

ANALISI DEL PROGRAMMA ESERCIZIO 5 EPICODE W2

Una volta ricevuto il file .txt con le stringhe del programma sono andato ad aprirlo con il blocco note. Dopo una attenta lettura ho visto che si tratta di un programma che si vuole utilizzare come assistente. Per essere più precisi il programma si occuperà di fare moltiplicazioni tra 2 numeri, divisioni sempre tra 2 numeri ed infine inserirà stringhe.

Inizialmente viene inclusa la libreria stdio.h che si occuperà della gestione degli input e dell'output, indicata nell'immagine qui sotto dalla freccia rossa.

Esercizio_5_Epicode_w2 - Blocco note di Windows

File Modifica Formato Visualizza ?

#include <stdio.h>

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

Il cerchio verde invece indica le funzioni void, funzioni che non restituiscono nessun risultato. Nel nostro esercizio abbiamo possiamo vedere 4 funzioni void.

Successivamente abbiamo la funzione int main (). La prima cosa che viene dichiarata è la variabile char che prende il nome di scelta ed ha valore 0. Dopo la variabile stringa viene richiamata la funzione void menu () che vedremo più avanti. A questo punto l'utente dovrà inserire dei valori in input che verranno associati alla variabile scelta. A questo punto si attiva la funzione switch, divisa nei tre casi possibili A (legata alla funzione void riguardante la moltiplicazione), B (relativa alla divisione) e C (la possibilità di inserire stringhe). Successivamente la risoluzione di uno di questi 3 casi il programma verrà chiuso, come ci suggerisce la presenza del return 0, in quanto ha completato il suo lavoro,

```
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;
}
```

A seguire la funzione principale abbiamo le sottofunzioni voi a cui abbiamo accennato sopra:

```
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);
}
```

La funzione `voi menu ()` si occupa di andare a mostrare in output all'utente a cosa serve il programma e a dargli la scelta dell'operazione che vuole eseguire, come mostrato dalla freccia gialla e da quella blu. Sotto abbiamo invece la funzione `void` che si occuperà della gestione dell'attività di moltiplicazione. Vengono dichiarate le 2 variabili intere `a` e `b` che dovranno essere successivamente inserite dall'utente (`scanf()`) e che il loro prodotto è la variabile intera "prodotto" che verrà mandata in output tramite il comando "`printf ("Il prodotto tra %d e %d è : %d", a, b, prodotto);`" i "`%d`" rappresentano la tipologia di variabile che verrà presa in considerazione mentre `a`, `b` e `prodotto` servono ad indicare esattamente i valori delle variabili che dovranno andare in output.

Ora passiamo alle rimanenti 2 funzioni `voi` che possiamo vedere nell'immagine sottostante:

```
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);
}
```

Abbiamo la funzione `void dividi ()` che si occuperà della divisione una volta che l'utente avrà inserito i valori. La funzione `void ins_string ()` si occupa invece di prendere in input una stringa di lunghezza 10 da parte dell'utilizzatore del programma.

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

Questo programma non è scritto in maniera perfetta. Infatti per come è scritto non contempla alcune possibilità per quanto riguarda gli input che potrebbero essere sfruttate in maniera negativa. Possono essere trovate nelle immagini con la sottolineatura in arancione.

Cominciando dalla seconda immagine possiamo vedere come il programmatore non abbia preso in considerazione la possibilità che l'utilizzatore nel momento della scelta all'interno dello switch possa scegliere un'opzione non contemplata (che non sia né A, né B, né C). Bisognerebbe andare ad inserire un comando default all'interno dello switch che avvisi l'utente della scelta sbagliata. Discorso simile può essere fatto all'interno della funzione `void moltiplica ()`. In fase di input l'utilizzatore potrebbe andare a inserire dei valori non numerici. A ciò possiamo aggiungere che per sicurezza le variabili potrebbero essere cambiate da `short int` a `float`, dando così la possibilità di inserire e ottenere numeri decimali. Per quanto riguarda la divisione abbiamo invece 3 problematiche: la prima è condivisa con la moltiplicazione, cioè la possibilità che vengano inseriti valori non numerici. La seconda invece, sua esclusiva, è che vengano fatte divisioni con divisore 0. La terza è che avendo solo parametri interi non vengono contemplati i numeri con la virgola e quindi anche qua sarebbe meglio andare a cambiare le variabili in `float` invece che `int`. Infine per quanto riguarda la funzione che si occupa dell'inserzione di stringhe, la problematica principale è che possano essere inseriti un numero di caratteri che superino il valore indicato all'interno delle parentesi quadre.

