Apri $IDLE$, vai su $FILE$ e seleziona $NEW\ FILE$. Si aprirà così una nuova finestra le cui righe non iniziano con ">>>".

- Scrivi `print("ciao")` e dai invio
- Scrivi `ciao=7` e dai invio
- Scrivi `print("ciao"*ciao)` e dai invio
- Scrivi `print()` e dai invio
- Scrivi `aslkjdhflkasjdf` (o qualcosa del genere) e dai invio
- Premi il tasto $F5$
- Scegli un percorso, chiama il programma $PROGRAMMA1$ e salva.

ESEMPIO 1) Scrivi un programma che preveda l'immissione di una parola che poi verrà stampata sullo schermo 100 volte.

```
print("Quale parola devo scrivere 100 volte?")
parola=input()
print(parola*100)
```

- Premi il tasto $F5$
- Scegli un percorso, chiama il programma $PROGRAMMA2$ e salva.

Il programma, per quanto breve, ha parecchi difetti estetici:

1) La risposta immessa dell'utente viene scritta nella riga successiva alla domanda
2) Sarebbe opportuno separare la risposta dell'utente e l'output con una riga bianca
3) Le parole ripetute sono tutte attaccate, sarebbe opportuno separarle con un ", " (cioè con una virgola separata da uno spazio)

Soluzioni:

1) Il problema risiede nel comportamento del comando $print$, che a fine esecuzione va a capo in automatico. Per controllare questo comportamento basta aggiungere alla fine (ma sempre dentro le parentesi) la scritta `,end=" "`. Il parametro end definisce il comportamento di $print$ alla fine della sua esecuzione e, per default, è posto uguale a `\n`. Scrivendo `end=" "` stiamo sostituendo l'andata a capo con uno spazio. Attenzione: questa modifica "colpisce" soltanto il $print$ dentro al quale viene scritta, la prossima chiamata a $print$ darà luogo al comportamento classico.

2.1) Correggi autonomamente i difetti 2) e 3).

2.2) Quale input verrà generato dal seguente programma immettendo il numero 20?

```
x=input("Inserisci un numero: ")
print(x*5)
```

2.3) Scrivi un programma che richieda l'immissione di un numero n e stampi sullo schermo n volte l'esclamazione "`Tombola!`", separando le successive esclamazioni con uno spazio.

2.4) Scrivi un programma che richieda l'immissione di un numero n e di una stringa p e che stampi sullo schermo n volte la stringa p.

2.5) Scrivi un programma che richieda l'immissione di due numeri n e m e che restituisca la scritta: `il prodotto dei numeri immessi è…` (con il prodotto al posto dei puntini) `ne sono certo!`

2.6) Scrivi un programma che richieda l'immissione di tre numeri a, b e c e che restituisca la scritta: `il risultato dell'espressione b*(c+2)-3a` (con i numeri immessi al posto di a, b e c) `è …` (con il risultato al posto dei puntini)

2.7) Scrivi i programmi che abbiano il funzionamento descritto in basso

A) Viene richiesto un numero n

Il programma scrive 5 righe composte ciascuna da n "O".

a fianco è mostrato il caso con $n = 20$

```
Numero di O: 20

OOOOOOOOOOOOOOOOOOOO
OOOOOOOOOOOOOOOOOOOO
OOOOOOOOOOOOOOOOOOOO
OOOOOOOOOOOOOOOOOOOO
OOOOOOOOOOOOOOOOOOOO
```

B) Viene richiesto un simbolo S
Viene richiesto il numero di colonne C
Viene richiesto il numero di righe R

Il programma disegna un rettangolo scrivendo C volte il simbolo S su R righe

a fianco è mostrato il caso con $S = "W"$, $C = 7$ e $R = 4$

```
Simbolo: W
Colonne: 7
Righe: 4

WWWWWWW
WWWWWWW
WWWWWWW
WWWWWWW
```

C) Viene richiesto un nome e un anno di nascita.

