Input/output

Titolo unità didattica: Introduzione al linguaggio C [03]

Titolo modulo : Input / Output in C [05-C]

Operazioni elementari di lettura da tastiera e visualizzazione in C

Argomenti trattati:

- ✔ operazione di visualizzazione in C: **`printf`**
- ✔ codici di formato in C
- ✔ operazione di lettura da tastiera in C: **`scanf`**
- ✔ **`I/O`** di caratteri in C

Prerequisiti richiesti: AP-03-04-C

Operazioni di input e output (i/o)

<stdio.h> libreria standard input e output

printf()

operazione di output

stampa a video l'argomento

scanf()

operazione di input

richiede di inserire un valore

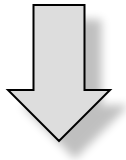
printf()

operazione di **output** in C

```
printf("Questo e' il mio primo programma C\n")
```

```
printf("valore di eta=%d/n",eta);
```

```
printf(<stringa di controllo>,<variabili>);
```



produce la visualizzazione dei **valori** delle **<variabili>**

la **<stringa di controllo>** contiene i **codici di formato** per i valori delle **<variabili>**

specificano le modalità di **conversione** tra la **rappresentazione interna** di un valore e la **rappresentazione** (notazione) **di visualizzazione**

Indicatore di conversione	**Descrizione**
%c	carattere
%d	decimale con segno
%i	intero decimale con segno (funziona diversamente con scanf)
%u	int in notazione decimale senza segno
%s	stringa
%o	ottale senza segno
%x o %X	esadecimale senza segno (x visualizzerà le lettere minuscole , X le maiuscole)
%h o %l	per gli interi indicano short e long

Proviamo

```
printf("%d\n", 455);
printf("%i\n", 455);
printf("%d\n", +455);
printf("%d\n", -455);
printf("%hd\n", 3200);
printf("%ld\n", 20000000000);
printf("%o\n", 455);
printf("%u\n", 455);
printf("%u\n", -455);
printf("%x\n", 455);
printf("%x\n", 455);
```

```
printf("%d\n", 455);                455
printf("%i\n", 455);                455
printf("%d\n", +455);               455
printf("%d\n", -455);               -455
printf("%hd\n", 3200);              3200
printf("%ld\n", 20000000000);       2000000000
printf("%o\n", 455);                707
printf("%u\n", 455);                455
printf("%u\n", -455);               4294966841
printf("%x\n", 455);                1c7
printf("%x\n", 455);                1c7
```

specificano le modalità di **conversione** tra la **rappresentazione interna** di un valore e la **rappresentazione** (notazione) **di visualizzazione**

Indicatore di conversione	**Descrizione**
%f	float o double in notazione [–]m.n
%e o %E	float o double in notazione esponenziale [–]m.nE[+-]xx
%L	indica un valore in virgola mobile long double

La notazione esponenziale:

$150{,}4582 = 1{,}504582 \times 10^{2} = 1{,}504582E+02$

Proviamo

```
printf("%e\n", 1234567.89);
printf("%e\n", +1234567.89);
printf("%e\n", -1234567.89);
printf("%E\n", 1234567.89);
printf("%f\n", 1234567.89);
```

Proviamo

```
printf("%e\n", 1234567.89);
printf("%e\n", +1234567.89);
printf("%e\n", -1234567.89);
printf("%E\n", 1234567.89);
printf("%f\n", 1234567.89);
```

```
1.234568e+06
1.234568e+06
-1.234568e+06
1.234568E+06
1234567.890000
```

codici di formato in C

il **numero** di **codici di formato** nella **<stringa di controllo>** deve essere **uguale** al **numero di variabili** in **<variabili>**, cioè deve essere **uguale** al **numero di valori** da visualizzare

la **corrispondenza** tra **codice di formato** nella **<stringa di controllo>** e **relativa variabile** in **<variabili>** è per **posto**

la **posizione** nella **<stringa di controllo>** di un codice di formato indica la **posizione** dove viene visualizzato nella riga dello schermo il corrispondente valore della **<variabile>**

