

INDICE.....	1
INTRODUZIONE ED OBIETTIVO.....	2
INSERIMENTO DELLE LIBRERIE.....	3
DICHIARAZIONE DELLE VARIABILI.....	4
INSERIMENTO DATI (INPUT).....	5
CALCOLI DELLE AREE.....	6
OUTPUT DEI RISULTATI.....	7
CHIUSURA DEL PROGRAMMA.....	8
VISUALIZZAZIONE DEI RISULTATI.....	9
CONCLUSIONE.....	10

INTRODUZIONE ED OBIETTIVO

INTRODUZIONE: L'esercizio rappresenta uno dei primi approcci pratici alla programmazione in linguaggio C e consente di familiarizzare con la sintassi del linguaggio, in particolare con la dichiarazione delle variabili e l'uso delle librerie standard (**stdio.h** e **math.h**) .

Attraverso la scrittura e l'esecuzione del codice si è imparato a utilizzare la funzione **printf()** per visualizzare messaggi e risultati a schermo, e la funzione **scanf()** per acquisire dati dall'utente.

Inoltre, si è imparato a gestire correttamente le operazioni matematiche con numeri decimali (tipo **double**) e ad applicare formule per calcolare aree di figure geometriche diverse in modo automatico.

OBIETTIVO: L'obiettivo di questo esercizio è scrivere un programma in linguaggio C che, dato un numero reale inserito da tastiera, calcoli e visualizzi l'area del **quadrato**, del **cerchio** e del **triangolo equilatero**. 25.10.2025

```
1
2  #include <stdio.h> /* include libreria di i/o */
3  #include <math.h> /* include libreria matematica per svolgere l'esercizio */
4  #define M_PI 3.14159265358979323846 /* definizione manuale di pi greco */
5
6  int main () {
7      /* DICHIARAZIONI VARIABILI */
8      double D; /* variabile input*/
9      double raggio; /* raggio del cerchio */
10     double area_quadrato, area_cerchio, area_triangolo; /* variabile output*/
11
12     /* INPUT */
13     printf("Inserisci un numero reale") ;
14     scanf("%lf", &D); /* Lettura del numero, %lf per double */
15
16     /* CALCOLI */
17     area_quadrato = D*D; /* calcolo area lato x lato */
18     raggio = D / 2.0 ;
19     area_cerchio = M_PI * raggio * raggio; /* Cerchio: area =  $\pi * r^2$ , dove  $r = D / 2$  */
20
21     area_triangolo = (sqrt(3.0) / 4.0) * D * D; /* Triangolo equilatero: area =  $(\sqrt{3} / 4) * lato^2$  */
22
23     /* OUTPUT */
24     printf("\n--- RISULTATI ---\n");
25     printf("Area quadrato: %.2f\n", area_quadrato); /* visualizzazione area quadrato*/
26     printf("Area cerchio: %.2f\n", area_cerchio); /* visualizzazione area cerchio*/
27     printf("Area triangolo: %.2f\n", area_triangolo); /* visualizzazione area triangolo*/
28
29     /* CHIUSURA */
30     return 0; /* Fine programma, tutto OK ( si spera )*/
31 }
32
```

INSERIMENTO DELLE LIBRERIE

```
#include <stdio.h> /* include libreria di i/o */
#include <math.h> /* include libreria matematica per svolgere l'esercizio */
#define M_PI 3.14159265358979323846 /* definizione manuale di pi greco */
```

In questa prima fase del programma sono state inserite le librerie fondamentali necessarie per permettere al codice di eseguire correttamente le operazioni di input/output e di utilizzare le funzioni matematiche.

In particolare, la libreria **#include <stdio.h>** serve per gestire l'interazione con l'utente tramite le funzioni **printf()** e **scanf()** cui consentono rispettivamente di visualizzare messaggi a schermo e di leggere valori inseriti da tastiera. Successivamente è stata aggiunta la libreria **#include <math.h>**, indispensabile per lo svolgimento dell'esercizio in quanto contiene funzioni matematiche avanzate come la radice quadrata **sqrt()**, utilizzata più avanti nel calcolo dell'area del triangolo equilatero.

