Progetto Settimana 2 Lezione 5

***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

Prima parte

```
#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");
}
```

Seconda parte

```
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);}
```

Cosa fa il programma

Il programma permette all'utilizzatore di fare una scelta tra 3 possibili compiti che è in grado di svolgere:

- **moltiplicare due numeri;**
- **dividere due numeri;**
- **inserire una stringa.**

Individuare casistiche non standard che il programma non gestisce

Nella prima parte, si nota che il programma non è in grado di gestire un input da parte dell'utilizzatore diverso dalle tre casistiche considerate nello switch, che non contempla il caso 'default'.

Nella seconda parte, all'interno della funzione moltiplica, non è presente un tipo di controllo sul tipo di dato inserito, cosicché l'utente potrebbe inserire indifferentemente lettere o numeri senza che il programma valuti l'input e possa quindi lavorare con le variabili presenti nella funzione. Inoltre, si è scelto di utilizzare un tipo di input di tipo short int, che obbliga chi utilizza il programma a moltiplicare soltanto numeri interi e di dimensioni limitate.
La funzione dividi, analogamente, non effettua alcun tipo di controllo sui dati inseriti. Anche qui si è scelto di utilizzare variabili di tipo intero, limitando non solo le operazioni possibili all'insieme degli interi, ma vincolando allo stesso insieme anche il risultato della loro divisione. Non gestisce inoltre il caso in cui il denominatore sia uguale a 0, e infine, al posto della funzione divisione è stata utilizzata la funzione che restituisce il resto intero della divisione.
La funzione ins_string, come le altre, non è in grado di gestire eventuali errori derivanti da un errato o volontario inserimento di valori diversi dalle stringhe. Non vi è, inoltre, un controllo sulla lunghezza del dato inserito, e questo potrebbe portare a problemi di buffer overflow.

Individuare eventuali errori di sintassi/logici

In *int main*(), al momento dell'assegnazione della risposta alla variabile di tipo *char scelta*, si è utilizzato *%d*, valido per l'assegnazione di numeri interi, al posto del %s, utilizzato per le stringhe.
La funzione di *switch* non prevede un caso di default ed è di tipo case-sensitive.
Nella funzione *moltiplica*() negli *scanf* sono stati inseriti un *%f* ed un *%d* al posto di *%hd*, utilizzati nel caso di dato *short int*.
Nella funzione *dividi*(), come già visto in precedenza, non è esaminato il caso denominatore=0 e, al posto del risultato della divisione, il programma restituisce il resto.

Proposta di Soluzione

```
#include <stdio.h>
#include <stdlib.h> // per il comando exit()

void menu();
void moltiplica();
void dividi();
void ins_string();
short int valida_short_int(); // verifica che il tipo di dato inserito sia di tipo short int
int valida_intero(); // verifica che il tipo di dato inserito sia di tipo int

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

  switch (scelta){
  case 'A':
  case 'a':
    moltiplica();
    break;
  case 'B':
  case 'b':
    dividi();
    break;
  case 'C':
  case 'c':
    ins_string();
    break;
  default:                          // di default, per ogni valore diverso dalle lettere a, A, b, B, c, C, arresta il programma
    printf("Scelta non valida.\n");
    exit(EXIT_FAILURE);}
  return 0;}
```

In questa e nella successiva pagina è riportata una possibile soluzione alla correzione degli errori di sintassi e alla gestione del dato in input. Sono state lasciate invariate le scelte riguardanti "la grandezza" dei numeri che il programma può moltiplicare o dividere, come anche la scelta di effettuare moltiplicazione e divisione soltanto fra numeri interi, in quanto si è ipotizzato che, essendo il programma impostato con l'utilizzo di quei soli insiemi, fosse necessità/convenienza dell'utilizzatore mantenerli tali. Eventualmente, per la modifica, sarebbe sufficiente variare il tipo di dato da short int/int a float, e modificare gli scanf e printf con il %f anziché %hd/%d, questo si nelle funzioni di calcolo che di verifica.

```
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;
  printf("Inserisci i due numeri da moltiplicare:");
  a=valida_short_int();  // valida il valore prima di passarlo alla variabile a
  b=valida_short_int(); // valida il valore prima di passarlo alla variabile b
  short int prodotto = a * b;
  printf("Il prodotto tra %d e %d e': %d\n", a, b, prodotto);}

void dividi(){
  int a, b;
  printf("Inserisci il numeratore:");
  a=valida_intero();  // valida il valore prima di passarlo alla variabile a
  printf("Inserisci il denominatore:");
  b=valida_intero(); // valida il valore prima di passarlo alla variabile b
  if (b == 0) {
    printf("Errore: Divisione per zero non consentita.\n"); // non permette la divisione per 0
    exit(EXIT_FAILURE);}
  int divisione = a / b;
  printf("La divisione tra %d e %d e': %d\n", a, b, divisione);}

void ins_string(){
    char stringa[10];  //abbiamo lasciato 10 come lunghezza del "testo", si può scegliere un qualsiasi valore a piacere
    printf("N.B.: Il massimo di caratteri consentiti per la stringa è 10"); // si avvisa l'utente del numero massimo di caratteri consentiti
    printf("\nInserisci la stringa:");
    getchar(); // Elimina il carattere 'a capo' lasciato nel buffer dalla scelta effettuata in menu()
    fgets(stringa, sizeof(stringa), stdin); // fgets per evitare buffer overflow
    printf("Hai inserito la stringa: %s\n", stringa);}

short int valida_short_int(){
  short int numero;
  while(1){
    printf("\n Inserisci un numero: ");
    if (scanf("%hd",&numero)==1){
      break;}
    else{
      while (getchar()!='\n'); //controlla se il dato inserito è un numero o una lettere, restituendo un messaggio di errore
      printf("\n\n Lettere non accettate, inserire un numero.\n");}}
  return numero;}

int valida_intero(){
  int numero;
  while(1){
    printf("\n Inserisci un numero: ");
    if (scanf("%d",&numero)==1){
      break;}
    else{
      while (getchar()!='\n'); //controlla se il dato inserito è un numero o una lettere, restituendo un messaggio di errore
      printf("\n\n Lettere non accettate, inserire un numero.\n");}}
  return numero;}
```