

Leggendo questo codice è abbastanza chiaro che è un programma per fare operazione matematiche.(e per salvare stringe)

In quanto è eseguito, il programma ci da un messaggio di benvenuta, e poi ci chiede che calcolo matematico vogliamo realizzare:

- Moltiplica
- Dividi
- Inserire una stringa

```
kali@kali: ~/Desktop
File Actions Edit View Help
GNU nano 7.2 s2_l5 *
#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");
    printf ("Come posso aiutarti?\n");
    printf ("A >> Moltiplicare due numeri\nB >> Dividere due numeri\nC >> Inserire una stringa\n");
}
```

Seconda parte del  
codice.

```
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 denominator:");
    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);
}
```


 ^G Help  
^X Exit

 ^O Write Out  
^R Read File

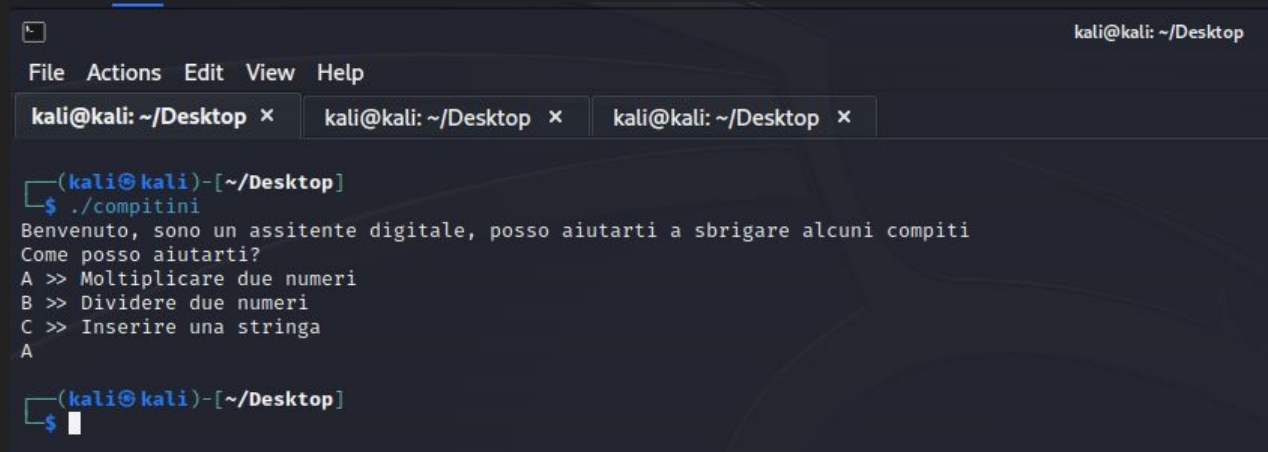
 ^W Where Is  
^N Replace

 ^K Cut  
^U Paste

 ^T Execute  
^J Justify

 ^C Location  
^\_ Go To Line

Qui abbiamo il  
primo problema  
logico. Il  
programma si  
chiude se  
facciamo  
qualsiasi scelta,  
non funziona.



The screenshot shows a Kali Linux terminal window with a menu-driven program. The window title is 'kali@kali: ~/Desktop'. The menu options are: A >> Moltiplicare due numeri, B >> Dividere due numeri, C >> Inserire una stringa. The user has selected option A.

```
(kali@kali)-[~/Desktop]
$ ./compitini
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti
Come posso aiutarti?
A >> Moltiplicare due numeri
B >> Dividere due numeri
C >> Inserire una stringa
A
(kali@kali)-[~/Desktop]
$
```

La prima modifica che ho fatto al codice ha stato spostare le definizioni delle funzione `void` all'inizio del file, prima di iniziare la funzione `main`.

```
kali@kali: ~/Desktop
File Actions Edit View Help
kali@kali: ~/Desktop x kali@kali: ~/Desktop x kali@kali: ~/Desktop x
GNU nano 7.2 s2_l5.c
#include <stdio.h>

//voids

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);
}

int main ()
```

⌘G Help ⌘O Write Out ⌘W Where Is ⌘K Cut ⌘T Execute ⌘C Location ⌘U Undo  
⌘X Exit ⌘R Read File ⌘N Replace ⌘V Paste ⌘J Justify ⌘/\_ Go To Line ⌘-E Redo

## PRIMA

Primo errore sistemato: il scanf cercava un tipo di argomento `%d` (variabile di intero)

L'ho cambiato per un tipo di argomento `%c` (variabile di lettera)

Ho anche aggiunto la possibilità di inserire una lettera minuscola come input, e un messaggio in caso si inserisca una risposta non valida.