Esempio

```
int miglia, km;

miglia = 1534;

km = 97;

printf("%d\n %d\n", miglia, km);
```

```
1534
 97
_
```

```
float lun_maratona = 42.195F;

printf(“La maratona e’ lunga %f chilometri\n”,
lun_maratona);
```

```
La maratona e’ lunga 42.195000 chilometri
_
```

codici di formato in C

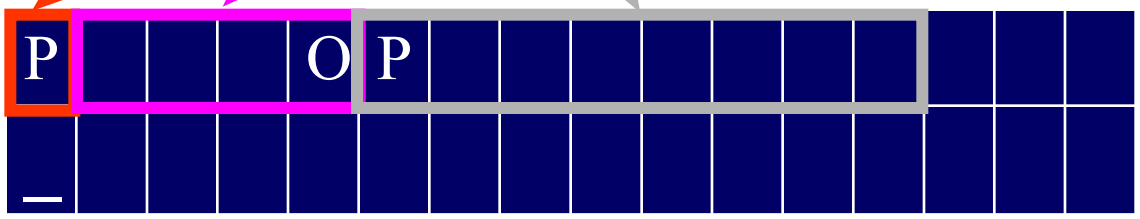
forma generale

`%` `-` `n` `.` `m` **carattere di conversione**

`%`	**carattere di inizio codice di formato**
`-`	**allineamento a sinistra**
`n`	**ampiezza del campo di visualizzazione**
`.`	**separatore**
`m`	per una stringa **massimo numero di caratteri da visualizzare** per un `int` **massimo numero di cifre da visualizzare** per un `float` **massimo numero di cifre da visualizzare dopo il punto frazionario**

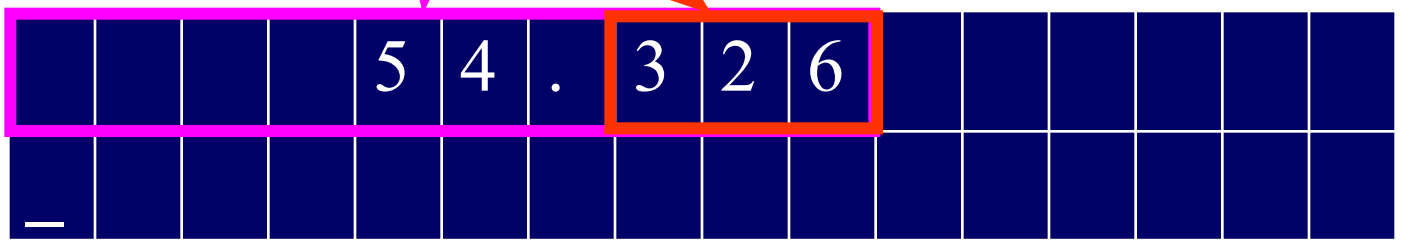
Esempio

```
char c1,c2;
c1 = 'P';
c2 = 'O';
printf("%c %4c %-8c\n", c1, c2, c1);
```



Esempio

```
float x;
x = 54.32572F;
printf("%10.3f\n", x);
```




```
printf("%10.5f\n", x);
```

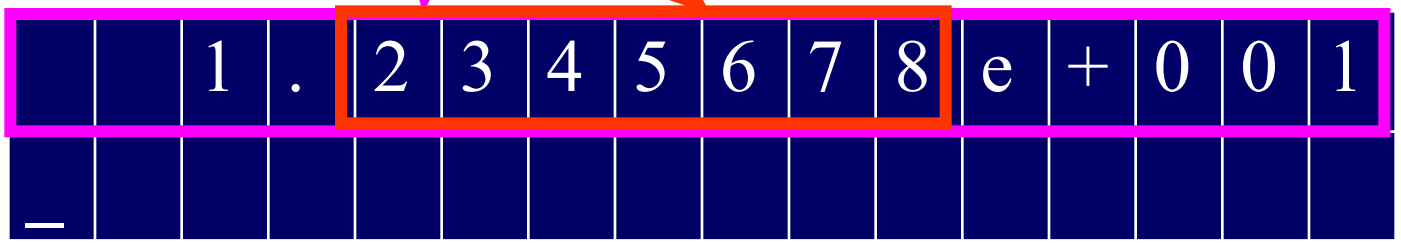
Proviamo

```
float x;
x = 54.3257F;
printf("%.3f\n",x);
printf("%10.3f\n",x);
```

```
54.326
    54.326
```