Infine, poiché il valore della costante π (pi greco) non è sempre riconosciuto in automatico dal compilatore, è stata definita manualmente tramite la direttiva

#define M_PI 3.14159265358979323846, così da garantire la compatibilità del programma su qualsiasi sistema e assicurare un'elevata precisione nei calcoli.

DICHIARAZIONE DELLE VARIABILI

```
int main () {  
    /* DICHIARAZIONI VARIABILI */  
    double D;      /* variabile input*/  
    double raggio; /* raggio del cerchio */  
    double area_quadrato, area_cerchio, area_triangolo; /* variabile output*/  
}
```

In questa fase del laboratorio vengono dichiarate tutte le variabili necessarie al calcolo delle aree.

All'interno della **funzione main()** è stato definito il tipo di dato da utilizzare(in questo caso double) il quale permette di gestire numeri reali con la parte decimale garantendo una maggiore precisione nei risultati matematici.

La *variabile D* rappresenta l'input fornito dall'utente, ovvero il valore del lato o del diametro che verrà inserito da tastiera.

Successivamente è stata dichiarata la *variabile raggio*, utile per calcolare l'area del cerchio dividendo il diametro per due.

Infine, sono state definite le tre variabili di output **area_quadrato**, **area_cerchio** e **area_triangolo**, che conterranno i risultati finali dei rispettivi calcoli geometrici.

Questa sezione è importante perché stabilisce la struttura logica del programma e specifica quali dati dovranno essere gestiti e con quale tipo numerico.

Senza una corretta dichiarazione delle variabili, il compilatore non sarebbe in grado di riservare lo spazio necessario né di interpretare correttamente le operazioni successive.

INSERIMENTO DATI (INPUT)

```
/* INPUT */  
printf("Inserisci un numero reale") ;  
scanf("%lf", &D); /* Lettura del numero, %lf per double */
```

In questo momento viene gestita l'interazione diretta con l'utente, permettendo di inserire da tastiera il valore numerico necessario per i calcoli successivi.

La funzione **printf()** viene utilizzata per visualizzare sullo schermo il messaggio *"Inserisci un numero reale"*, che funge da istruzione per l'utente.

Subito dopo, la funzione **scanf()** permette di leggere il valore digitato e di memorizzarlo all'interno della *variabile D*.

Il formato **%lf** specifica che l'input atteso è di tipo *double*, quindi un numero reale che può contenere decimali.

L'*operatore &* davanti alla *variabile D* serve per passare al programma l'indirizzo di memoria dove salvare il valore inserito.

Questo passaggio consente di acquisire il dato di partenza da utilizzare nei calcoli delle aree: senza il programma non avrebbe alcun valore su cui eseguire le operazioni matematiche.

CALCOLI DELLE AREE

```
/* CALCOLI */
area_quadrato = D*D; /* calcolo area lato x lato */
raggio = D / 2.0 ;
area_cerchio = M_PI * raggio * raggio; /* Cerchio: area =  $\pi * r^2$ , dove  $r = D / 2$  */

area_triangolo = (sqrt(3.0) / 4.0) * D * D; /* Triangolo equilatero: area =  $(\sqrt{3} / 4) * lato^2$  */
```

In questa sezione il programma elabora i dati inseriti per calcolare le aree delle tre figure geometriche: quadrato, cerchio e triangolo equilatero.

Il primo calcolo riguarda il **quadrato**, dove l'area viene ottenuta moltiplicando il lato per se stesso, secondo la **formula $area_quadrato = D * D$** ;

Successivamente viene calcolato il **raggio del cerchio** dividendo il diametro per due con l'istruzione **$raggio = D / 2.0$** ;

Una volta trovato il raggio, si procede con la formula dell'area del cerchio **$area_cerchio = M_PI * raggio * raggio$** ;, dove *M_PI* rappresenta il valore di π definito manualmente nella parte iniziale del codice.

Infine, viene calcolata l'**area del triangolo equilatero** mediante la formula **$area_triangolo = (sqrt(3.0) / 4.0) * D * D$** ;

In questa fase vengono quindi applicate le principali formule geometriche, trasformando il valore fornito dall'utente in tre risultati distinti.

