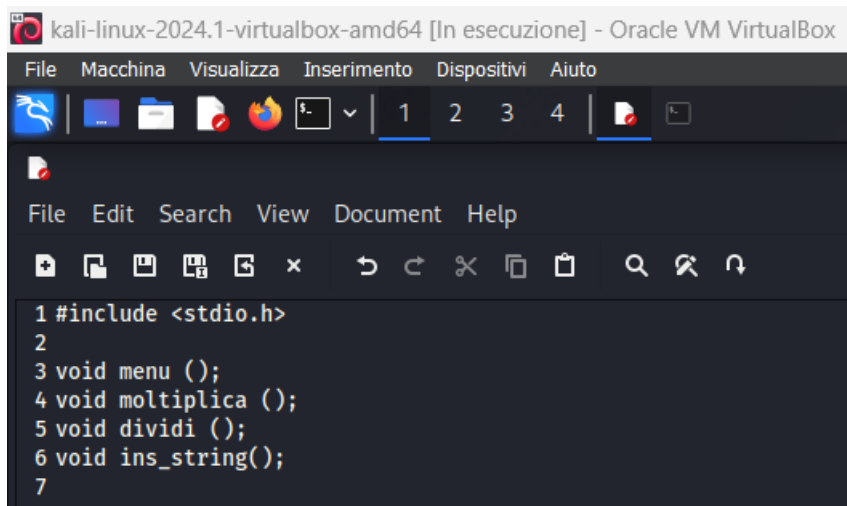


## REPORT CORREZIONE CODICE S2L5

Descrizione del codice : un “assistente” capace di eseguire 3 tipi di operazioni

- 1) Moltiplicazione
- 2) Divisione
- 3) Scrivere una stringa

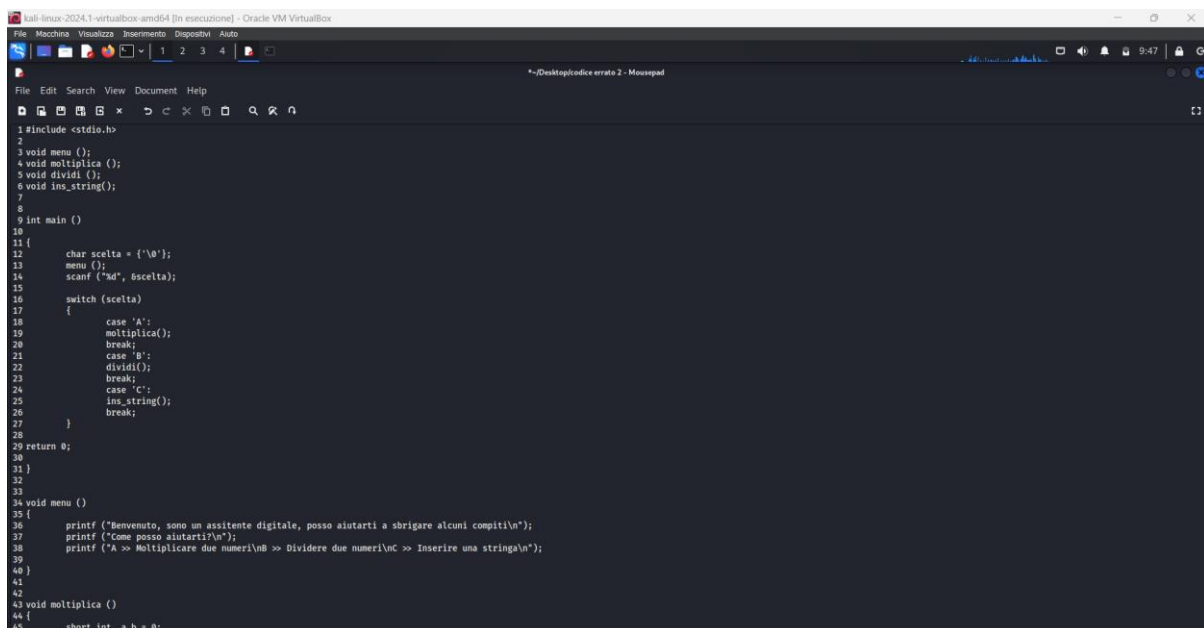
L'utente può inserire i fattori di cui necessita e il programma eseguirà queste 3 operazioni



The screenshot shows a code editor window titled "kali-linux-2024.1-virtualbox-amd64 [In esecuzione] - Oracle VM VirtualBox". The editor has a menu bar with "File", "Macchina", "Visualizza", "Inserimento", "Dispositivi", and "Aiuto". Below the menu bar is a toolbar with icons for file operations and editing. The code in the editor is as follows:

```
1 #include <stdio.h>
2
3 void menu ();
4 void moltiplica ();
5 void dividi ();
6 void ins_string();
7
8
```