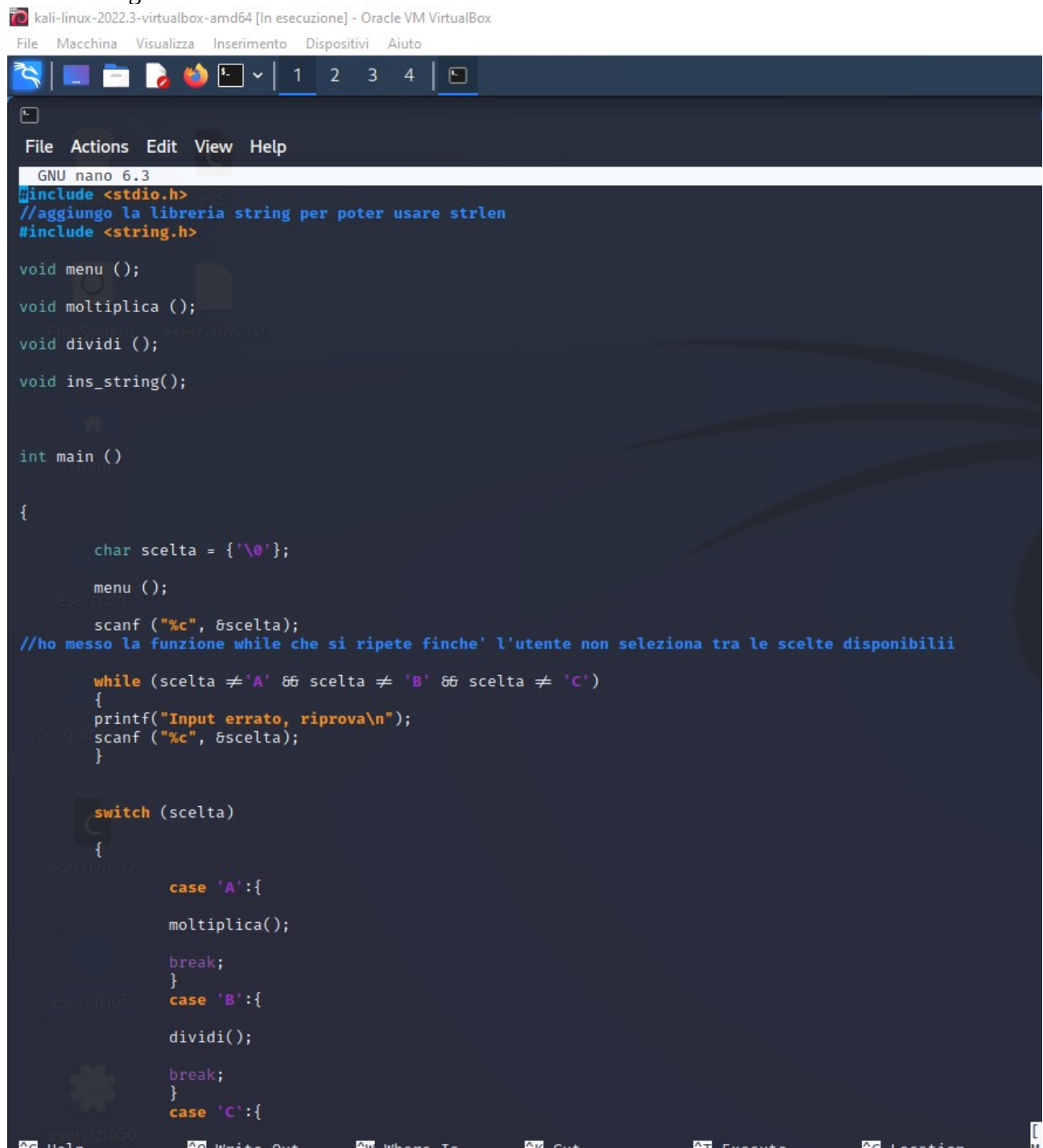
Oltre a queste problematiche possiamo vedere come siano stati commessi anche degli errori di sintassi (cerchiati in rosso nelle immagini):

-Nella funzione `int main ()` nello `scanf` vi è scritto `%d` riferito alla variabile scelta. Ma la variabile scelta non è una variabile numerica quindi il `%d` va sostituito con un `%c`

-All'interno della funzione `moltiplica` abbiamo un errore simile: nello `scanf` per inserire il primo moltiplicatore ci si rifà alla tipologia di variabile `%f` per indicare la variabile a che però è un intero. Se vogliamo rimanere nel gruppo dei numeri interi basta semplicemente cambiare il `%f` in `%d`

-Nella funzione che si occupa delle divisioni abbiamo un'espressione scritta male. Infatti `a % b` non vuol dire fare la divisione tra i valori `a` e `b`, bensì calcolarne il modulo, cioè il resto della loro divisione.

