

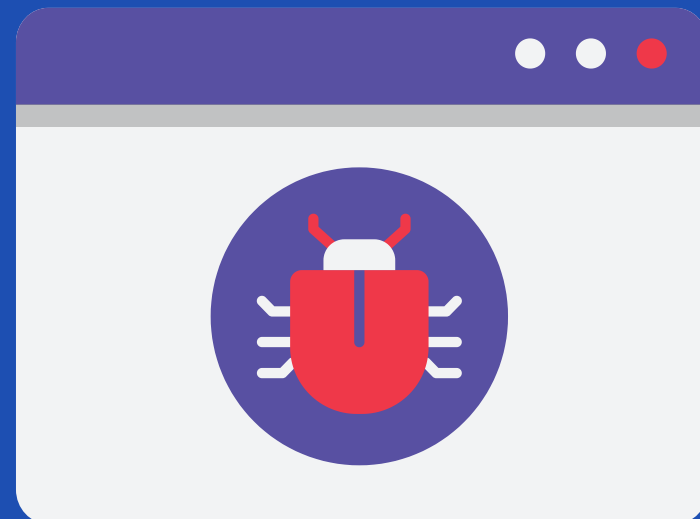


Programmazione C

PROGETTO S2-L5

GIULIA FIACCHI

"BUG HUNTING"





FUNZIONE DEL PROGRAMMA



Dal codice si può evincere che si tratti di un AI che aiuta l'utente ad eseguire alcune operazioni matematiche e nello specifico sono: moltiplicazione, divisione e l'inserimento di una stringa. E chiede all'utente di inserire i numeri per eseguire il calcolo con velocità e facilità.

ANALIZZIAMO INSIEME



Dalla dicitura `#include <stdio.h>` si capisce che si sta usando una libreria che sfrutta input/output ed è fondamentale per utilizzare la funzione "printf".

Le funzioni sotto elencate sono:

- `menu`
- `moltiplica`
- `dividi`
- `ins_string`

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

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

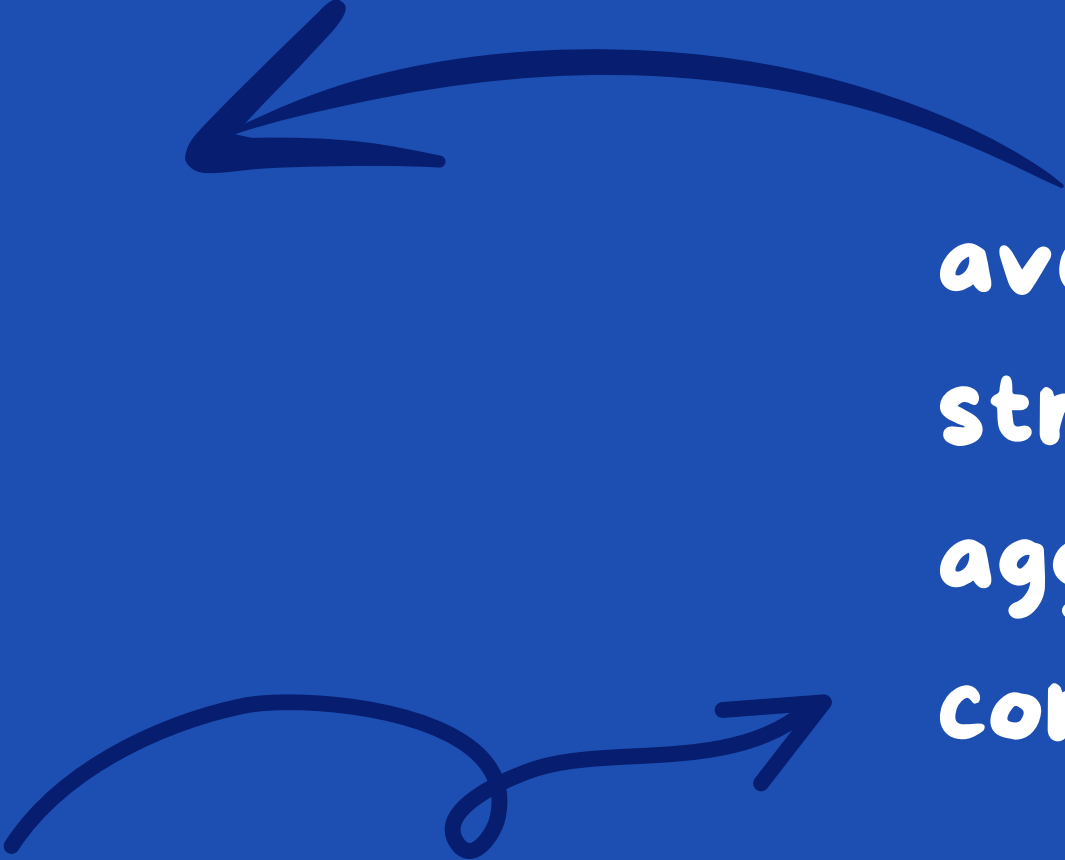
Analizziamo il codice punto per punto e vediamo quali sono i suoi punti critici