Esempio

```
float x = 12.345678F;
printf("%16.7e\n", x);
```




```
double xx = 12.34567890123456;
printf("%22.15e\n%22.15f\n", xx,xx);
```

```
1.234567890123456e+001
    12.345678901234560
```

scanf()

istruzione di input

scanf()

scanf è la funzione duale di printf
legge da input (tastiera) un valore intero e lo assegna alla variabile base

&base indica (l'indirizzo del)la locazione di memoria associata a base
scanf memorizza in tale locazione il valore letto
quando viene eseguita scanf il programma si mette in attesa che l'utente immetta un valore. Quando l'utente digita Invio
1. la sequenza di caratteri immessa viene convertita in un intero (formato %d) e
2. l'intero ottenuto viene assegnato alla variabile base (viene cioé scritto nella/e cella/e di memoria a partire dall'indirizzo passato a scanf)

N.B. il precedente valore della variabile base va perduto

operazione di **input** in C

Esempio

```
#include <stdio.h>
void main()
{
   int x;
   printf("Inserire un intero: ");
   scanf ("%d", &x);
   printf ("valore inserito = %d\n", x);
}
```

```
Inserire un intero: 21
valore inserito = 21

_
```

premere sul tasto Enter (Invio)

operazione di **input** in C

Esempio

```
#include <stdio.h>
void main()
{
   int x,y;
   printf("Inserire due interi: ");
   scanf ("%d%d", &x,&y);
   printf ("primo valore inserito = %d\n
          secondo valore inserito = %d\n",x,y);
}
```

```
Inserire due interi: 321 654
primo valore inserito = 321
secondo valore inserito = 654
```

premere sul tasto Enter (Invio)

operazione di **input** in C

Esempio

```
#include <stdio.h>
void main()
{
   int x,y;
   printf("Inserire due interi: ");
   scanf ("%d%d", &x,&y);
   printf ("primo valore inserito = %d\n",x);
   printf ("secondo valore inserito = %d\n",y);
}
```

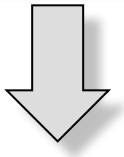
```
Inserire due interi: 321 654
primo valore inserito = 321
secondo valore inserito = 654
```

operazione di **input** in C

```
scanf ("%d", &x);
```

```
scanf ("%d%d", &x,&y);
```

```
scanf(<stringa di controllo>,<indirizzi variabili>);
```



sospende l'esecuzione del programma e **resta in attesa dell'immissione da tastiera** dei **valori** delle **<variabili>**

la **<stringa di controllo>** contiene i **codici di formato** per i valori delle **<variabili>**

operazione di **input** in C

Esempio

```
#include <stdio.h>
void main()
{
 float x, y;
 printf("Inserire due numeri reali:\n");
 scanf("%f%f",&x,&y);
 printf(" x =%f\n y =%f\n",x,y);
}
```

```
Inserire due numeri reali :
1.0
2.45
 x = 1.000000
 y = 2.450000
_
```

operazione di **input** in C

Esempio

```
#include <stdio.h>
void main()
{
 float x, y;
 printf("Inserire due numeri reali:\n");
 scanf("%f%f",&x,&y);
 printf(" x =%f\n y =%f\n",x,y);
}
```

```
Inserire due numeri reali :
54.3257
256.67543E2
 x = 54.325699
 y = 25667.542969
_
```

operazione di **input** in C

Esempio

```
#include <stdio.h>
void main()
{
 float x, y;
 printf("Inserire due numeri reali:\n");
 scanf("%f%f",&x,&y);
 printf(" x =%20.3f\n y =%12.5f\n",x,y);
}
```

```
Inserire due numeri reali :
54.3257
256.67543E2
 x =        54.326
 y = 25667.54297

_
```

operazione di **input** in C

Esempio

```
#include <stdio.h>
void main()
{
 float x, y;
 printf("Inserire due numeri reali:\n");
 scanf("%f%f",&x,&y);
 printf(" x =%15.7e\n y =%15.7e\n",x,y);
}
```

```
Inserire due numeri reali :
54.3257
256.67543E2
 x = 5.4325699e+001
 y = 2.5667543e+004

_
```

operazione di **input** in C

Esempio

```
#include <stdio.h>
void main()
{
 int x;
 char y;
 printf("Inserire un intero e un carattere: ");
 scanf ("%d%c",&x,&y);
 printf ("x = %d e y = %c \n", x, y);
}
```

```
Inserire un intero e un carattere: 21A
x = 21 e y = A

_
```

operazione di **input** in C

Esempio

```
#include <stdio.h>
void main()
{
 char x,y;
 printf("Inserire due caratteri: ");
 scanf ("%c%c",&x,&y);
 printf ("x = %c e y = %c \n", x, y);
}
```

```
Inserire due caratteri: GH
x =
 e y = G

_
```

?

