

# Esercitazione Programmazione Procedurale con Laboratorio

## Esercizi di programmazione

### Esercizio 1      Moltiplicazione di due float

Definire due variabili con nome  $a$  e  $b$  entrambe di tipo *float* e poi stamparle su schermo (prima di stamparle, assegnare 5.678 ad  $a$  e 1.23456 a  $b$ ). Poi definire una terza variabile di tipo *double* (nome a scelta dello studente) e assegnarle il valore derivante dalla moltiplicazione di  $a * b$ . Infine, stampare questa terza variabile (formato %f).

#### Soluzione dell'esercizio 1

```
1 #include <stdio.h>
2
3 int main(void) {
4     float a = 5.678;
5     float b = 1.23456;
6
7     printf("Stampa di a: %f\n", a);
8     printf("Stampa di b: %f\n", b);
9
10    double c = a*b;
11    printf("Stampa di b: %f\n", c);
12
13    return 0;
14 }
```

### Esercizio 2      Calcolo aree

Si scriva un programma in linguaggio C che, dato un numero positivo in virgola mobile  $D$  immesso da tastiera, calcoli e stampi:

- l'area del quadrato di lato  $D$ ;
- l'area del cerchio di diametro  $D$ ;
- l'area del triangolo equilatero di lato  $D$ .

Si può utilizzare anche le funzioni *pow(base, esponente)* (esponenziazione rispetto a base), *sqrt(numero)* (radice quadrata di un numero), e la costante  $M\_PI$  (Pi greco), dichiarate in *math.h*. Per compilare con la libreria matematica aggiungere il flag *-lm* al comando *gcc*: *gcc -o eseguibile file.c -lm*.

#### Soluzione dell'esercizio 2

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4
5 int main(void) {
6
7     float d; /* numero inserito */
8     float aq, ac, at; /* area quadrato, cerchio, triangolo */
9     float r; /* raggio del cerchio */
10    float rad3_4; /* costante pari a radice(3)/4 */
11
12    rad3_4 = sqrt(3) / 4;
13    printf("Calcolo di aree\n\n");
14
15    printf("Immetti il valore di D: ");
```

```

16 scanf("%f", &d);
17
18 aq = pow(d, 2);
19
20 r = d/2;
21
22 ac = M_PI * pow(r, 2);
23 at = rad3_4 * pow( d, 2 );
24
25 printf("\n") ;
26 printf("Le aree calcolate sono:\n");
27 printf("Area del quadrato di lato %f = %f\n", d, aq);
28 printf("Area del cerchio di diametro %f = %f\n",d, ac);
29 printf("Area del triangolo equilatero di lato %f = %f\n", d, at);
30 }

```

### Esercizio 3 Swap del valore di due variabili

Leggere due valori da tastiera (*scanf()*) ed assegnarli a due variabili *x* e *y*. Stampare il valore di *x* e *y*, scambiare il loro valore (*x* prende il valore di *y* e *y* prende il valore di *x*), e ristampare il loro valore sullo schermo.

#### Soluzione dell'esercizio 3

```

1 #include <stdio.h>
2
3 int main(void) {
4     int x, y, temp;
5
6     printf("Enter the value of x and y\n");
7     scanf("%d%d", &x, &y);
8
9     printf("Before Swapping\nx = %d\ny = %d\n",x,y);
10
11    temp= x;
12    x= y;
13    y= temp;
14
15    printf("After Swapping\nx = %d\ny = %d\n",x,y);
16
17    return 0;
18 }

```

### Esercizio 4 Enum giorni settimana

Definire un tipo *enum week* che contiene i giorni della settimana a partire da *sunday* (il cui valore corrisponde a 0 quindi). Definire due variabili *today* e *tomorrow* di tipo *enum week*. Assegnare a *today* il giorno di oggi (*wednesday*), e a *tomorrow* il giorno successivo (farlo utilizzando un'operazione aritmetica, utilizzando *today + 1*). Ricordarsi che quando si arriva a *Sunday* si riparte da *Monday*: si deve quindi utilizzare l'operatore modulo della divisione intera ( $\%$ , esempio,  $5\%2 = 1$ ) nel calcolo di *tomorrow*. Stampare il valore intero corrispondente a *tomorrow* (formato di stampa per un *enum*: "%d").

In secondo luogo, sempre all'interno della stessa funzione *main()*, definire anche un *enum* senza nome che contiene solo *eastern* (Pasqua), che deve avere valore uguale al valore di *sunday* in *enum week* (cioè 0). Alla fine della solita funzione *main()*, controllare che *today* non sia *eastern* (valore differente): in pratica, controllare che oggi non possa essere Pasqua! Questo viene controllato comparando i due valori con un comando condizionale *if*. Quindi, stampare "Oggi non può essere Pasqua\n", in caso *if(eastern != today)*.

#### Soluzione dell'esercizio 4

```

1 #include <stdio.h>
2 enum week{sunday, monday, tuesday, wednesday, thursday, friday, saturday};
3
4 int main(void) {
5
6     //Esempio di tipo enum senza nome che mi serve serve per definire una costante eastern (ha valore
7     //equivalente a sunday)
8     enum {easter = 0};
9
10    enum week today, tomorrow;

```

```

10 today= wednesday;
11 tomorrow = (today + 1) % (saturday + 1);
12 printf("Tomorrow is %d\n", tomorrow);
13
14 //Scommentare se volete stampare il nome del giorno
15 /*
16 if (today == sunday)
17 printf("Today is Sunday\n");
18 else
19 if (today == monday)
20 printf("Today is Monday\n");
21 else
22 if (today == tuesday)
23 printf("Today is Tuesday\n");
24 else
25 if (today == wednesday)
26 printf("Today is Wednesday\n");
27 else
28 if (today == thursday)
29 printf("Today is Thursday\n");
30 else
31 if (today == friday)
32 printf("Today is Friday\n");
33 else
34 if (today == saturday)
35 printf("Today is Saturday\n");
36 */
37
38 // Se oggi non e' domenica, non puo' essere Pasqua!
39 if (easter != today)
40 printf("Today cannot not be Eastern\n");
41
42 return 0;
43 }

```