

```

#include <stdio.h>
//Daniele Zizzi Progetto S2 L5

void menu (); //stampa il menu
void moltiplica (); //funzione per effettuare la moltiplicazione
void dividi (); //funzione per effettuare la divisione
void ins_string(); //funzione che permette di inserire una stringa

int main ()
{
    char scelta = {'\0'}; //inizializzo una variabile di tipo char come vuota o null
    do
    {
        menu (); //richiamo la funzione per stampare il menu nel do while, in modo che mi stampi il menu ogni volta
        //scanf ("%d", &scelta);
        scanf (" %c", &scelta); //inserisco la scelta, %d è per gli interi, quindi v'è usato %c
        switch (scelta) //in base alla lettera scelta avvia il codice presente nei case, come il funzionamento di un interruttore vero e proprio
        {
            case 'A':
                moltiplica(); //richiamo la funziona moltiplica
                break;
            case 'B':
                dividi(); //richiamo la funzione dividi
                break;
            case 'C':
                ins_string(); //richiamo la funzione inserisci stringa
                break;
            case 'Q':
                printf("Arrivederci\n"); //stampa arrivederci, nel caso si voglia uscire dal programma
                break;
            default:
                printf("Scelta non presente\nInserirla nuovamente\n"); //se la scelta non è presente nei case, finisce nel default, che gestisce tutti gli altri casi
                break; //permette di uscire dal case, una volta eseguito il codice
        }
        while(scelta != 'Q'); //ripeto il codice presente nel do, finchè non inserisco una scelta valida
        //al posto del case, e' anche possibile gestire la situazione dei casi, concatenando if else
    }
    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\nQ >> Per uscire dal programma\n");
}

```

```
void moltiplica ()
{
    short int  a,b = 0;//inizializza due variabili di tipo short int è da 16 bit (2 byte). Se è signed, quindi con segno i valori vanno da -32768 a 32767, unsigned(senza segno), vanno da da 0 a 65535
    short int prodotto = 0;//buona norma dichiarare tutte le variabili all'inizio ed inizializzarle a 0
    //è consigliabile dichiarare le variabili in tipo int, se non si vuole incorrere in errori quando si moltiplicano 2 numeri grandi. Oppure float se si vuole fare la moltiplicazione con la virgola
    printf ("Inserisci i due numeri da moltiplicare\n");
    printf ("Primo:");
    //scanf ("%f", &a);si utilizza %hd per gli interi di tipo short e non %f
    //scanf ("%d", &b);si utilizza %hd per gli interi di tipo short e non %d
    scanf (" %hd", &a);//per le variabili di tipo short, si aggiunge %h prima del tipo. Pertanto per varabili di tipo short int la sintassi sarà %hd. h sta per half
    printf ("Secondo:");
    scanf (" %hd", &b);

    prodotto = a * b;//assegna a prodotto, il risultato della moltiplicazione tra i valori di a e b

    printf ("Il prodotto tra %d e %d e': %d\n", a,b,prodotto);//stampa il valore di a, b e prodotto
    /*se invece, si vuole moltiplicare un numero n di valori tra loro, è possibile inizializzare un array con n valori, dove n cambierà in base alla scelta dell'utente
    andando a moltiplicare tutti i valori dell'array, tra di loro attraverso un ciclo for finchè i non raggiunge il valore di n*/
}

void dividi ()
{
    int  a,b = 0;//inizializzo due variabili di tipo int, dove andrò a memorizzare i due valori da dividere
    float divisione = 0;
    printf ("Inserisci il numeratore:");
    scanf (" %d", &a);
    //printf ("Inserisci il denominatore:");
    printf ("Inserisci il denominatore:");//non denominatore ma denominatore
    scanf (" %d", &b);

    //int divisione = a % b; inizializzando la variabile divisione in int, non si potranno avere risultati con la virgola. Il simbolo % restituisce il resto
    divisione = (float)a / b; //con il float si possono avere numeri dopo la virgola e il simbolo / ne permette la divisione. (float)permette di assegnare un valore float quando due interi generano un risultato con la virgola

    printf ("La divisione tra %d e %d e': %0.2f\n", a,b,divisione);//%d si usa per un valore di tipo int, %f per un valore di tipo float. Quindi stampo il valore a, b e il risultato della divisione tra i due. %0.2f stampa solo due numeri dopo la virgola
}

void ins_string ()
{
    char stringa[10];
    printf ("Inserisci la stringa:");
    //scanf ("%s", &stringa); con il comando scanf, se immettiamo più di 10 caratteri, ovvero oltre la dimensione dell'array, si verifica lo stack overflow
    fgets (stringa, 10, stdin);//il comando fgets, permette di evitare lo stack overflow, causato dal superamento dell'inserimento dei caratteri nell'array
    printf ("%s", stringa);//stampo l'intero array
}
```