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

```
#include <string.h>
```

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

avendo anche la funzione
stringa è necessario
aggiungere la sua libreria
corrispondente



```
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;
}
```

scanf ("%c", &scelta); perchè avendo scelto char dobbiamo richiedere la lettura di un carattere e non di un intero che è ciò che fa %d

devono essere aggiunti sotto ai case ed ivenziati:

case 'a': - case 'b': - case 'c':

questo serve per permettere all'utente di inserire il carattere sia maiuscolo che minuscolo di modo che la mancanza dell'uno o dell'altro non provochi errore

AGGIUNGERE

default:

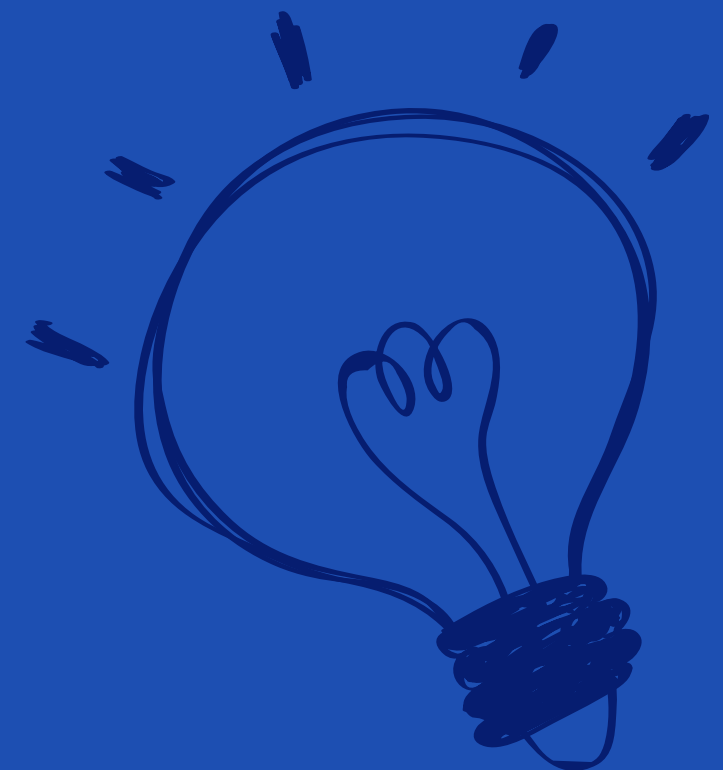
printf(Errore\nInserisci di nuovo\n);

il default è utile per avvisare l'utente che se non opta per i caratteri indicati nei case, il menu andrà in errore e dovrà indicare il carattere corretto