ERRORE 1) mancanza della libreria “#include <string.h>” per far funzionare la variabile “stringa” nella riga 2



The screenshot shows the same code editor window with the completed code. The code is as follows:

```
1 #include <stdio.h>
2
3 void menu ();
4 void moltiplica ();
5 void dividi ();
6 void ins_string();
7
8
9 int main ()
10 {
11     char scelta = {'\0'};
12     menu ();
13     scanf ("%d", &scelta);
14     switch (scelta)
15     {
16         case 'A':
17             moltiplica();
18             break;
19         case 'B':
20             dividi();
21             break;
22         case 'C':
23             ins_string();
24             break;
25     }
26 }
27
28 return 0;
29 }
30
31 void menu ()
32 {
33     printf ("Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti\n");
34     printf ("Come posso aiutarti?\n");
35     printf ("A >> Moltiplicare due numeri\nB >> Dividere due numeri\nC >> Inserire una stringa\n");
36 }
37
38 void moltiplica ()
39 {
40     short int a b a b;
41 }
```

ERRORE 2) mancanza di una condizione che impedisca un input errato, per evitare immissioni di codici sorgente malevoli

ERRORE 3) tipo di variabile errato all'interno dello scanf alla riga 14 (%d invece di %c)

ERRORE 4 ) è preferibile il tipo di variabile INT invece di SHORT INT inquanto, SHORT INT possiede meno BYTE (2) e quindi essere facilmente più attaccabile causando un buffer overflow specialmente in una operazione come la moltiplicazione che accresce il numero dei moltiplicatori(errone a riga 45)

ERRORE 5) tipo di variabile errato all'interno dello scanf alla riga 47 (%f invece di %d)

ERRORE 6) è preferibile il tipo di variabile FLOAT invece di INT inquanto, FLOAT possiede più BYTE (4) e comprende anche numeri reali, che spesso sono risultanti dalle operazioni di divisione, con apposita correzione alle tipologie di variabili dei printf/scanf ( da "%d" a "%f") (righe 58-66)

ERRORE 7) probabile errore alla riga 61, c'è scritto "denumeratore" invece di "denominatore"

ERRORE 8) errore alla riga 64, operazione di resto intero (a%b) invece della normale operazione di divisione (a/b)

ERRORE 9) alla riga 75 il numero di celle riservato all' Array è troppo basso con il rischio di un buffer overflow

ERRORE 10) dalla riga 75 alla riga 77 il numero di celle non è provvisto di eventuale codice di controllo variabili che impedisca un eccessivo input di caratteri, risultanti in problemi di sicurezza

```
kali-linux-2024.1-virtualbox-amd64 [in esecuzione] - Oracle VM VirtualBox

File Macchina Visualizza Inserimento Dispositivi Aiuto
Shell No. 1

void menu ()
{
    printf ("Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti!\n");
    printf ("Come posso aiutarti?\n");
    printf ("a -> Multiplicare due numeri\nb -> Dividere due numeri\nc -> Inserire una stringa\n");
}

void multiplica ()
{
    int a,b = 0;
    printf ("Inserisci i due numeri da moltiplicare:\n");
    scanf ("%d", &a);
    scanf ("%d", &b);

    int prodotto = a * b;

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

void dividi ()
{
    float a,b = 0;
    printf ("Inserisci il numeratore:\n");
    scanf ("%f", &a);
    printf ("Inserisci il denominatore:\n");
    scanf ("%f", &b);

    float divisione = a / b;

    printf ("La divisione tra %f e %f e' %f", a,b,divisione);
}

void ins_string ()
{
    char stringa[20];
    printf ("Inserisci la stringa:\n");
    scanf ("%s", &stringa);
    if (strlen(stringa) > 20)
    {
        printf ("I caratteri superano le 20 unita'\n");
    }
    printf ("\nla stringa che hai scritto e' %s", stringa);
}

#include <stdio.h>
#include <string.h>

void menu ();
void multiplica ();
void dividi ();
void ins_string();

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

    if(scelta == 'a' || scelta == 'b' || scelta == 'c') {
        switch (scelta)
        {
            case 'a':
                multiplica();
                break;
            case 'b':
                dividi();
                break;
            case 'c':
                ins_string();
                break;
        }
    } else {
        printf("Il carattere inserito non e' valido");
    }
    return 0;
}

void menu ()
{
    printf ("Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti!\n");
    printf ("Come posso aiutarti?\n");
    printf ("a -> Multiplicare due numeri\nb -> Dividere due numeri\nc -> Inserire una stringa\n");
}

void multiplica ()
{
    int a,b = 0;
}
```

## RISULTATO :

Una volta eseguite tutte le correzioni otteniamo un codice più sicuro e privo di errori capace di eseguire le operazioni citate nella descrizione del programma

