

PROBLEMA: Menù di scelta utente SEMPLICE

Menù di scelta semplice tra più opzioni INDIPENDENTI tra loro

TABELLE DEI DATI

1

DATI DI INPUT DEL PROBLEMA PRINCIPALE (PROCEDURA MAIN)				
Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
scelta	INT	STATICA	$(scelta \geq 0)$ AND $(scelta \leq xx)$	Numero intero rappresentante la scelta effettuata dall'utente all'interno del menù (xx dipenderà dal numero di azioni previste dal programma)
...				

2

DATI DI OUTPUT DEL PROBLEMA PRINCIPALE (PROCEDURA MAIN)				
Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
...				

3

DATI DI ELABORAZIONE o DI LAVORO DEL PROBLEMA PRINCIPALE (PROCEDURA MAIN)				
Nome variabile oppure nome costante	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
...				

PSEUDODODIFICA

ALGORITMO Menù_Scelta_Semplice

PROCEDURA main()

/ dati di input */*

scelta: **INT** (ma anche eventualmente **CHAR**)

.....

/ dati di output */*

.....

/ dati di lavoro o di elaborazione */*

.....

INIZIO

/ Menù di scelta utente */*

RIPETI

```
Scrivi ("1 Azione_1")
Scrivi ("2 Azione_2")
Scrivi ("3 Azione_3")
Scrivi ("4 Azione_4")
Scrivi (".....")
Scrivi (".....")
Scrivi ("0 USCITA")
```

1

1 Preparazione stringhe da visualizzare all'interno del menù utente

```
Scrivi("Inserisci scelta: ")
Leggi(scelta)
```

2

2 Acquisizione della scelta effettuata dall'utente

NEL CASO CHE (scelta) SIA

1 : <B1>

2 : <B2>

3 : <B3>

4 : <B4>

.....

0 : **Scrivi("Bye bye!")**

//gestione delle scelte non corrette

ALTRIMENTI :

Scrivi("Scelta non consentita")

FINE CASO

3

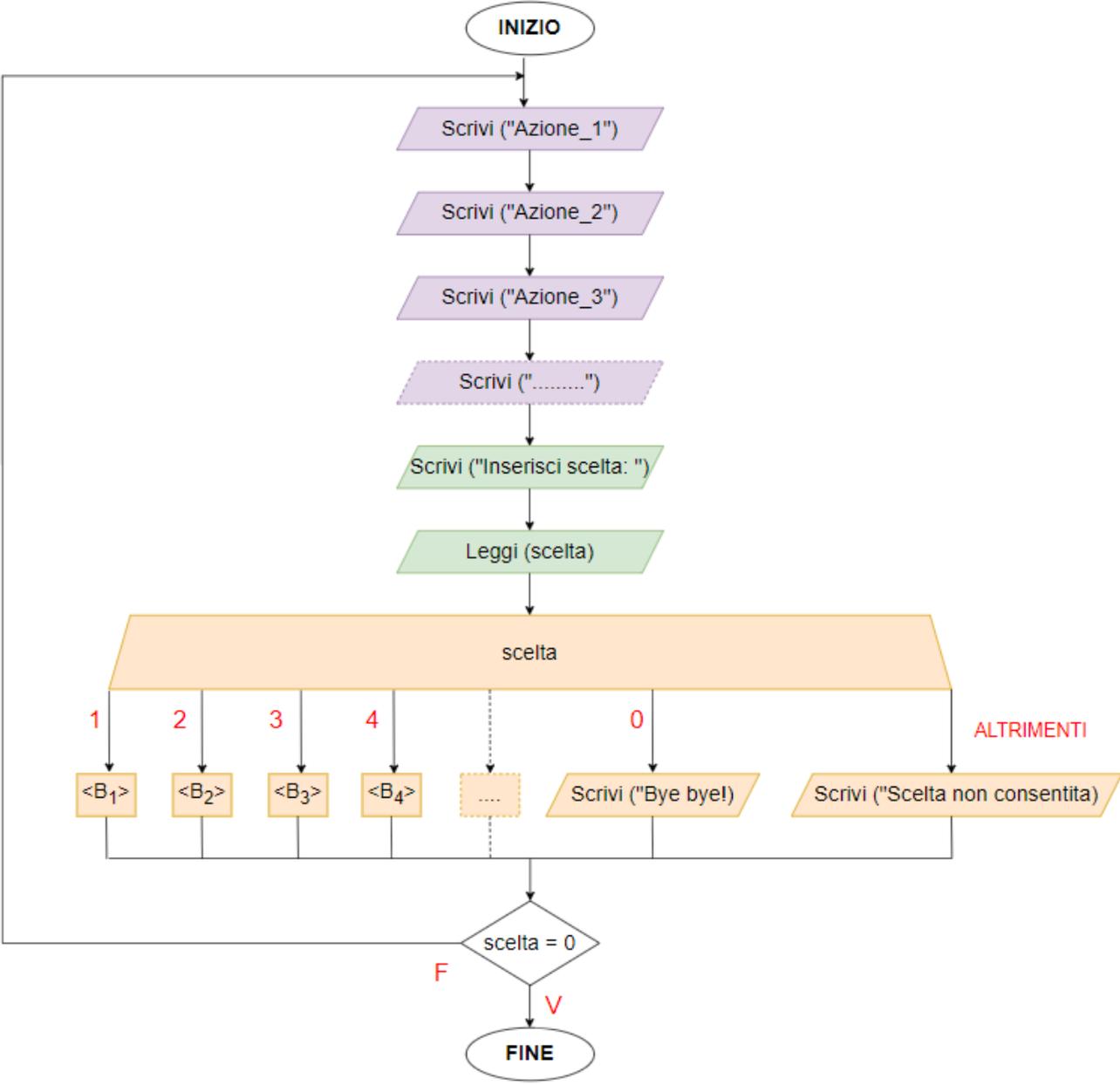
3 Gestione delle azioni da effettuare in base alla scelta effettuata dall'utente

