

# Progetto Week 2

## -Programmazione in C-

**Ho diviso il programma in blocchi di codici, per poi analizzarli singolarmente.**

### PARTE 1

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

```
void menu ();
```

```
void moltiplica ();
```

```
void dividi ();
```

```
void ins_string();
```

```
int main ()
```

```
{
```

```
    char scelta = {'\0'};
```

```
    menu ();
```

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

```
/*
```

**COSA FA IL CODICE:** Void è una funzione che non da nessuna funzione di ritorno. Viene usata per modificare le variabili esterne ad un programma, ma accessibili dalle funzioni del codice. Con questa funzione, vengono, pertanto, dichiarate 4 funzioni: "menu", "moltiplica", "dividi" e "ins\_string".

La funzione "menu" intenderà visualizzare un menu iniziale, mentre le funzioni "moltiplica", "dividi" e "ins\_string" effettueranno rispettivamente una moltiplicazione, una divisione e l'inserimento di una stringa.

**ERRORE SINTATTICO:** La variabile "scelta" è definita in char, quindi come carattere, ma la tipologia di argomento in scanf "%d", che identifica i n interi, non è congruente.

**SOLUZIONE:** Quando andrò a scrivere il comando di scanf dovrò sostituire la tipologia di argomento "%d" con "%c". Quest'ultima, difatti, identifica i caratteri.

\*/

## PARTE 2

```
switch (scelta)
{
    case 'A':
        moltiplica();
        break;
    case 'B':
        dividi();
        break;
    case 'C':
        ins_string();
        break;
}

return 0;

//Return indica il valore di ritorno della funzione main
}

/*
```

**COSA FA IL CODICE:** Usa il comando "switch" per verificare il valore della variabile "scelta" e, in base a quello che verrà selezionato, l'utente eseguirà una delle 3 funzioni indicate

"Break" serve a interrompere la casistica e passare alla successiva.

Non riscontro errori

\*/

### PARTE 3

```
void menu ()
```

```
{
```

```
    printf("Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni  
compiti\n");
```

```
    printf("Come posso aiutarti?\n");
```

```
    printf("A >> Moltiplicare due numeri\nB >> Dividere due numeri\nC >> Inserire una  
stringa\n");
```

```
}
```

/\*

**COSA FA IL CODICE:** Grazie a questi comandi, l'utente visualizzerà un messaggio di benvenuto e le opzioni di scelta, definite precedentemente con "switch".

Non riscontro errori

\*/

### PARTE 4

```
void moltiplica ()
```

```
{
```

```
    short int a,b = 0;
```

```
    printf("Inserisci i due numeri da moltiplicare:");
```

```
    scanf("%f", &a);
```

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

```

short int prodotto = a * b;

printf("Il prodotto tra %d e %d e': %d", a,b,prodotto);

}

/*

```

**COSA FA IL CODICE:** Moltiplica due numeri

**ERRORE SINTATTICO:** Ho riscontrato un'incongruenza tra:

- la funzione scanf dell'intero a utilizza il formato %f.
- la funzione scanf dell'intero b utilizza il formato %d
- la variabile "prodotto" è dichiarata come "short int", ma non specifica il range di a e b.

**SOLUZIONE:** nel dubbio, andrei a modificare i tipi di variabile short int con int. E, certamente, sostituirei il formato %f usato da scanf del numero a, con %d

```

*/

```

## PARTE 5

```

void dividi ()

{

    int a,b = 0;

    printf("Inserisci il numeratore:");

    scanf ("%d", &a);

    printf("Inserisci il denominatore:");

    scanf ("%d", &b);


    int divisione = a % b;

```

```
printf("La divisione tra %d e %d e': %d", a,b,divisione);  
  
}  
  
/*
```

**COSA FA IL CODICE:** Divide due numeri

**ERRORE SINTATTICO:** La funzione "dividi" nella riga "int divisione = a % b" utilizza l'operatore del modulo % che restituisce il resto della divisione.

**SOLUZIONE:** è opportuno sostituire il "%" con "/", che darà in output il quoziente esatto.

**CASISTICA CHE IL PROGRAMMA NON HA CONTEMPLATO:** Non ci sono controlli per verificare che il denominatore sia diverso da zero.

**SOLUZIONE:** Bisognerà inserire una barra di codice, che usi il costrutto if, dopo le istruzioni di scanf, come segue:

```
if (b == 0)  
{  
  
    printf("l'espressione è priva di significato\n");  
  
    return 0;  
  
}  
  
*/
```

## PARTE 6

```
void ins_string ()  
{  
  
    char stringa[10];  
  
    printf("Inserisci la stringa:");  
  
    scanf ("%s", &stringa);  
  
}
```

```
/*
```

**COSA FA IL CODICE:** Legge una stringa di 10 caratteri e la salva in un array.

**ERRORE LOGICO:** Forse è opportuno inserire un ultima riga di codice sotto scanf, come segue:

```
printf("la stringa inserita è: %s", stringa);
```

```
*/
```

### **SOLUZIONE FINALE:**

Avrei scritto il programma, con le correzioni descritte sopra e come mostro di seguito:

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

```
void menu ();
```

```
void moltiplica ();
```

```
void dividi ();
```

```
void ins_string();
```

```
int main ()
```

```
{
```

```
char scelta = {'\0'};
```

```
menu ();
```

```
scanf ("%c", &scelta);
```

```
switch (scelta)
```

```
{
```

```
    case 'A':  
        moltiplica();  
        break;  
    case 'B':  
        dividi();  
        break;  
    case 'C':  
        ins_string();  
        break;  
}  
return 0;  
}  
  
void menu ()  
{  
  
    printf ("Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni  
compiti\n");  
  
    printf ("Come posso aiutarti?\n");  
  
    printf ("A >> Moltiplicare due numeri\nB >> Dividere due numeri\nC >> Inserire  
una stringa\n");  
}  
  
void moltiplica ()  
{  
  
    int a,b = 0;  
  
    printf ("Inserisci i due numeri da moltiplicare:");
```

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

scanf ("%d", &b);

int prodotto = a * b;


printf ("Il prodotto tra i due numeri e': %d", a,b,prodotto);

}
```

```
void dividi ()

{

int a,b = 0;

printf ("Inserisci il numeratore:");

scanf ("%d", &a);

printf ("Inserisci il denominatore:");

scanf ("%d", &b);

if (b == 0)

{

printf("Errore: il denominatore non può essere 0\n");

return;

}

int divisione = a / b;

printf ("La divisione tra i due numeri e': %d", a,b,divisione);

}


void ins_string ()

{
```



```
char stringa[10];
```

```
printf ("Inserisci la stringa:");
```

```
scanf ("%s", &stringa);
```

```
printf ("la stringa inserita è: %s", stringa);
```