Relazione
Progetto Epicode S2-L5
Studente: Simone Mininni

Codice esercizio:

```
#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 assitente 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 denumeratore:");
    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);
}
```

Traccia

EPICODE

Esercizio
Traccia e requisiti

Traccia:

Per agire come un Hacker bisogna capire come pensare fuori dagli schemi. L'esercizio di oggi ha lo scopo di allenare l'osservazione critica.

Dato il codice in allegato, si richiede allo studente di:

- Capire cosa fa il programma senza eseguirlo
- Individuare dal codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati)
- Individuare eventuali errori di sintassi / logici
- Proporre una soluzione per ognuno di essi

Esercizio_10_Epicode.c

1. Il programma consta di una funzione main da dove parte il programma, qui l' utente può decidere se chiamare una delle seguenti funzioni:
moltiplica, dividi, inserisci stringa.
Le funzioni sono di tipo void quindi, una volta che terminano le task, non restituiranno alcun valore al main. Successivamente il programma termina.

2. Situazioni potenziali che il programma non gestisce:
   - Inserimento di un carattere alfanumerico diverso da quello proposto nella funzione menu().
   - Nella funzione moltiplica() le variabili sono di tipo short int non considerando che l' utente potrebbe inserire numeri più grandi di 2 byte o che il risultato della moltiplicazione possa essere più grande di 2 byte.
   - Nella funzione dividi() il risultato divisione è di tipo int non considerando che ci potrebbe essere resto diverso da 0 e quindi sarebbe necessario rappresentarlo con un numero reale, quindi di tipo float o double.
   - Nella funzione ins_string() non si gestisce un eventuale errore di stack overflow, l'utente potrebbe inserire un numero di caratteri superiore al limite della array stringa.
   - Quando la funzione chiamata restituisce il controllo alla funzione chiamante il programma termina non considerando la possibilità che l' utente voglia eseguire un'altra operazione senza rieseguire il programma.

3. Errori logici trovati:

```
- int main ()
-
- {
-         char scelta = {'\0'};
-         menu ();
-         //scanf ("%d", &scelta); //Errore logico, come vuole la funzione menu(),
-         scanf deve prendere un carattere di tipo char.
-         //Soluzione
-         scanf ("%c", &scelta);
- ——————————————————————————————————————————————————————————————————————————————
- void moltiplica ()
- {
-         short int  a,b = 0;
-         printf ("Inserisci i due numeri da moltiplicare:");
-         //scanf ("%f", &a);//Errore logico, scanf deve prendere un numero intero e non
-         un float
-         //Soluzione
-         scanf ("%d", &a);
- ——————————————————————————————————————————————————————————————————————————————
-
- void dividi ()
```

```
-   {
-       ——------
-
-       //int divisione = a % b; //Errore logico, % è operatore resto e non divisione.
-       //Soluzione
-       int divisione = a / b;
-       ——--------
-
-   }
```

4. Codice con soluzione a tutte le problematiche studiate precedentemente:

```
#include <stdio.h>

//Funzioni, dichiarazioni
void menu ();
void moltiplica ();
void dividi ();
void ins_string();
void clearBuffer();


//Main
int main ()

{
  char scelta = {'\0'};

 //Il programma termina solo quando l' utente inserisce il carattere opportuno!
 //La logica del do while permette anche di gestire eventuali errori di inserimento da
 parte dell' utente
 do{

            menu ();
            scanf ("%c", &scelta);
            clearBuffer();

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

```
                case 'B':
                dividi();
                break;
                case 'C':
                ins_string();
                break;
                default:
                if(scelta!='F'){
                  printf("\nErrore nell'inserimento!");
                }
                break;
                      }


  }while(scelta != 'F');

  printf("Programma terminato, arrivederci!\n");

  return 0;

}

//Funzioni
void menu ()
{
  printf("\n\n");
          printf ("Benvenuto, sono un assitente 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");
  printf("F >> Terminare il programma\n");

}


void moltiplica ()
{
  int  a,b = 0;// definiamo tipo int per avere un range maggiore
  printf ("Inserisci i due numeri da moltiplicare:");
  scanf ("%d", &a);
  scanf ("%d", &b);
  clearBuffer();// Pulire il buffer di tastiera per scanf
```

```
 int prodotto = a * b;
 printf ("Il prodotto tra %d e %d e': %d\n", a,b,prodotto);
}


void dividi ()
{
  int  a,b = 0;
  printf ("Inserisci il numeratore:");
  scanf ("%d", &a);
  printf ("Inserisci il denominatore:");
  scanf ("%d", &b);
  clearBuffer();

  // passiamo a float con un casting esplicito per considerare soluzioni reali
  float divisione = (float)a/b;

  printf ("La divisione tra %d e %d e': %.2f\n", a,b,divisione);
}


void ins_string ()
{
  char stringa[10];
  printf ("Inserisci la stringa: ");
  scanf ("%9s", &stringa);
  /** gestiamo l' errore di stack overflow controllando il numero di caratteri
  assegnabili alla array stringa e poi chiamiamo la funzione clearBuffer() per
  eliminare i caratteri in eccesso salvati temporaneamente nel buffer**/
  clearBuffer();

  printf("La tua frase è: %s\n", stringa);
}

//Funzione per ripulire il buffer.
void clearBuffer(){
  while(getchar() != '\n');
}
```