N.B. la scelta di assegnare all'uscita il valore 0 e non il semplice progressivo rispetto all'ultima azione da intraprendere è per facilitare l'azione eventuale di ampliamento del menù

FINCHE' (scelta = 0)

FINE

FLOWCHART



CODIFICA C

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
int scelta;
/* Ciclo di gestione del menù di scelta utente */
do
{
system("CLS"); /* FONDAMENTALE per poter resettare ogni volta il menù utente*/
/* Visualizzazione menù utente ed acquisizione scelta */
printf("\n1 Azione_1");
printf("\n2 Azione_2");
printf("\n3 Azione_3");
printf("\n4 Azione_4");
printf("\n.....");
//printf("\nn Azione_n");
printf("\n0 USCITA");

printf("\n\nInserisci scelta: ");
scanf("%d",&scelta);

/* Esecuzione azione associata all'azione scelta dall'utente */
switch(scelta)
{
case 1:
{
printf("Eseguo Azione_1\n\n");
break;
}
case 2:
{
printf("Eseguo Azione_2\n\n");
break;
}
case 3:
{
printf("Eseguo Azione_3\n\n");
break;
}
case 4:
{
printf("Eseguo Azione_4\n\n");
break;
}
//eventuale aggiunta degli altri casi
...
case 0:
{
printf("Bye bye!\n\n");
break;
}
default:
{
printf("Scelta utente non consentita!\n\n");
}
}