Operazioni di input e output (i/o)

<stdio.h> libreria standard input e output

putchar()

operazione di output

di un singolo carattere

getchar()

operazione di input

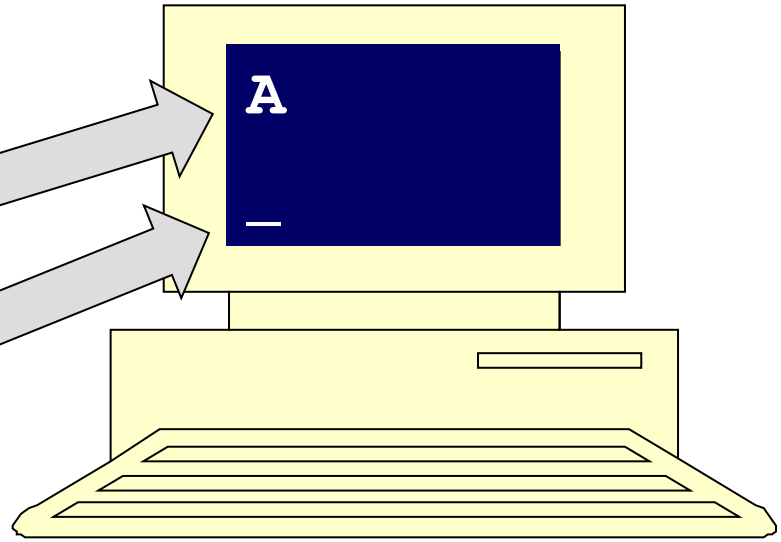
di un singolo carattere

operazione di **output** di un singolo carattere

```
putchar(c);
```

Esempio

```
char c = 'A';
putchar(c);
putchar("\n");
```

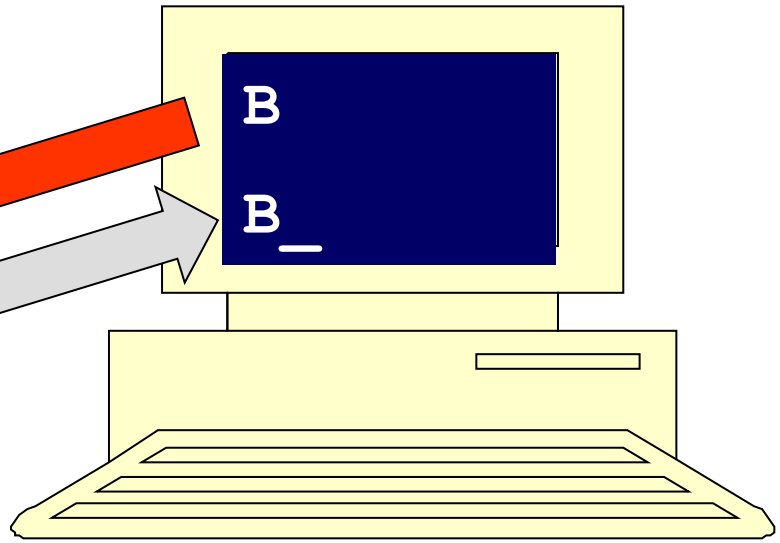


operazione di **input** di un singolo carattere

```
c = getchar();
```

Esempio

```
char c ;
c = getchar();
putchar(c);
```



Proviamo getchat e putchar

```
char carattere;
printf("Inserisci un carattere \n");
carattere = getchar();

putchar(carattere);

printf("\n%c - il tuo carattere \n", carattere);
```

Esercizi

Esempio
calcolo e visualizzazione della circonferenza di un cerchio, dato (lettura da tastiera) il suo raggio

```
const float pi_greco = 3.1415926F;
float raggio, circon;
scanf("%f", &raggio);
circon = 2.F*pi_greco*raggio;
printf ("raggio=%f  circonferenza=%f\n",raggio,circon);
```

```
10
raggio=10.000000  circonferenza=62.831848
```

