

# Esercizi per il Corso di Metodologie di Programmazione

## Capitolo 6

Università degli Studi di Roma "La Sapienza"

### Esercizio 1

Scrivere un programma che calcoli la somma di tutti i numeri pari da 2 a 100 - estremi inclusi - e ne stampi l'output.

**Esempio:**

*The sum is 2550*

### Esercizio 2

Scrivere un programma che permetta l'inserimento di una sequenza di numeri interi la cui fine è determinata da un carattere non numerico. Il programma dovrà calcolare e stampare a video il valore massimo all'interno di tale sequenza.

**Consiglio:** il metodo *hasNextInt* della classe *Scanner* può essere di aiuto per controllare se un valore inserito all'interno della sequenza sia gestibile come intero.

**Esempio:**

*Enter a series of numbers, then type Q to process: 8 11 9 8 2 3 100 1 1 1 99 Q*

*Largest: 100*

### Esercizio 3

Data una parola letta da input che non contenga spazi, calcolare e stampare a video il numero di vocali in essa contenute.

**Esempio:**

*Enter a word with no spaces: pippo*

*Number of vowels: 2*

### Esercizio 4

Data una sequenza di dodici temperature lette da input, ognuna appartenente ad uno specifico mese, calcolare e stampare la temperatura massima ed il suo corrispettivo mese.

**Esempio:**

**2.3 3.1 10.3 11.2 16.8 18.8 24.7 31.5 38.2 32.5 18.2 11.1 6.5**

*Highest Month: 9*  
*Highest Value: 38.2*

## Esercizio 5

Data una sequenza di numeri letta da input, terminata dall'inserimento di un carattere non numerico, trovare e stampare tutti i numeri adiacenti.

**Esempio:**

Enter a series of numbers, then type Q to process: 3 4 4 5 6 9 7 7 8 3

*Adjacent Numbers: 4 7*

## Esercizio 6

Data una sequenza di numeri letta da input, terminata dall'inserimento di un carattere non numerico, calcolare e stampare il valore minimo.

**Esempio:**

Enter a series of numbers: 3 4 5 6 1 8 9 99 8 7 Q

*The minimum value was: 1.0*

## Esercizio 7

Data una parola letta da input, stamparla a video sostituendo le vocali con il simbolo \_. Non è permesso l'uso di nessuna funzione ausiliaria.

**Esempio:**

Enter a word with no spaces: *eccebmbo*

*\_cc\_b\_mb\_*

## Esercizio 8

Data una parola letta da input, stamparla a video in modo tale che appaia un carattere per riga.

**Esempio:**

Enter a word:

*nanni*

*n*

*a*

*n*

*n*

*i*

## Esercizio 9

Data una parola letta da input, capovolgerla e stamparne il risultato. Non è ammesso l'uso di nessuna funzione.

**Esempio:**

*Enter a word to permute: **aprile***  
*Your permuted word is: elirpa*

## Esercizio 10

Data una parola letta da input, contare e stampare il numero di vocali in essa contenute.

**Esempio:**

*Enter a word: **sachertorte***  
*4 vowel(s).*

## Esercizio 11

Data una stringa letta da input, stampare tutte le possibili sottostringhe.

**Esempio:**

*Enter a word: **getout***

*g*  
*e*  
*t*  
*o*  
*u*  
*t*  
*ge*  
*et*  
*to*  
*ou*  
*ut*  
*get*  
*eto*  
*tou*  
*out*  
*geto*  
*etou*  
*tout*  
*getou*  
*etout*  
*getout*

## Esercizio 12

Creare un programma che stampi il risultato di  $2^i$ ,  $i \in [0, 20]$ .

**Esempio:** visibile nel file *output\_es\_12.txt*.

## Esercizio 13

Dato un numero letto da input, stampare la sua conversione in binario, visualizzando una cifra per riga.

**Esempio:**

*Enter a number to print in binary: 54*

0  
1  
1  
0  
1

## Esercizio 14

Creare un programma che legga da input un'immagine e la modifichi variando la colorazione del rosso per ciascun pixel.

**Consiglio:** Un approccio per la risoluzione dell'esercizio può essere il seguente:

Carico l'immagine

Per ogni riga

```
{
    Per ogni colonna di ciascuna riga
    {
        Prelevo il colore del pixel corrente
        Prelevo il parametro Red del pixel acquisito
        Modifico tale parametro a piacere
        Creo un nuovo colore contenente il parametro modificato
        Modifico il pixel dell'immagine con il nuovo colore
    }
}
```

Il caricamento dell'immagine potrà avvenire grazie al metodo *load* della classe *Picture*, fornita dall'esercizio. Tale classe possiede un metodo chiamato *getColorAt*, che fornisce un oggetto di tipo *Color* contenente i colori del pixel nel punto  $(x, y)$  in formato RGB, e un metodo *setColorAt* che assegna un oggetto di tipo *Color* ad un pixel, definito allo stesso modo tramite un punto di coordinate  $(x, y)$ .

Per ottenere il colore rosso da un oggetto di tipo *Color* è possibile utilizzare il metodo *getRed*, mentre la sua istanziazione può avvenire tramite il costruttore *Color(red, green, blue)*.

La modifica del parametro Red avverrà all'interno di un intervallo che va da 0 a 255.

**Esempio:** un esempio di output è visibile nel file *frame\_14.png*.

## Esercizio 15

Ripetere lo stesso esercizio di prima ma agendo sul colore Verde.

**Esempio:** un esempio di output è visibile nel file *frame\_15.png*.

## Esercizio 16

Stampare la tavola pitagorica per tutti i numeri che vanno da 1 a 10.

**Esempio:** visibile nel file *output\_16.txt*.

## Esercizio 17

Creare un programma che replichi l'output visibile in *output\_es\_17.txt*, richiedendo in input il lato del quadrato rappresentato nelle due versioni.

## Esercizio 18

Creare un programma che converta i Dollari in Yen. A tale scopo, è richiesto:

- Il cambio Dollaro/Yen
- La quantità di Dollari da convertire

Il programma stamperà dunque il risultato della conversione in Yen.

**Esempio:**

*Enter today's US Dollar to Japanese yen exchange rate (yen per Dollar):*

**10**

*Enter US Dollar amount to convert to Japanese yen:*

**100**

*100.000000 US Dollars equals 1000.000000 Japanese yen.*

## Esercizio 19

Creare un programma che stampi un rombo dato un valore di input che rappresenta la sua diagonale (assumendo che entrambe le diagonali siano uguali).

**Esempio:** visibile nel file *output\_19.txt*.

## Esercizio 20

Creare un programma che stampi una X tramite l'utilizzo di asterischi all'interno di un quadrato, anch'esso disegnato allo stesso modo, il cui lato è richiesto in input all'utente. Per semplicità, assumiamo che questo quadrato abbia almeno lunghezza 5.

**Esempio:** una serie di esecuzioni di tale programma è visibile nel file *output\_20.txt*.