di scegliere un numero a caso in [1,9], il programma inizia chiedendo all'utente un numero, da memorizzare nella variabile **maximum**, che poi usa per scegliere un numero a caso in [1,**maximum**]
4. Tracciare il numero di tentativi errati effettuati dall'utente prima di indovinare il numero, e stampare il numero di tentativi quando alla fine l'utente indovina.

Esercizio

Scrivere un programma che **ripetutamente stampa il doppio del numero inserito** dall'utente. Il programma deve **terminare quando viene inserito il numero 0**, e produrre una stampa finale che indica il numero di numeri **inseriti in totale** dall'utente.

Esercizio

Scrivere un programma che usa il costrutto while per scorrere i numeri e stampare i primi 5 numeri pari.

```
n_pari = 0
i = 0
while n_pari < 5:
    if i%2==0:
        print(i)
        n_pari = n_pari +1
    i = i+1  # che succede se questa riga viene cancellata?
```

Completare il programma seguente in modo che stampi solo i primi 5 numeri pari.

```
n_pari = 0
for i in range(30):
    ???
```

while

```
line = input()
while line != "end":
    print(line)
    line = input()
print("finito")
```

Ripete quello che inserisce
l'utente finché l'utente non
inserisce "end"

Le seguenti versioni sono **equivalenti**:

```
ancora = True
while ancora:
    line = input()
    if line != "end":
        print(line)
    else:
        ancora = False
print("finito")
```

```
while True:
    line = input()
    if line != "end":
        print(line)
    else:
        break
print("finito")
```

```
while True:
    line = input()
    if line == "end":
        break
    print(line)
print("finito")
```

Esercizio

```
n = int(input("insert number"))
if n < 5:
    while not (n==5):
        print(n)
        n = n+1
print("Finished")
```

Per quali valori di input produce la stampa Finished?

- solo gli interi minori di 5
- solo 5
- solo gli interi maggiori o uguali a 5
- qualsiasi intero

Cosa stampa se l'utente inserisce 1?

- 1 2 3 4 Finished
- 4
- 1 2 3 4 5 Finished
- 5 Finished

Cosa accade se l'utente inserisce 9?

- stampa 9 10 11 12 13... loop infinito di numeri
- stampa 9 8 7 6 Finished
- stampa Finished
- stampa 9 Finished

Esercizio

1. Scrivere un programma che saluta l'utente stampando una citazione presa a caso tra un gruppetto predefinito di citazioni (sugg. rappresentare una citazione come una stringa)
2. Modificare il programma precedente facendo in modo che venga scritta un'altra citazione ogni volta che l'utente preme il tasto Enter/Return. Il programma termina quando l'utente inserisce la parola 'exit'

Esercizio

- Tracciare il flusso di controllo del seguente programma:

```
1 n = 5
2 while n > 0:
3     print(n)
4     n = n-1
5 print("Ignition!")
```

- Tracciare il flusso di controllo dei seguenti due programmi:

```
1 n_pari = 0
2 i = 0
3 while n_pari < 5:
4     if i%2==0:
5         print(i)
6         n_pari = n_pari +1
7     i = i+1
```

```
1 n_pari = 0
2 i = 1
3 while n_pari < 5:
4     if i%2==0:
5         print(i)
6         n_pari = n_pari +1
```

quiz

```
x = True
print(x or not x)
```

stampa True o False?

```
x = True
print(x and not x)
```

stampa True o False?

```
is_hungry = True
is_happy = False
print (is_hungry or (not is_hungry and is_happy))
```

stampa True o False?

```
name = 'John'
age = 24
print(name == 'John' or age==30)
```

stampa True o False?

quiz

```
in_california = True
is_raining = False
print((in_california and is_raining) or
      (in_california or ((is_raining and in_california) and
                          (not is_raining))))
```

stampa True o False?

```
x=4
y=4
z=5
print(x == y and z==y)
```

stampa True o False?

```
x=4
y=4
z=5
print(x == y or x==z)
```

stampa True o False?