Esempio
calcolo e visualizzazione dell'area di un rettangolo, date (lettura da tastiera) la sua base e la sua altezza

```
#include <stdio.h>
void main()
{
 int base, altezza, area;
 printf("Immettere la base del rettangolo (int): ");
 scanf("%d", &base);
 printf("\nImmettere l'altezza del rettangolo (int): ");
 scanf("%d", &altezza);
 area = base * altezza;
 printf("\n Area del rettangolo (base=%d,altezza=%d):%d\n",
     base, altezza, area);
}
```

```
Immettere la base del rettangolo (int): 5
Immettere l'altezza del rettangolo (int): 3
 Area del rettangolo (base= 5, altezza= 3):15
_
```

Proviamo

```c
int numero;

/** %d legge e scrive i numeri in base 10 **/
printf("Inserisci un numero\n");
scanf("%d", &numero);
// prova inserendo 2 8 9 16 20
printf("%d\n", numero);

/** %o  scrive i numeri in base 8**/
printf("%#o\n", numero);
/** %x scrive i numeri in base 8**/

printf("%#x\n", numero);
```

Proviamo

```c
char carattere;

printf("Inserisci un carattere\n");
scanf("%c", &carattere);
// prova inserendo c aaa
printf("%c\n", carattere);
```

Esercizio 1

Scrivere un programma che visualizzi la seguente frase

```
Corrono nuvole insistenti: corrono!

Sgocciola pioggia regolare: sgocciola!...
```

Soluzione 1

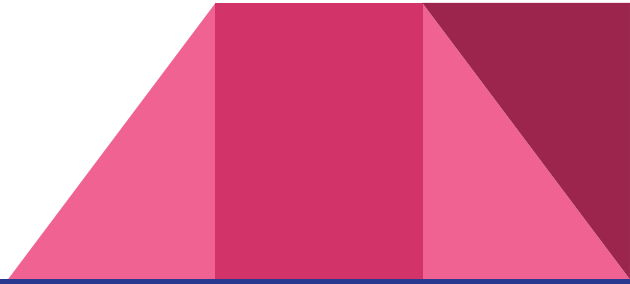
```
#include<stdio.h>

int main (){
        printf (“Corrono nuvole insistenti: corrono!\n”);
        printf(“Sgocciola pioggia regolare: sgocciola!...\n”)
        return 0;

}
```

Esercizio 2

Scrivere, esattamente come mostrato, ed eseguire il seguente codice

```
#include <stdio.h>


void main()
 {
  printf("Il mio programma in C \n");
  printf("contiene qualche errore.")
 }
```

Verificare e correggere gli errori di compilazione.



Esercizio 3

Scrivere , esattamente come mostrato, ed eseguire il seguente codice

```
#include <stdio.h>


void main()
  printf("Il mio programma in C ");  printf(" funziona?");
```

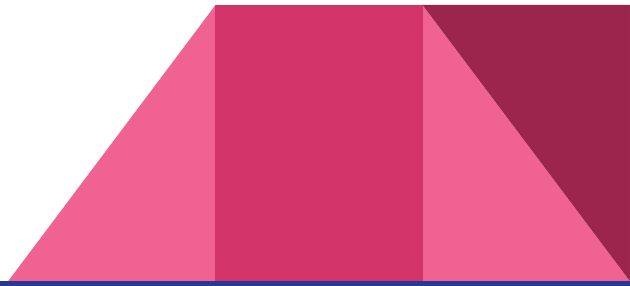
Verificare e correggere gli errori di compilazione.

Esercizio 4

Scrivere, esattamente come mostrato, ed eseguire il seguente codice

```
#include stdio.h


void main()
  printf("Il mio programma in C");
  printf("\t funziona?");
```

Verificare e correggere gli errori di compilazione.

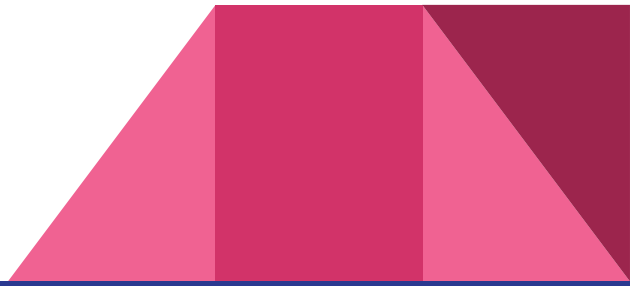


Esercizio 5

Scrivere, esattamente come mostrato, ed eseguire il seguente codice

```
#include <stdio>

void main() {printf("\t Il mio programma in C") printf("\n
funziona?"); }
```

Verificare e correggere gli errori di compilazione.