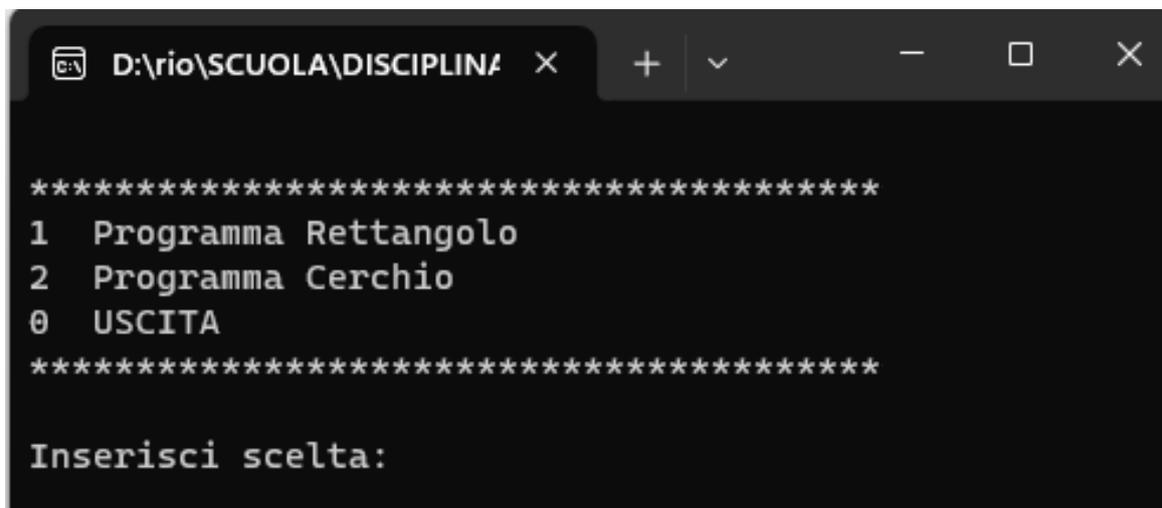
/* FONDAMENTALE per poter vedere l'esito delle azioni relative alla scelta effettuata */
system("PAUSE");
}
while (scelta != 0);
return 0;
}
```

PROBLEMA di Esempio: Menù di scelta utente SEMPLICE

MENU' GEOMETRIA

Costruire un menù di interazione utente che permette di stampare a video:

- o l'area ed il perimetro del rettangolo, assegnati base ed altezza;
- o la circonferenza ed area del cerchio, assegnato il raggio.



