



## Prova Pratica del 10-02-2021

**ATTENZIONE:** al fine di consentire una corretta valutazione automatica del proprio svolgimento è necessario nell'ambiente "Programmazione" di EXAM.NET selezionare il linguaggio Python e fare uso di due programmi separati:

1. Un programma con nome **es1** per il primo esercizio,
2. Un programma con nome **es2** per il secondo esercizio.

Nel momento in cui si effettua una modifica allo svolgimento di uno dei due esercizi, **è importante non creare ulteriori programmi** ma rimpiazzare **es1** o **es2**:

1. cliccare su "Salva Programma",
2. selezionare **es1** o **es2** in base all'esercizio di cui si vuole cambiare lo svolgimento,
3. cliccare su "Rimpiazza".

**È fondamentale inoltre attenersi alle indicazioni su input e output riportate di seguito per ciascuno dei due esercizi.**

**Esercizio 1.**

Scrivere un programma Python che letta da input una sequenza di stringhe terminata da '.', stampa SI se tale sequenza, presa nell'ordine dalla prima all'ultima stringa che la compone, rappresenta la prima linea della definizione di una funzione Python, NO altrimenti.

Il programma deve stampare solo **SI/NO** senza andare a capo e senza aggiungere spazi o altre stampe.

**N.B.: si può supporre che le stringhe in input non contengano spazi bianchi.**

Si ricorda che in Python la prima linea della definizione di una funzione Python segue la seguente sintassi "def nomeFunzione(P1,..., Pn):" ovvero:

- inizia con la parola chiave "def" e termina con :
- i nomi della funzione e dei parametri (separati da virgola) devono aderire alle stesse regole che valgono per i nomi di variabili: possono contenere soltanto lettere dell'alfabeto inglese, cifre e \_ (ovvero, il carattere underscore o tratto basso) ma non possono iniziare con una cifra
- i parametri sono opzionali

**Esempi.**

Input	Output	Motivazione
def pari ( n ) :	SI	La sequenza in input rappresenta la prima linea della definizione della funzione: def pari(n):

**Prova Pratica del 10-02-2021**

<pre> pari ( n ) def : </pre>	NO	La sequenza in input è <code>pari(n)def:</code> che non rappresenta la prima linea della definizione di una funzione Python.
<pre> def lpari ( n ) : </pre>	NO	Il nome della funzione sarebbe <code>lpari</code> ma in Python il nome della funzione non può iniziare con una cifra.
<pre> def print_matrice ( N , M , MAT ) : </pre>	SI	La sequenza in input rappresenta la prima linea della definizione della funzione: <code>def print_matrice(N,M,MAT):</code>
<pre> def print_matrice ( N , M MAT ) : </pre>	NO	I parametri <code>M</code> e <code>MAT</code> non sono separati da virgola.
<pre> def print)matrice ( N , M , MAT ) : </pre>	NO	Il nome della funzione sarebbe <code>print)matrice</code> ma in Python il nome della funzione può contenere soltanto lettere dell'alfabeto inglese, cifre e <code>_</code> (underscore o tratto basso).
<pre> def print_matrice ( , N , </pre>	NO	La virgola deve separare due parametri.



## Prova Pratica del 10-02-2021

M , MAT ) :		
def mcm ( N, M , ) :	NO	La stringa "N," dopo la parentesi tonda aperta dovrebbe rappresentare un parametro, tuttavia contiene una virgola e dunque non è valido. Inoltre, non è possibile avere una virgola prima della parentesi tonda chiusa.
def stampaCiao ( ) :	SI	La sequenza in input rappresenta la prima linea della definizione della funzione: <code>def stampaCiao() :</code>
def stampa-Ciao ) :	NO	Il nome della funzione <code>stampa-Ciao</code> contiene il carattere - (ovvero il meno) che non è consentito, ed inoltre dovrebbe essere seguito da una parentesi tonda aperta.

**Esercizio 2.**

I Nonogram sono dei rompicapo logici grafici in cui le celle di una matrice devono essere colorate o lasciate in bianco in base a dei numeri a lato della matrice utili a svelare un'immagine nascosta. Un esempio di nonogram è riportato a destra, dove l'immagine nascosta è la lettera L.

I numeri a lato indicano quante celle consecutive devono essere riempite per ciascuna riga e colonna della matrice. Nell'esempio, sull'ultima riga a lato c'è "3" che significa che 3 celle consecutive devono essere colorate. Scrivere un programma Python che verifichi se un nonogram è stato risolto correttamente. Il programma presi in input nell'ordine:

	0	5	1	1	0
1					
1					
1					
1					
3					

1. un intero N positivo e maggiore di 0,
  2. una sequenza lunga  $N*N$  di 0 e 1, che rappresenta la matrice riga per riga (0 indica cella bianca, 1 indica cella colorata),
  3. una lista di N interi positivi che indicano i numeri a lato per ciascuna colonna,
  4. una lista di N interi positivi che indicano i numeri a lato per ciascuna riga,
- stampa **SI** se il nonogram è stato risolto correttamente e **NO** altrimenti.  
 Il programma deve stampare solo **SI/NO** senza andare a capo e senza aggiungere spazi o altre stampe.

**N.B.: si può supporre che l'input sia ben formato e dato nell'ordine descritto.**



## Prova Pratica del 10-02-2021

## Esempi.

Input	Output	Motivazione
5 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 1 1 1 1 0 0 5 1 1 0 1 1 1 1 3	SI	In input si avrebbe un nonogram risolto correttamente che corrisponde a quello riportato come esempio.
5 0 0 0 0 0 0 1 0 0 0 0 1 0	NO	In input si avrebbe una variante del nonogram riportato come esempio in cui la prima riga è tutta bianca, e non rappresenta un nonogram risolto correttamente.

**Prova Pratica del 10-02-2021**

0 0 0 1 0 0 0 0 1 0 0 0 1 1 1 1 0 0 5 1 1 1 0 1 1 1 1 1 3		
3 0 0 0 1 1 1 1 0 0 2 1 1 0 3 1	SI	Il nonogram in input è risolto correttamente.
4 1 0 0	NO	Il nonogram in input non è risolto correttamente.

**Prova Pratica del 10-02-2021**

0		
1		
1		
0		
0		
1		
1		
1		
0		
0		
0		
0		
1		
3		
2		
1		
1		
1		
2		
3		
0		