Esercizio 6

Scrivere un programma che visualizzi un quadrato di asterischi, come segue:

```
**********
*        *
*        *
*        *
*        *
*        *
**********
```

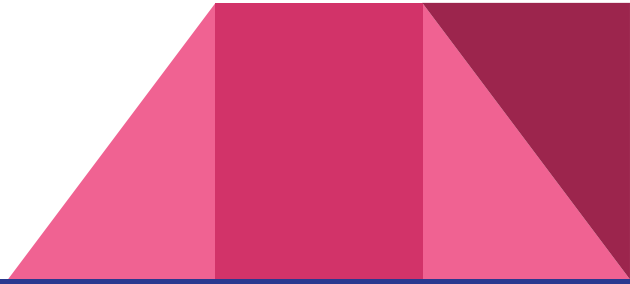
Esercizio 7

Cosa visualizza la seguente funzione?

printf(“*\n**\n***\n****\n*****\n”);

Esercizio 8

Dichiarare ed inizializzare la seguente variabile x = 12356332.34123222. Visualizzarla usando un’appropriata stringa di riferimento in printf. Usare il carattere di conversione f, per la visualizzazione come parte intera e parte frazionaria e il carattere di conversione e per la visualizzazione in notazione scientifica.



Esercizio 9

Scrivere un programma che calcoli il volume di un parallelepipedo rettangolo. La base, l'altezza e la profondità vengono scelte dall'utente e sono dati di input per il programma.

Esercizio 10

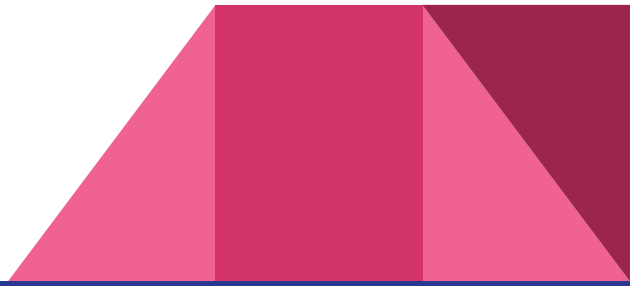
Date le seguenti operazioni, scrivere un programma che dichiari opportunamente le variabili e visualizzi i risultati delle operazioni

```
a = 6;
b = 'A';
c = 12331222;

d = 8.14245322;

e = ((c * d)^2)%10;

visualizzare a,b,c,d,e
```

Esercizio 11 v

Individuare gli errori nel seguente programma

```
#include <stdio.h>

void main () {
        const unsigned char a = 255u;
        const short b = 512;
        const unsigned long c = 10000000ul
        unsigned short x
        unsigned short y;
        x = a-b;
        y = (b-a) * c;
        printf("x risulta = %u\n",x);
        printf("y risulta = %u\n",y);

}
```

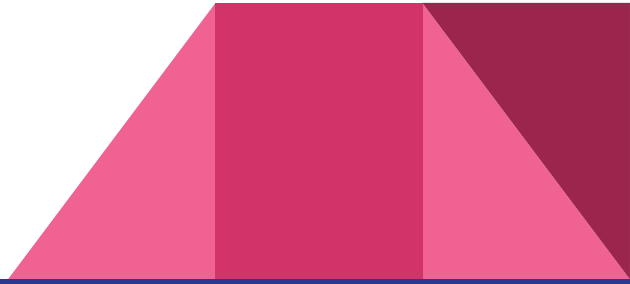


Esercizio 12

Scrivere ed eseguire il seguente programma:

```
void main() {
        const float a = 8.32453u; int b, c;
        scanf("%d",&c);
        b = a/c;
        printf ("divisione=%f \n", b);

}
```

Verificare e correggere gli errori.