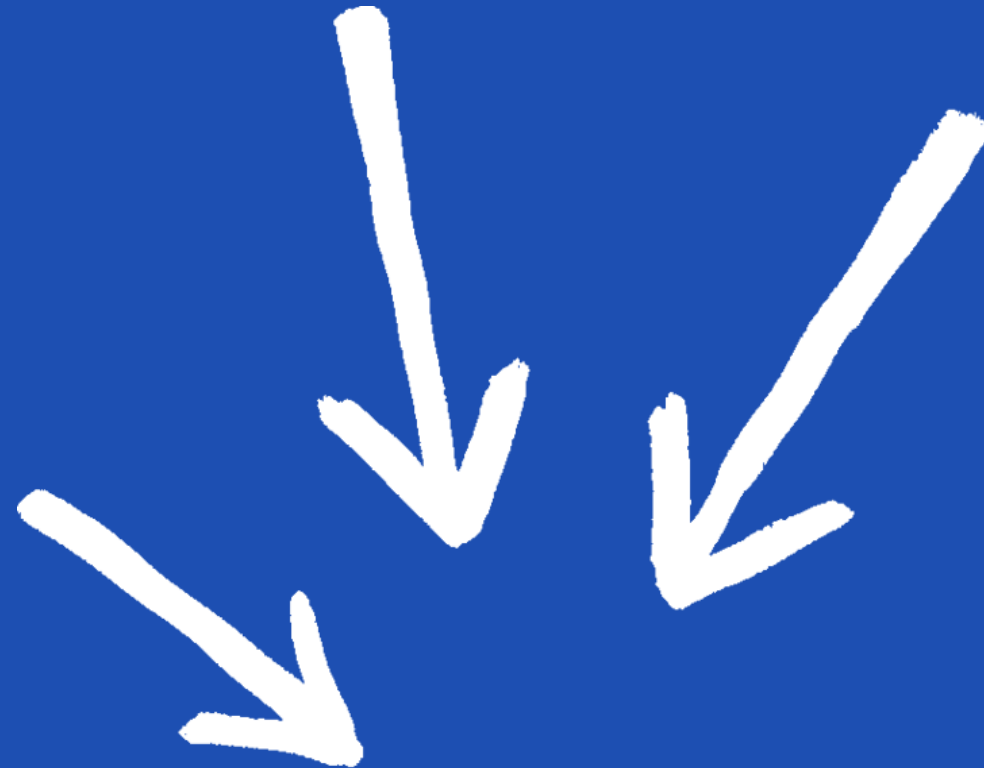
In più è necessario aggiungere un ciclo "do while" per dare la possibilità all'utente di ritornare al menù e poter scegliere un case diverso e continuare le operazioni. E nel while inserire (scelta == 0) di modo che continui ad eseguire il codice.

```
do {
```

```
} while (scelta == 0
```

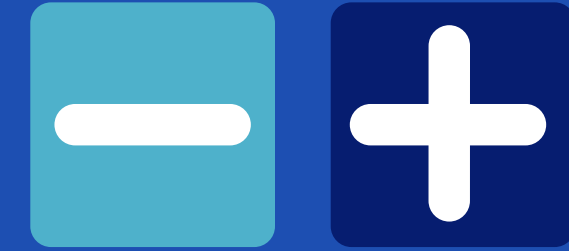



```
void menu ()  
{  
    printf ("Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti\n");  
    printf ("Come posso aiutarti?\n");  
    printf ("A >> Moltiplicare due numeri\nB >> Dividere due numeri\nC >> Inserire una stringa\n");  
}
```



utilizza la funzione void perchè non ha bisogno di un ritorno di valore

MOLTIPLICAZIONE



```
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;

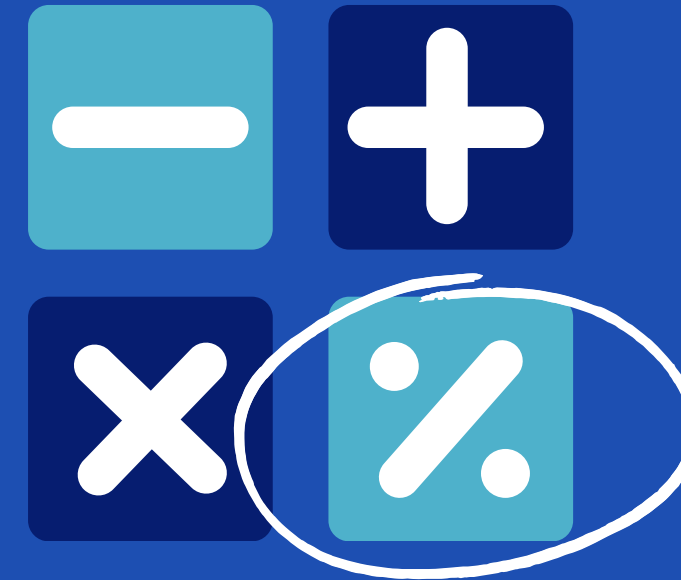
    printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);
}
```

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

perchè avendo scelto int
dobbiamo richiedere la lettura
di un intero e non di un
carattere che è ciò che fa %f

in più è necessario usare su entrambi gli scanf e sul printf %hd perchè si è scelto di
utilizzare la funzione "short int" che quindi ha una taglia diversa, 16 bit e non 32 come int

DIVISIONE



```
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);
}
```

qui % deve essere
sostituito da / perchè si
tratta di una divisione

`int divisione = a / b`

STRINGA

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



Per evitare un overflow è poi necessario definire i caratteri ammessi che, non superino la grandezza dell'array.

Quindi potremmo mettere:

```
char stringa [100];  
scanf ("%99s", stringa)
```



In questo caso la funzione stringa che, si occupa di stampare la stringa inserita dall'utente, non è così utile al fine del funzionamento del codice e per le utilità del programma.