

```

#include <stdio.h>

void menu ();
void moltiplica ();
void dividi ();
void ins_string();

int main ()
{
    char scelta = {'\0'};
    menu ();
    scanf ("%d", &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 ()
{
    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);
}

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);
}

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

```

## DESCRIZIONE DEL PROGRAMMA

**#include <stdio.h>** include la libreria standard di input/output in modo che possiamo utilizzare le funzioni di input/output come **printf** e **scanf**.

Le **funzioni** vengono dichiarate prima del **main**, e definite dopo; questo va bene e permette di avere un codice ordinato.

Il codice di questo programma presenta all'utente un menù con una serie di opzioni tra le quali può scegliere associate ad un carattere specifico; in base al carattere inserito dall'utente (utilizzo del costrutto **switch**) può essere realizzata una moltiplicazione tra due numeri immessi dall'utente con visualizzazione del prodotto, una divisione tra due numeri sempre immessi dall'utente (specificando numeratore e denominatore) con visualizzazione del risultato della divisione, un inserimento di una stringa composta da 10 caratteri digitata dall'utente oppure nessuna tra le tre opzioni nel caso venga digitato un carattere non associato a nessuna operazione.

Si noti come una volta eseguita una delle operazioni il programma termina (utilizzo del **break** all'interno di ogni opzione dello switch).

## ERRORI LOGICI/SINTATTICI E INDIVIDUAZIONE CASISTICHE NON GESTITE DAL PROGRAMMA

All'interno del *main* il **%d** all'interno dello **scanf** è un errore logico poiché esso fa riferimento ad un numero intero mentre scelta è un char; si sarebbe dovuto usare **%c** pertanto.

All'interno della funzione *moltiplica* vi è un problema simile; le variabili a, b e prodotto sono dichiarate di tipo **short int** e pertanto lo specificatore di formato ogni volta da usare deve essere **%hd** al posto di **%d**; sarebbe comunque più opportuno logicamente dichiararle tutte di tipo **int** e usare **%d** in ogni parte del codice, poiché un tipo **int** permette di salvare un range di valori maggiore (4 byte) rispetto al **short** (2 byte).

Nella funzione *dividi* innanzitutto se si vuole fare una divisione si deve usare **/** nella sintassi e non **%** che invece restituisce il resto della divisione. Inoltre, poiché logicamente è molto probabile che il risultato di una divisione abbia dei numeri decimali non è consigliato dichiarare le variabili di tipo **int**, ma sarebbe più opportuno farlo con **float** (nel caso si dichiarino float solo la variabile divisione è necessario fare il **cast** a float di numeratore e denominatore nell'operazione di divisione); di conseguenza non va usato il **%d** ma il **%f** specifico per le variabili float. Inoltre sarebbe opportuno effettuare un controllo sul denominatore diverso da 0 prima di eseguire la divisione, restituendo un messaggio che dice che la divisione è impossibile altrimenti.

Nella funzione per l'inserimento della stringa non va utilizzato **&** nello **scanf** poiché si tratta di una stringa.

All'infuori degli errori logici o sintattici rilevati e corretti precedentemente, è possibile notare alcuni casi possibili non standard non gestiti dal programma. Per prima cosa, nel caso in cui l'utente digiti un carattere non associato a nessuna opzione sarebbe opportuno visualizzare un messaggio del tipo "Opzione non valida" e riconsegnare all'utente di immettere un nuovo carattere fino a quando non ne inserisce uno valido; in alternativa si può anche inserire nello switch un'opzione **default** che interrompe il programma e visualizza un messaggio di errore nel caso il carattere digitato non rientri tra quelli specificati nello switch, ma questo tipo di scelta dipende dall'utilizzo che se ne vuole fare del programma e dalle modalità di interazione con l'utente desiderate.

Nella funzione moltiplica solo `b` viene inizializzato a 0 immediatamente dopo essere dichiarato; è buona norma farlo per tutte le variabili.

Un'altra situazione da considerare è quella della funzione che permette di inserire una stringa da parte dell'utente che può avere massimo 10 caratteri; sarebbe opportuno far sapere all'utente tramite un messaggio visualizzato che può inserire al massimo 10 caratteri e stampare a schermo la stringa digitata. In alternativa si può anche fare in modo di prendere soltanto i primi 10 caratteri della stringa inserita dall'utente, indipendentemente dalla sua lunghezza.

Un'ultima importante osservazione riguarda la possibilità che le funzioni restituiscano un output non vuoto, per esempio il risultato della moltiplicazione o della divisione o la stringa digitata nelle funzioni possono essere fornite in output e salvate in una variabile nel main, pronte per essere utilizzate per altre operazioni (dovremmo quindi scrivere davanti alla funzione non **void** ma il tipo di dato e il **return** finale con il nome della variabile da restituire).