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

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

    return 0;
}
```

## DOPO

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

    switch (scelta)
    {
        case 'A':
        case 'a':
            multiplica();
            break;
        case 'B':
        case 'b':
            dividi();
            break;
        case 'C':
        case 'c':
            ins_string();
            break;
        default:
            printf ("Per favore solo inserire risposte valide\n");
    }

    return 0;
}
```

Un secondo **problema**. La funzione di moltiplicare non funziona correttamente, ci prende il primo valore come 0, e non ci lascia fare una distinzione intuitivamente fra il primo numero e il secondo

In questo codice vediamo che si sta usando il tipo di variabile **short int**, quindi non ci permette mettere molti numeri come input.

```
(kali@kali)-[~/Desktop]
$ ./compitini
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti
Come posso aiutarti?
A >> Moltiplicare due numeri
B >> Dividere due numeri
C >> Inserire una stringa
a
Inserisci i due numeri da moltiplicare:2 5
Il prodotto tra 0 e 5 e': 0
```

```
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);
}
```

Ecco, problema risolto e funzione ottimizzata. Ora ci permette di inserire ogni numero in ordine, facendolo funzionare.

```
void moltiplica ()
{
    float c,d = 0;
    printf ("Inserisci il primo numero da moltiplicare:");
    scanf ("%f", &c);
    printf ("Inserisci il secondo numero da moltiplicare:");
    scanf ("%f", &d);

    float prodotto = c * d;

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

Ho usato anche il tipo di variabile `%f` che ci permetterà inserire numeri decimali o negativi, e nella risposta il tipo di funzione `%g` che ci dirà i numeri dalla maniera più corta possibile, così non ci dice i 0 decimali innescessari.

(kali@kali)-[~/Desktop]

\$ ./compitini

Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti  
Come posso aiutarti?

A >> Moltiplicare due numeri

B >> Dividere due numeri

C >> Inserire una stringa

a

Inserisci il primo numero da moltiplicare:2.6

Inserisci il secondo numero da moltiplicare:1.2

Il prodotto tra 2.6 e 1.2 e': 3.12

(kali@kali)-[~/Desktop]

\$

Ora al void di divisione. Si può migliorare.

Ho aggiunto una risposta in caso si prove a dividere per 0, e ho anche aggiunto la possibilità di usare numeri decimali e negativi.

```
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 dividi ()
{
    float c, d;
    printf("Inserisci il numeratore: ");
    scanf("%f", &c);
    printf("Inserisci il denominatore: ");
    scanf("%f", &d);

    if (d != 0)
    {
        float divisione = c / d;
        printf("La divisione tra %g e %g e': %g\n", c, d, divisione);
    }
    else
    {
        printf("Impossibile dividere per zero\n");
    }
}
```



Ora al void di stringa. Questa funzione del nostro programma in realtà non fa molto per un utente, quindi ho aggiunto anche una funzione che ci fa una sorta di echo, dove il programma ci ripete quello che abbiamo scritto, sempre sia di 10 carattere o meno.

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

```
void ins_string()  
{  
    char stringa[11];  
    printf("Inserisci la stringa: ");  
    scanf("%10s", stringa);  
    printf("La stringa inserita e': %s\n", stringa);  
}
```

Voilà, tutto  
funziona!

```
print(":~")
```

```
(kali@kali)~[~/Desktop]
$ gcc -g s2_l5.c -o compitini

(kali@kali)~[~/Desktop]
$ ./compitini
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti
Come posso aiutarti?
A >> Moltiplicare due numeri
B >> Dividere due numeri
C >> Inserire una stringa
a
Inserisci il primo numero da moltiplicare:5
Inserisci il secondo numero da moltiplicare:2.7
Il prodotto tra 5 e 2.7 e': 13.5

(kali@kali)~[~/Desktop]
$ ./compitini
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti
Come posso aiutarti?
A >> Moltiplicare due numeri
B >> Dividere due numeri
C >> Inserire una stringa
b
Inserisci il numeratore: -4
Inserisci il denominatore: 2
La divisione tra -4 e 2 e': -2

(kali@kali)~[~/Desktop]
$ ./compitini
Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti
Come posso aiutarti?
A >> Moltiplicare due numeri
B >> Dividere due numeri
C >> Inserire una stringa
C
Inserisci la stringa: Ciao00
La stringa inserita e': Ciao00

(kali@kali)~[~/Desktop]
$
```