```
D:\rio\SCUOLA\DISCIPLINA/ x + v - □ ×  
  
*****  
1 Programma Rettangolo  
2 Programma Cerchio  
0 USCITA  
*****  
  
Inserisci scelta:
```

TABELLE DEI DATI

1

DATI DI INPUT DEL PROBLEMA PRINCIPALE (PROCEDURA MAIN)				
Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
scelta	INT	STATICA	$(scelta \geq 0) \text{ AND } (scelta \leq 2)$	Numero intero immesso da tastiera rappresentante la scelta effettuata dall'utente all'interno del menù
b	REAL	STATICA	$b > 0$	Base REALE del rettangolo immesso da tastiera
h	REAL	STATICA	$h > 0$	Altezza REALE del rettangolo immesso da tastiera
r	REAL	STATICA	$h > 0$	Raggio REALE del cerchio immesso da tastiera

2

DATI DI OUTPUT DEL PROBLEMA PRINCIPALE (PROCEDURA MAIN)				
Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
area_r	REAL	STATICA	$area > 0$	Area del rettangolo da visualizzare a video calcolata con la formula $area_r = b * h$
perimetro	REAL	STATICA	$perimetro > 0$	Perimetro del rettangolo da visualizzare a video calcolata con la formula $perimetro = 2 * (b + h)$
area_c	REAL	STATICA	$area > 0$	Area del cerchio da visualizzare a video calcolata con la formula $area_c = PIGRECO * r * r$
circonferenza	REAL	STATICA	$perimetro > 0$	Circonferenza del cerchio da visualizzare a video calcolata con la formula $circonferenza = 2 * PIGRECO * r$

3

DATI DI ELABORAZIONE o DI LAVORO DEL PROBLEMA PRINCIPALE (PROCEDURA MAIN)				
Nome variabile oppure nome costante	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
PIGRECO	REAL	STATICA	3.14	Costante che rappresenta il rapporto tra circonferenza ed il diametro di qualsiasi circonferenza

ALGORITMO Menù_Geometria

PROCEDURA main()

/* variabili di input PROCEDURA main()*/

scelta : **INT**

/* variabili di input programma Rettangolo */

b, h : **REAL**

/* variabili di output programma Rettangolo */

area_r, perimetro : **REAL**

/* variabili di input programma Cerchio */

r : **REAL**

/* variabili di output programma Cerchio */

area_c, circonferenza : **REAL**

INIZIO

/* Menù di scelta utente SEMPLICE */

RIPETI

Scrivi ("1 Programma RETTANGOLO")
Scrivi ("2 Programma CERCHIO")
Scrivi ("0 USCITA")

1

Scrivi("Inserisci scelta: ")
Leggi(scelta)

2

NEL CASO CHE (scelta) SIA

1 : /* Leggi e controlla base ed altezza del RETTANGOLO */

RIPETI

Scrivi ("Inserisci base RETTANGOLO: ")

Leggi (b)

FINCHE' (b > 0)

RIPETI

Scrivi ("Inserisci altezza RETTANGOLO: ")

Leggi (h)

FINCHE' (h > 0)

/* calcola e stampa l'area del RETTANGOLO */

area_r \leftarrow b * h

Scrivi (area_r)

/* calcola e stampa il perimetro del RETTANGOLO */

perimetro \leftarrow 2 * (b + h)

Scrivi (perimetro)

2 : /* Leggi e controlla il raggio del CERCHIO */

RIPETI

Scrivi ("Inserisci raggio CERCHIO: ")

Leggi (r)

FINCHE' (r > 0)

/* calcola e stampa l'area del CERCHIO */

area_c \leftarrow PIGRECO * r * r

Scrivi (area_c)

/* calcola e stampa la circonferenza del CERCHIO */

circonferenza \leftarrow 2 * PIGRECO * r

Scrivi (circonferenza)

0 : Scrivi("Bye bye!")

/* gestione delle scelte utente non corrette */

ALTRIMENTI :

Scrivi("Scelta utente non consentita")

FINE CASO

FINCHE' (scelta = 0)

FINE

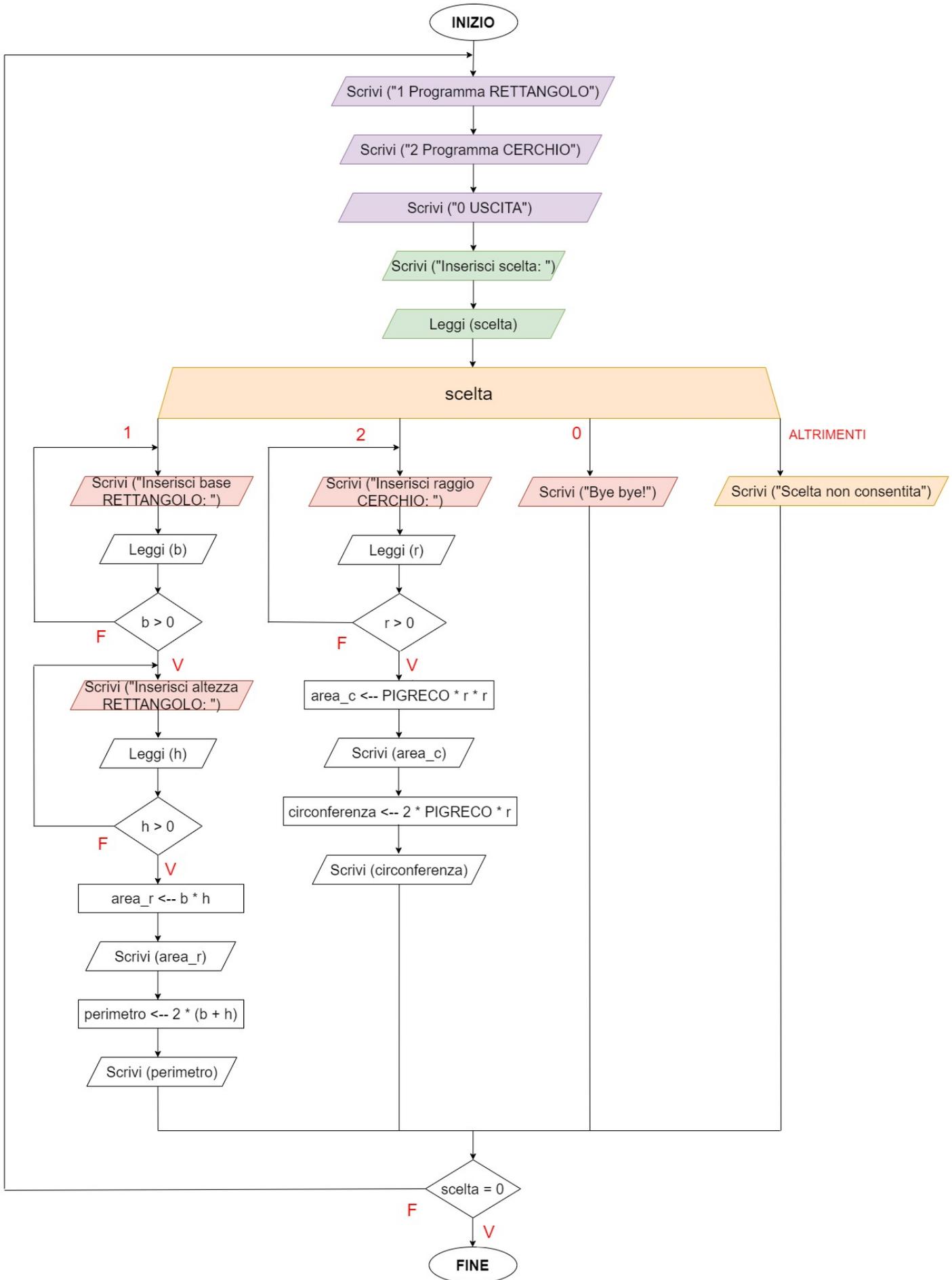
1 Preparazione stringhe da visualizzare all'interno del menù utente;

2 Acquisizione della scelta effettuata dall'utente;

3 Gestione delle azioni da effettuare in base alla scelta effettuata dall'utente ed alla propedeuticità dell'azione che deve essere rispettata.

N.B. la scelta di assegnare all'uscita il valore 0 e non il semplice progressivo rispetto all'ultima azione da intraprendere è per facilitare l'azione eventuale di ampliamento del menù

FLOWCHART



CODIFICA

