

Programma in c utilizzando il programma Visual Studio Code

Con la simulazione odierna creeremo un programma utilizzando il linguaggio c (linguaggio di programmazione di basso livello) tramite il programma Visual Studio Code, direttamente collegato col terminale di linux che ci permetterà di lanciare il programma.

Questo programma è diviso in due parti:

- nella prima inseriremo due numeri da tastiera, con la quale il sistema dovrà calcolare e stampare l'area di tre poligoni definiti;
- nella seconda parte verranno richiesti altri due numeri da inserire da tastiera e il sistema stamperà la media dei numeri inseriti in precedenza e gli ultimi inseriti. La stampa finale sarà in due modi differenti, una arrotondando a un numero intero, e l'altra lasciando anche le cifre decimali.

In **Fig 1** e **Fig 2** possiamo vedere le stringhe di codice che abbiamo utilizzato per la creazione del programma.

In Fig 1 possiamo vedere che abbiamo utilizzato una sola libreria che è “**stdio.h**” essendo il programma abbastanza semplice da generare e con operazioni rapide, in caso ce ne fosse bisogno avremmo potuto utilizzare anche la libreria “**math.h**”, essendo più specifica può dare un aiuto con gli operatori di calcolo.

In secondo luogo abbiamo inizializzato il programma con il comando **"int main"**.

Con i comandi **float** e **int** andiamo a definire le variabili e il tipo di variabile che utilizzeremo in seguito. In questo caso abbiamo utilizzato **int** per i numeri interi e **float** per quelli decimali.

Altre istruzioni che utilizzeremo sono **“printf”** e **“scanf”** che sono usate rispettivamente per stampare a schermo e per richiedere all’ utente un dato da tastiera.

Infatti con le prime due istruzioni andremo a richiedere all'utente di inserire due numeri (definiti con le variabili `x` e `y`), mentre con gli ultimi due sulla riga **35** e **36** andremo a stampare il risultato delle operazioni, il primo con due cifre decimali e il secondo con quattro mediante l'istruzione `"%.2"` e `"%.4"`.

Con le linee “29/30/31” utilizziamo le variabili “AQ-AC-AT” per calcolare le aree che sono richieste dal nostro problema utilizzando i due numeri inseriti dall’ esterno.

```

C Primo_esercizio.c > main()
1 #include <stdio.h>
2 /*
3 il primo programma stampa le aree di tre poligoni inserendo un numero "x"
4 */
5 int main () {
6
7     float x;
8     float y;
9     float z;
10    float a;
11    float AQ;
12    float AC;
13    float AT;
14    int contatore = 0;
15    float Media;
16
17    /*inserimento da tastiera*/
18
19    printf ("Inserisci il primo numero:");
20    scanf ("%f", &x);
21
22
23    printf ("\nInserisci il secondo numero (sara' l'altezza del triangolo):\n");
24    scanf ("%f", &y);
25
26
27
28    /*calcoliamo le aree di un quadrato, un cerchio ed un triangolo*/
29
30    AQ= x*x; /*area del quadrato*/
31    AC= x*x*3.14; /*area del cerchio*/
32    AT= (x*y)/2; /*area del triangolo, abbiamo usato il secondo numero inserito per dare al sistema l'altezza*/
33
34
35    printf ("\nL'area del quadrato e': %.2f ", AQ);
36    printf ("\nL'area del cerchio e': %.4f ", AC);

```

Fig 1

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
(kali@kali)~/.Desktop/Code studio]
$ gcc -o out Primo_esercizio.c ./out rm out
(kali@kali)~/.Desktop/Code studio]
$

```

Fig 1

Fig 2

In Fig 2 abbiamo la seconda parte della simulazione dove andremo a far inserire all'operatore altri due numeri da tastiera e sommando i numeri inseriti all'inizio con quelli inseriti in seguito andremo a calcolare la media aritmetica, per permettere un calcolo corretto utilizziamo una variabile intera "contatore" per controllare che la variabile sia diversa da 0 in modo tale da non considerarla tra i valori.

```

35 printf ("\nL'area del quadrato e': %.2f ", AQ);
36 printf ("\nL'area del cerchio e': %.4f ", AC);
37 printf ("\nL'area del triangolo e': %.2f ", AT);
38
39
40 /*secondo esercizio, calcolo della media aritmetica*/
41 printf ("\n\nAdesso andremo a calcolare la media dei numeri inseriti in precedenza, e di a
42 printf ("\nInserisci il terzo numero:");
43 scanf ("%f", &z);
44
45 printf ("\nInserisci il quarto numero:");
46 scanf ("%f", &a);
47
48 if (x != 0) {
49     contatore++;
50 }
51 if (y != 0) {
52     contatore++;
53 }
54 if (z != 0) {
55     contatore++;
56 }
57 if (a != 0) {
58     contatore++;
59 }
60
61 Media= (x+y+z+a)/contatore;
62
63
64 printf("\nHai inserito %d numeri diversi da zero.\n", contatore);
65 printf("\nQuindi la media dei valori inseriti con cifre decimali e' %.4f.\n", Media);
66 printf("\nInvece la media dei valori inseriti arrotondandola e' %.0f.\n", Media);
67
68 return 0;
69

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

(kali@kali) - [~/Desktop/Code studio]
$ gcc -o out Primo_esercizio.c && ./out && rm out

```

(kali@kali) - [~/Desktop/Code studio]

Una volta completato il programma andiamo ad analizzare il risultato lanciandolo tramite il terminale mediante il comando "gcc -o out Primo_esercizio.c" && ./out rm out".

Fig 3

I vari comandi servono per:

- "gcc" per far inizializzare il compilatore;
- "-o out" per richiamare il file eseguibile e nominarlo out;
- "Primo_esercizio.c" indica il file sorgente;
- "&&" è l'operatore logico che dice al compilatore di eseguire il comando successivo SE E SOLO SE quello precedente è andato a buon fine;
- "./out" va a cercare il file out eseguibile nella cartella corrente;
- "rm out" fa in modo che il file eseguibile venga cancellato in modo da non lasciare file inutili al termine.

1 2 3 4

Terminal Help

sonomma.c programma Primo_esercizio.c X Extension: C/C++ Extension

Primo_esercizio.c > main()

```

35 printf ("\nL'area del quadrato e': %.2f ", AQ);
36 printf ("\nL'area del cerchio e': %.4f ", AC);
37 printf ("\nL'area del triangolo e': %.2f ", AT);
38
39

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

(kali@kali) - [~/Desktop/Code studio]
$ gcc -o out Primo_esercizio.c && ./out && rm out

```

Come possiamo vedere dalla **Fig 4** il programma funziona correttamente ed esegue tutte le istruzioni senza nessun errore.

Fig 4

```
(kali@kali)-[~/Desktop/Code studio]
$ gcc -o out Primo_esercizio.c && ./out && rm out
Inserisci il primo numero:8

Inserisci il secondo numero (sara' l'altezza del triangolo):
6

L'area de el quadrato e': 64.00
L'area de el cerchio e': 200.9600
L'area de el triangolo e': 24.00

Adesso andremo a calcolare la media dei numeri inseriti in precedenza, e di altri valori che inseriremo adesso.

Inserisci il terzo numero:4

Inserisci il quarto numero:0

Hai inserito 3 numeri diversi da zero.

Quindi la media dei valori inseriti con cifre decimali e' 6.0000.

Invece la media dei valori inseriti arrotondandola e' 6.

(kali@kali)-[~/Desktop/Code studio]
$
```