Questo passaggio rappresenta il cuore logico, in quanto effettua le elaborazioni numeriche vere e proprie utilizzando le funzioni matematiche incluse precedentemente nella libreria **<math.h>**.

OUTPUT DEI RISULTATI

```
/* OUTPUT */
printf("\n--- RISULTATI ---\n");
printf("Area quadrato:   %.2f\n", area_quadrato); /* visualizzazione area quadrato*/
printf("Area cerchio:    %.2f\n", area_cerchio); /* visualizzazione area cerchio*/
printf("Area triangolo:  %.2f\n", area_triangolo); /* visualizzazione area triangolo*/
```

In questa parte del programma vengono mostrati a video i risultati finali dei calcoli effettuati.

La funzione **printf()** viene utilizzata per stampare sullo schermo le tre aree calcolate in precedenza: quella del quadrato, del cerchio e del triangolo equilatero.

Per migliorare la leggibilità, il testo è stato formattato con due cifre decimali utilizzando il **formato %.2f**, che indica a **printf** di visualizzare solo due numeri dopo la virgola; in questo modo i risultati vengono presentati in una forma più chiara e precisa.

Ogni istruzione printf riporta una descrizione testuale seguita dal valore della variabile corrispondente, ad esempio *“Area quadrato: ...”* o *“Area cerchio: ...”*.

L’obiettivo di questa fase è permettere all’utente di visualizzare immediatamente l’esito dei calcoli effettuati, rendendo il programma completo dal punto di vista funzionale e comunicativo.

CHIUSURA DEL PROGRAMMA

```
/* CHIUSURA */  
return 0; /* Fine programma, tutto OK ( si spera )*/  
}
```

L'ultima parte del codice ha la funzione di concludere correttamente l'esecuzione del programma.

Il comando **return 0;** segnala al sistema operativo che il programma è terminato senza errori, restituendo un valore di uscita pari a zero, che rappresenta il successo dell'esecuzione.

Questa chiusura è essenziale in ogni programma scritto in linguaggio C, poiché conferma che il flusso logico si è concluso come previsto e che la memoria allocata viene rilasciata in modo ordinato.

In sintesi, questo passaggio conclude il ciclo completo di input, elaborazione e output, garantendo che l'intero programma funzioni in modo coerente e stabile.

VISUALIZZAZIONE DEI RISULTATI FINALI

```
--- RISULTATI ---  
Area quadrato: 25.00  
Area cerchio: 19.63  
Area triangolo: 10.83
```

Dopo l'inserimento del valore richiesto e l'elaborazione delle formule matematiche, il programma restituisce sullo schermo i risultati delle tre aree calcolate.

Come mostrato vengono visualizzati i seguenti risultati:

Area quadrato = 25.00,

Area cerchio = 19.63

Area triangolo = 10.83.

Ogni valore è espresso con due cifre decimali, come definito nella formattazione del comando printf.

Questa fase rappresenta la verifica pratica del corretto funzionamento dell'intero programma, poiché consente di osservare l'esito delle operazioni matematiche e confermare la coerenza dei risultati rispetto alle formule geometriche applicate.

Il terminale mostra dunque un output ordinato e facilmente leggibile, confermando che le variabili sono state gestite correttamente e che il programma è in grado di elaborare e visualizzare dati numerici in modo preciso e affidabile.

CONCLUSIONE

Il programma si è concluso correttamente restituendo i risultati attesi, dimostrando la piena comprensione delle principali strutture di base del linguaggio C.

Attraverso questo esercizio è stato possibile consolidare concetti come l'inclusione delle librerie, la dichiarazione e l'uso delle variabili, la gestione dell'input e dell'output, e l'applicazione di formule matematiche mediante operatori aritmetici e funzioni specifiche della libreria `<math.h>`.

La verifica del risultato finale ha confermato la correttezza dei calcoli e il corretto flusso logico del programma, che esegue con precisione tutte le operazioni richieste. Dal punto di vista didattico, questo esercizio rappresenta un'importante introduzione al linguaggio C e alla logica di programmazione strutturata, consentendo di comprendere come un problema matematico possa essere tradotto in una sequenza di istruzioni eseguibili.