Introduzione al ciclo for:

In modalità *SHELL*:

- Scrivi `for i in range(80):` (scrivi anche i due punti) dai invio

Cosa succede?

- Scrivi `print("ciao")` e dai invio

Cosa succede?

- dai invio un'altra volta

Finalmente otteniamo l'output desiderato.

3.1A) In modalità *SHELL*, scrivi le istruzioni a generare l'output mostrato a fianco:

```
1) ciao
2) ciao
3) ciao
...
```
} scritto 20 volte

3.1B) Scrivi un programma che generi l'output mostrato a fianco:

Chiama il programma *programmaFOR*

```
gradino
 gradino
  gradino
   gradino
    gradino
     gradino
      ...
```
} 20 righe, ciascun gradino sfalsato di 1 verso destra rispetto al precedente

3.2) Scrivi un programma che generi l'output mostrato a fianco:

Sovrascrivi il programma *programmaFOR*

```
XXXXXXXXXX
OXXXXXXXXX
OOXXXXXXXX
OOOXXXXXXX
OOOOXXXXXX
OOOOOXXXXX
OOOOOOXXXX
OOOOOOOXXX
OOOOOOOOXX
OOOOOOOOOX
OOOOOOOOOO
```
} 11 righe

10 colonne

3.3) Scrivi un programma che generi l'output mostrato a fianco:

Sovrascrivi il programma $programmaFOR$

5 spazi 5 spazi 5 spazi

```
1) ciao    2)ciao     3) ciao     4) ciao
5) ciao    6)ciao     7) ciao     8) ciao
9) ciao    10)ciao    11) ciao    12) ciao
13) ciao   14)ciao    15) ciao    16) ciao
17) ciao   18)ciao    19) ciao    20) ciao
21) ciao   22)ciao    23) ciao    24) ciao
25) ciao   26)ciao    27) ciao    28) ciao
29) ciao   30)ciao    31) ciao    32) ciao
33) ciao   34)ciao    35) ciao    36) ciao
37) ciao   38)ciao    39) ciao    40) ciao
41) ciao   42)ciao    43) ciao    44) ciao
```

3.4) Scrivi un programma che stampi, sulla stessa riga, i primi 10 quadrati perfetti a partire da 0 (sovrascrivi il programma $programmaFOR$). *Nota bene: in Python, l'operazione di elevamento a potenza di scrive utilizzando due asterischi:* `**` *(ad esempio* 3^7 *si scrive* `3**7`*).*

3.5) Fino ad ora abbiamo visto cicli for applicati a singole istruzioni. Vediamo se riuscite a prevedere l'effetto dei seguenti programmi:

A)

```
for a in range(10):
    print(a,end="-")
for b in range(a+1):
    print(b,end="")
```

B)

```
for a in range(10):
    print(a,end="-")
    for b in range(10):
        print(b,end="")
```

C)

```
for a in range(10):
    print(a,end="-")
    for b in range(a+1):
        print(b,end="")
```

3.6) Scrivi un programma che richieda un numero di partenza A e uno di arrivo B e che scriva tutti i numeri da A e B (entrambi compresi), separati da `", "`

3.7) Scrivi dei programmi che generino i seguenti output:

A)

```
Inserisci il numero finale di X:

X
XX
XXX
XXXX
XXXXX
XXXXXX
XXXXXXX
XXXXXXXX
```

numero di X scelto precedentemente

B)

```
Inserisci il numero iniziale di X:
```

numero di X scelto precedentemente

```
XXXXXXXXXX
XXXXXXXXX
XXXXXXXX
XXXXXXX
XXXXXX
XXXXX
XXXX
XXX
XX
X
```

C)

```
123456789
12345678
1234567
123456
12345
1234
123
12
1
```

D)

```
Inserisci il numero iniziale di spazi N:
```

N spazi

```
X         X
 X       X
  X     X
   X   X
    X X
     X
```

Siamo finalmente in grado di “fare un po’ di matematica”: il ciclo for permette infatti di definire alcune importanti funzioni aritmetiche, alcune delle quale sono nostre vecchie conoscenze di Calcolo Combinatorio:

3.8) Scrivi dei programmi che permettano di calcolare i seguenti valori (stavolta uso il nome del programma indicato)

	Input richiesto	Risultato	Nome programma
	N	$1+2+3+\cdots+N$ (cioè $\sum_{k=1}^{N} k$)	*Somma di Gauss*
A)	A, B	$A+(A+1)+(A+2)+\cdots+B$ (cioè $\sum_{k=A}^{B} k$)	*Progressione aritmetica*
B)	N	$1^2+2^2+3^2+\cdots+N^2$ (cioè $\sum_{k=1}^{N} k^2$)	*Somma dei quadrati*
C)	N, m	$1^m+2^m+3^m+\cdots+N^m$ (cioè $\sum_{k=1}^{N} k^m$)	*Somma delle potenze*
D)	N	$1\cdot 2\cdot 3\cdot \ldots \cdot N$ (cioè $N!$)	*Fattoriale*
E)	n, k	$\frac{n!}{k!\cdot(n-k)!}=\frac{\overbrace{n\cdot(n-1)\cdot(n-2)\ldots\cdot(n-k+1)}^{k\ fattori}}{\underbrace{k\cdot(k-1)\cdot(k-2)\ldots\cdot 1}_{k\ fattori}}$ (cioè $\binom{n}{k}$)	*Coefficiente binomiale*
F)	n	Calcolo dell’n-esimo numero di Fibonacci $(F_0=0; F_1=1; F_{n+2}=F_n+F_{n+1})$	*Fibonacci*
G)	N	$1!+2!+3!+\cdots+N!$ (cioè $\sum_{k=1}^{N} k!$)	*Somma di fattoriali*

Risolviamo insieme il problema di scrivere il programma per calcolare $\sum_{k=1}^{N} k$. Abbiamo bisogno di una variabile “sulla quale” sommare tutti i numeri che si creano (nel programma seguente si tratta della S).

```
N=int(input("inserisci N: "))
S=0
for k in range(1,N+1):
    S=S+k
print(S)
```

Il cuore del programma è l'istruzione `S=S+k`. Da un punto di vista matematico pare un'assurdità, in informatica però vuol dire che il "nuovo" valore di S è posto uguale al "vecchio" valore di S più k. Seguendo la stessa sintassi è possibile risolvere tutti i problemi proposti.

RIASSUNTO DELLE ISTRUZIONI VISTE

descrizione

ciclo for

```
for k in range(N):
    ...
```

Esegue una certa sequenza di istruzioni N volte, incrementando ad ogni ripetizione il valore della variabile k da 0 a N-1

```
for k in range(A,B):
    ...
```

Esegue una certa sequenza di istruzioni B-A volte, incrementando ad ogni ripetizione il valore della variabile k da A a B-1