Viene restituita la frase: `Caro …` (NOME), `il 31 dicembre del 2018 avrai sicuramente` (NUMERO) `anni.`

a fianco è mostrato il caso con $NOME = "Napoleone"$, $NASCITA = 1769$.

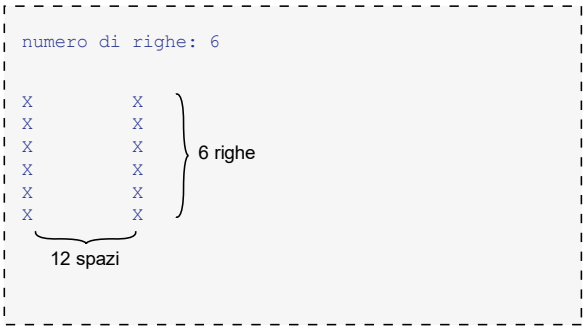
```
Nome: Napoleone
Anno di Nascita: 1769

Caro Napoleone, nel 2018 avrai compiuto 249 anni.
```

D) Vengono richiesto un numero n

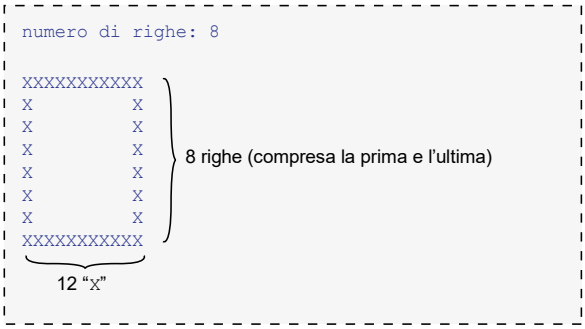
Vengono disegnate n righe fatte così: una X seguita da $2n$ spazi e poi un'altra X

a fianco è mostrato il caso con $n = 6$



E) Il problema è simile al precedente, stavolta però la figura deve essere chiusa sopra e sotto. Inoltre il numero n di righe determina in questa configurazione $2(n-2)$ spazi fra le X centrali.

a fianco è mostrato il caso con $n = 6$



RIASSUNTO DELLE ISTRUZIONI VISTE

descrizione del comando

input()

```
n=input("Scrivi qualcosa: ")
```

Stampa a video la scritta "Scrivi qualcosa: " e aspetta che l'utente inserisca una dato. Esso verrà registrato nella variabile *n* in forma di stringa.

descrizione del comando

eval()

```
n=eval(p)
```

Trasforma la stringa *p* nel numero rappresentato da *p* e lo registra nella variabile *n*. Restituisce un messaggio d'errore se *p* non è una stringa o se *p* è una stringa che contiene caratteri diversi da cifre.

descrizione del comando

int()

```
n=int(x)
```

Trasforma *x* in un numero intero troncando le cifre dopo la virgola. Funziona se l'argomento *x* è una stringa che rappresenta un numero intero o se *x* è un numero reale, restituisce un messaggio d'errore in caso contrario.

specifiche del comando

print()

- per controllare il comportamento di print a conclusione della riga di stampa, si imposta il parametro end=”...” in coda all’argomento, separato da una virgola

Esempio

```
print(”a”,end=”.X:”)
print(”b”)
```

PROGRAMMA

output →

```
a.X:b
```

- per stampare con un’unica istruzione parti fisse e variabili si separano le parti con una virgola

Esempio

```
print(”7*3=”,7*3)
```

output →

```
7*3= 21
```

- per controllare il comportamento di print nel separare parti fisse e variabili, si imposta il parametro sep=”...” in coda all’argomento, separato da una virgola

Esempio

```
print(”7*3=”,7*3,sep=””)
```

output →

```
7*3=21
```

Esempio riassuntivo

```
a=4
print("il risultato dell'operazione ",a,"*10 è ",sep="",end="...")
print(a*10,"!",sep="")
```

output →

```
il risultato dell'operazione 4*10 è ...40!
```