Di seguito invece ci sono le immagini del codice che ho modificato con qualche commento per indicare meglio i cambiamenti:



```
GNU nano 6.3
#include <stdio.h>
//aggiungo la libreria string per poter usare strlen
#include <string.h>

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

int main ()
{
    char scelta = {'\0'};

    menu ();

    scanf ("%c", &scelta);
    //ho messo la funzione while che si ripete finche' l'utente non seleziona tra le scelte disponibili
    while (scelta != 'A' && scelta != 'B' && scelta != 'C')
    {
        printf("Input errato, riprova\n");
        scanf ("%c", &scelta);
    }

    switch (scelta)
    {
        case 'A':{
            moltiplica();

            break;
        }
        case 'B':{
            dividi();

            break;
        }
        case 'C':{
```

File Actions Edit View Help

GNU nano 6.3

```
}
    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");
}

void moltiplica ()
{
    //ho messo come variabile float per fare le moltiplicazioni anche con i decimali
    float a,b;
    printf("Inserisci i due numeri da moltiplicare;\n");
    scanf("%f", &a);
    scanf("%f", &b);
    float prodotto = a * b;
    printf ("Il prodotto tra %f e %f e' : %f", a,b,prodotto);

    //non sono riuscito a trovare una soluzione per non fare inserire valori non numerici
}
```

```
void dividi ()
{
    int a,b = 0;
    printf ("Inserisci il numeratore:");
    scanf ("%d", &a);
    printf ("Inserisci il denominatore:");
    scanf ("%d", &b);
    //ho messo questo if per evitare che l'utente divida per 0
    if ( b!=0){
        int divisione = a / b;
        printf ("La divisione tra %d e %d e': %d", a,b,divisione);
    }
    else{
        printf("Impossibile eseguire questa divisione\n");
    }

    // sono riuscito a trovare una soluzione per escludere tutti i valori non numerici ma onestamente non so come
}
```

```
void ins_string ()
{
    //ho messo il ciclo do while per controllare che l'utente non metta più dei 10 caratteri di lunghezza
    //per fare cio' ho aggiunto le 2 variabili l e rapporto
    char stringa[10];
    float l;
    float rapporto;
    do{
        printf ("Inserisci la stringa:");
        scanf ("%s", &stringa);
        l = strlen(stringa); //l rappresenta la lunghezza in caratteri della stringa chiamata stringa
        rapporto = l / 10; //rapporto equivale a l diviso 10
    }
    while (rapporto > 1); //in questo modo se la lunghezza della stringa stringa supera il valore 1 vuol dire che è più lunga di 10 e viene chiesto all'utente di inserirla nuovamente
    rewind(stdin);
    printf("La tua stringa e' %s\n", stringa);
}
```

Invece nelle prossime immagini si può vedere come gira il codice con i cambiamenti che ho apportato:

```
(kali@kali)-[~/Desktop]
$ gcc esercizio5.c -o esercizio5

(kali@kali)-[~/Desktop]
$ ./esercizio5
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
q
Input errato, riprova
Input errato, riprova
█
```

Qui si vede come risponde il programma se si prova ad inserire una lettera diversa sa A, B o C.

Qui cosa accade se si prova a dividere per 0

```
(kali@kali)-[~/Desktop]
$ ./esercizio5
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:1
Inserisci il denominatore:0
Impossibile eseguire questa divisione
```

```
(kali@kali)-[~/Desktop]
$ ./esercizio5
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:q
Inserisci il denominatore:Impossibile eseguire questa divisione

(kali@kali)-[~/Desktop]
$ ./esercizio5
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:1
Inserisci il denominatore:q
Impossibile eseguire questa divisione
```

Possiamo vedere cosa accade se si inserisce una lettera prima al numeratore e poi al denominatore

Cosa succede quando si fa un input inferiore ai 10 caratteri

```
(kali@kali)-[~/Desktop]
$ ./esercizio5
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:a
La tua stringa e' a
```

```
(kali@kali)-[~/Desktop]
$ ./esercizio5
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:aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Inserisci la stringa:█
```

Cosa succede quando l'input è superiore ai 10 caratteri