```
#include <stdio.h>
#include <stdlib.h>
#define PIGRECO 3.14

int main(int argc, char *argv[])
{
    /* variabili di input PROCEDURA main()*/
    int scelta;
    /* variabili di input programma Rettangolo */
    float b, h;
    /* variabili di output programma Rettangolo */
    float area_r, perimetro;
    /* variabili di input programma Cerchio */
    float r;
    /* variabili di output programma Cerchio */
    float area_c, circonferenza;

    /* Ciclo di gestione del menù di scelta utente */
    do
    {
        /* FONDAMENTALE per poter resettare ogni volta il menù utente*/
        system("CLS");

        /* Visualizzazione menù utente ed acquisizione scelta */
        printf("\n*****");
        printf("\n1 Programma Rettangolo");
        printf("\n2 Programma Cerchio ");
        printf("\n0 USCITA");
        printf("\n*****");
        printf("\n\nInserisci scelta: ");
        scanf("%d",&scelta);
        /* Esecuzione azione associata all'azione scelta dall'utente */
        switch(scelta)
        {
            case 1:
                {
                    printf("Eseguo Programma RETTANGOLO\n\n");
                    /* Leggo e controllo la misura della base del RETTANGOLO */
                    do
                    {
                        printf("Inserisci base del RETTANGOLO: ");
                        scanf ("%f", &b);
                    }
                    while (b <= 0);
                    /* Leggo e controllo la misura della altezza del RETTANGOLO */
                    do
                    {
                        printf("Inserisci altezza del RETTANGOLO: ");
                        scanf ("%f", &h);
                    }
                    while (h <= 0);
                    /* Calcolo e visualizzo l'area del RETTANGOLO */
                    area_r = b * h;
                    printf ("L'area del RETTANGOLO e': %.2f\n", area_r);
                    /* Calcolo e visualizzo il perimetro del RETTANGOLO */
                    perimetro = 2 * (b + h);
                    printf ("Il perimetro del RETTANGOLO e': %.2f\n\n", perimetro);
                    break;
                }
            }
        }
    }
```

```

case 2:
{
printf("Eseguo Programma CERCHIO\n\n");
/* Leggo e controllo la misura del raggio del CERCHIO */
do
{
printf("Inserisci raggio: ");
scanf ("%f", &r);
}
while (r <= 0);
/* Calcolo e visualizzo l'area del CERCHIO */
area_c = r * r * 3.14;
printf ("L'area del cerchio e': %.2f\n", area_c);
/* Calcolo e visualizzo la circonferenza del CERCHIO */
circonferenza = 2 * PIGRECO * r;
printf ("la circonferenza del CERCHIO e': %.2f\n\n", circonferenza);
break;
}
case 0:
{
printf("Bye bye!\n\n");
break;
}
default:
{
printf("Scelta utente non consentita!\n\n");
}
}

/* FONDAMENTALE per poter vedere l'esito delle azioni relative alla scelta effettuata */
system("PAUSE");
}
while (scelta != 0);

return 0;
}

```