Esercizio 13

Ottimizzare e correggere il seguente programma

```
#include <stdio.h>


void main()
 {
 float a;
 int b, c, d;


 a = 3;


 scanf("%d",&b);
 scanf("%d",&c);


 d = (a/c) * b;


 printf ("b=%f  \n", b);
 printf ("c=%f  \n", c);
 printf ("operazione=%f  \n", d);


}
```

Esercizio 14 v

Completare e correggere il seguente programma

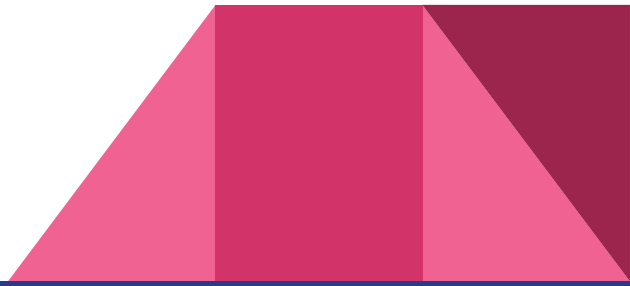
```
#include <stdio.h>


void main()
 {
  const ... a = 4194937293;
  const short b = 32765;


  ... x;


  x = a * b ;


  printf("x risulta = %15.0 ... \n", x);
}
```

Esercizio 15 v

Scrivere un programma che calcoli e stampi il risultato della seguente formula

$$\frac{x^2 + 3x}{0.5 + x} + \frac{0.1x + 1}{x^3 + x} * 5$$

dove il coefficiente x è inserito dall'utente ed è il dato di input del programma.



Esercizi di autovalutazione

Per casa

Esercizio 1

Riempire gli spazi in ognuna delle seguenti righe :

a) Ogni programma C incomincia la propria esecuzione con la funzione _____.
b) La ____ inizia il corpo di ogni funzione mentre la ___ termina il corpo di ogni funzione.
c) Ogni istruzione termina con ___
d) La funzione _____ della libreria standard visualizza le informazioni sullo schermo.
e) La sequenza di escape \n rappresenta il carattere ____ che muove il cursore nella posizione iniziale della riga successiva dello schermo.
f) La funzione _____ della libreria standard è utilizzata per leggere i dati dalla tastiera.
g) La specifica di conversione ____ è usata in una stringa di controllo del formato di una funzione scanf, per indicare che sarà immesso un intero, mentre in una stringa di controllo printf, per indicare che sarà visualizzato un intero.

Esercizio 2

Rispondere con VERO o FALSO :

a) Quando sarà invocata la funzione printf(), questa comincerà a visualizzare sempre dall'inizio di una nuova riga.
b) I commenti inducono il computer a visualizzare sullo schermo il testo racchiuso tra /* e */ durante l'esecuzione del programma.
c) La sequenza di escape \n, quando è usata in una stringa di controllo del formato di una funzione printf, induce il cursore a posizionarsi all'inizio della riga successiva dello schermo.
d) Tutte le variabili devono essere dichiarate prima di essere utilizzate
e) A tutte le variabili dovrà essere assegnato un tipo, quando saranno dichiarate.
f) il C considera identiche le variabili number e NumBer
g) Le dichiarazioni possono apparire in qualsiasi parte del corpo di una funzione
h) Tutti gli argomenti successivi alla stringa di controllo del formato in una funzione printf devono essere preceduti da un ampersand (&)
i) L'operatore modulo (%) può essere usato soltanto con degli operatori interi
j) Gli operatori aritmetici *, /, %, +, - hanno tutti la stessa priorità
k) Un programma C che visualizza 3 righe sullo schermo dovrà necessariamente avere tre printf.

Soluzioni

Soluzione 1

Riempire gli spazi in ognuna delle seguenti righe :

a) Ogni programma C incomincia la propria esecuzione con la funzione main.

b) La { inizia il corpo di ogni funzione mentre la } termina il corpo di ogni funzione.

c) Ogni istruzione termina con ;

d) La funzione printf() della libreria standard visualizza le informazioni sullo schermo.

e) La sequenza di escape \n rappresenta il carattere invio (o newline) che muove il cursore nella posizione iniziale della riga successiva dello schermo.

f) La funzione scanf() della libreria standard è utilizzata per leggere i dati dalla tastiera.

g) La specifica di conversione %d è usata in una stringa di controllo del formato di una funzione scanf, per indicare che sarà immesso un intero, mentre in una stringa di controllo printf, per indicare che sarà visualizzato un intero.

Soluzione 2

a) Falso.
b) Falso.
c) Vero.
d) Vero.
e) Vero.
f) Falso.
g) Falso.
h) Falso.
i) Vero.
j) Falso.
k) Falso.