

Linguaggi I2

Serie input/output

1. Scrivere un programma che chieda in input base e altezza di partenza di un rettangolo e visualizzi in output una tabella comprendente: i dati inseriti, il valore del perimetro, il valore dell'area e il rapporto area/perimetro per i 10 (oppure n da input) rettangoli successivi con base e altezza incrementati di 0.1 (oppure step da input):

| Base | Altezza | Perimetro | Area | Rapporto |
|------|---------|-----------|------|----------|
| 2.0 | 3.0 | 10.0 | 6.0 | 1.66 |
| 2.1 | 3.1 | 10.4 | 6.5 | 1.59 |
| ... | | | | |

Utilizzare le funzioni di output formattando i dati della tabella in modo coerente.

2. Scrivere un programma che determini il valore di $n!$, in cui n è un valore letto in input, ≥ 0 .

a) Mostrando in output solo il risultato finale

Es: Inserire n: -2
 Inserire n: 3

 Risultato: 6

b) Mostrando in output tutti i risultati parziali incolonnati correttamente (si può decidere di non mostrare il risultato parziale di 0 e quello del valore finale).

Es: Inserire n: 4

 Risultato parziale (1): 1
 Risultato parziale (2): 2
 Risultato parziale (3): 6
 Risultato parziale (4): 24

 Risultato finale: 24

3. Scrivere un programma che determini il valore numerico di $n!/k!$ (corrispondente a $n*(n-1)*...*(k+1)$), in cui $k < n$ e entrambi i valori sono interi positivi.

Verificare l'input prima di eseguire il calcolo.

Es: Inserire prima n poi k: 4 5
 Inserire prima n poi k: 3 -1
 Inserire prima n poi k: 5 4
 Risultato: 5

```
Inserire prima n poi k: 6 3
Risultato: 120
(4)
```

4. Scrivere un programma che, letto un numero intero da input, sia in grado di riconoscere un numero intero positivo di 4 cifre e di scomporlo nelle sue componenti.

Il programma si comporta come nell'esempio seguente:

```
Es:   Inserire numero:  26
        NO
        Inserire numero: 2638
        SI
        2   6   3   8
        Inserire numero:...
        ...
```