

Compito - Mendolia Valerio - Per agire come un Hacker bisogna capire come pensare fuori dagli schemi.

Il programma in allegato effettua dei calcoli in base alle scelte dell'utente utilizzando le varie funzioni richiamandole con uno switch.

Inizialmente il programma si compila, ma presenta degli errori di input che non consentono l'accesso alle varie funzioni.

Come prima analizzo il codice commentandolo in modo che riesco a trovare più informazioni possibili e gli eventuali fix da eseguire:

```
#include <stdio.h>
#include <string.h> // andrà ad aiutarmi nei filtri fix/
void menu ();
void moltiplica ();
void dividi ();
void ins_string();

int main ()
{
    //char scelta = {'\0'}; // errore?
    char scelta; // Fix
    menu ();
    scanf ("%c", &scelta); //fix
    //scanf ("%d", &scelta); //errore?
    // inserire un uscita dal programma?
    switch (scelta)
    {
        case 'A':
            moltiplica();
            break;
        case 'B':
            dividi(); // non si può dividere per zero si blocca il programma
            break;
        case 'C':
            ins_string(); // non lo metterei nel mio programma, apre le porte a molti tipi d'attacco.
            break;
    }
    return 0;
}
```

```

void moltiplica ()
{
    short int a,b = 0; // short Int? Per le moltiplicazioni non è il massimo, sostituire con int o float o double
    printf ("Inserisci i due numeri da moltiplicare:"); // solo numeri non caratteri
    scanf ("%f", &a); // si può mettere float al posto di int per i numeri con la virgola
    scanf ("%d", &b); // Perché qua è stato messo %d? non era meglio mettere entrambi su f?

    short int prodotto = a * b; // short Int? Per le moltiplicazioni non è il massimo, sostituire con int o float o double
    printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto); // il parametro a prima è dichiarato con f ora con d
}

void dividi ()
{
    int a,b = 0; //float? double?
    printf ("Inserisci il numeratore:"); // solo numeri non caratteri
    scanf ("%d", &a);
    printf ("Inserisci il denominatore:"); // si può mettere float al posto di int per i numeri con la virgola
    scanf ("%d", &b);

    int divisione = a % b; // non si può dividere per zero il programma si blocca

    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); // lunghezza stringa da filtrare, forse anche caratteri
    //stringa da stampare forse?
    //Bufferoverflow, fuzz testing?
}

//dopo le operazioni si può tornare al menù principale?

```

Successivamente individuo nel codice sorgente casistiche non standard, comportamenti potenziali non contemplati ed eventuali errori di sintassi o logici:

Verrà catalogato con “Prima” e “Dopo” con una spiegazione:

Prima:

```
#include <stdio.h>
```

Dopo:

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h> // andrà ad aiutarmi nei filtri fix/

```

Aggiungo la libreria string.h e stdlib.h, potrebbe aiutarmi a fare qualche filtro per le stringhe.

Prima:

```
char scelta = {'\0'};
```

Dopo:

```

//char scelta = {'\0'}; // errore?
char scelta; // Fix

```

Modifico la scelta errata del char in modo da far leggere l’input dell’utente.

Prima:

```
scanf ("%d", &scelta);
```

Dopo:

```
scanf (" %c", &scelta); //fix
```

Modifico il valore %d con %c di char per la risposta utente.

Prima:

```
// inserire un uscita dal programma?
switch (scelta)
{
    case 'A':
        moltiplica();
        break;
    case 'B':
        dividi(); // non si può dividere per zero si blocca il programma
        break;
    case 'C':
        ins_string(); // non lo metterei nel mio programma, apre le porte a molti tipi d'attacco.
        break;
}
```

Dopo:

```
// inserire un uscita dal programma?
switch (scelta)
{
    case 'A':
        moltiplica();
        break;
    case 'B':
        dividi(); // non si può dividere per zero si blocca il programma
        break;
    case 'C':
        ins_string(); // non lo metterei nel mio programma, apre le porte a molti tipi d'attacco.
        break;

    case 'D':
        return 0; // esco dal programma

    default: main(); // Ritorno al main in caso di input errato.
}
}
```

Cambio lo switch inserendo l'opzione D per uscire dal programma e il ritorno alla funzione main() con la funzione associata a switch(default) in caso di input errato dell'utente(Quindi non può inserire altro).

Vado ad aggiungere l'opzione D nel dialogo della funzione menu() (Questo screen non lo allego).

Continuo:

Prima:

```

void moltiplica ()
{
    short int a,b = 0; // short Int? Per le moltiplicazioni non è il massimo, sostituire con int o float o double
    printf ("Inserisci i due numeri da moltiplicare:"); // solo numeri non caratteri
    scanf ("%f", &a); // si può mettere float al posto di int per i numeri con la virgola
    scanf ("%d", &b); // Perché qua è stato messo %d? non era meglio mettere entrambi su f?

    short int prodotto = a * b; // short Int? Per le moltiplicazioni non è il massimo, sostituire con int o float o double

    printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto); // il parametro a prima è dichiarato con f ora con d
}

```

Dopo:

```

void moltiplica ()
{
    float a,b = 0; // short Int? Per le moltiplicazioni non è il massimo, sostituire con int o float o double
    printf ("\nInserisci il primo numero da moltiplicare:\n"); // solo numeri non caratteri
    scanf (" %f", &a); // si può mettere float al posto di int per i numeri con la virgola
    printf ("\nInserisci il secondo numero da moltiplicare:\n"); // solo numeri non caratteri
    scanf (" %f", &b); // Perché qua è stato messo %d? non era meglio mettere entrambi su f?

    if(a >100 || a <1){
        printf ("\n-Inserisci il primo numero compreso tra 1 e 100\n"); // controllo gli input inseriti dall'utente
        return moltiplica();
    } else if (b<1 || b>100){
        printf ("\nInserisci il primo numero compreso tra 1 e 100\n"); // controllo gli input inseriti dall'utente
        return moltiplica();
    } else {

        float prodotto = a * b; // short Int? Per le moltiplicazioni non è il massimo, sostituire con int o float o double
        printf ("\nIl prodotto tra %f e %f e': %f\n", a,b,prodotto); // il parametro a prima è dichiarato con f ora con d
    }

    main(); // Ritorno al inizio dopo l'operazione
}

```

Prima di tutto è stata cambiata la tipizzazione da 'short int' a 'float', così possiamo usare le virgole, successivamente è stato aggiunto un controllo numero per non far superare di troppo le moltiplicazioni e stampare il risultato corretto. E infine il ritorno al Main, per effettuare nuovamente alcune operazioni.

Prima:

```

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);
}

```

Dopo:

```

86
87 void dividi ()
88 {
89     int a,b = 0; //float? double? int?
90     printf ("\nInserisci il numeratore:\n"); // solo numeri non caratteri
91     scanf ("%d", &a);
92     printf ("\nInserisci il denominatore:\n"); // si può mettere float al posto di int per i numeri con la virgola
93     scanf ("%d", &b);
94
95
96     if(a == 0){
97         printf ("\nNon puoi dividere per zero, riprova\n"); // controllo gli input inseriti dall'utente
98         return dividi();
99     }
100     else if (a == 0){
101         printf ("\nNon puoi dividere per zero, riprova\n"); // controllo gli input inseriti dall'utente
102         return dividi();
103     }
104
105     else if (b == 0){
106         printf ("\nNon puoi dividere per zero, riprova\n"); // controllo gli input inseriti dall'utente
107         return dividi();
108     }
109     else {
110
111         int divisione = a / b; // non si può dividere per zero il programma si blocca tolgo il % che da il resto non fa la divisione
112         printf ("\nLa divisione tra %d e %d e': %d\n", a,b,divisione);
113     }
114 }
115 main(); //Ritorno al inizio dopo l'operazione
116 }
117

```

E' stata impedita la divisione per zero con dei filtri, successivamente ho cambiato il metodo divisionale invece di usare '%' che da il resto ho messo proprio l'operazione di divisione '/' Inserito anche in questo caso la modalità per tornare al menù principale.

Prima:

```

72
73 void ins_string ()
74 {
75     char stringa[10];
76     printf ("Inserisci la stringa:");
77     scanf ("%s", &stringa);
78 }
79
80

```

Dopo:

```

122 void ins_string ()
123 {
124     char stringa[20];
125     printf ("Inserisci la stringa:");
126     scanf ("%s", stringa);
127     while(strlen(stringa) > 20 || strlen(stringa) < 1) // controllo lunghezza e se viene inserito l'username
128     {
129         printf ("\nErrore, stringa non inserita oppure ha superato il massimo dei caratteri consentiti(20) .");
130         printf ("\n Riprova inserisci il tuo la tua stringa(Massimo 20 caratteri): ");
131         scanf ("%s", stringa);
132     }
133     printf ("\nLa mia stringa e' : %s\n", stringa);
134     //stringa da stampare forse?
135     //Bufferoverflow, fuzz testing?
136     main(); //Ritorno al inizio dopo l'operazione
137 }
138
139 //dopo le operazioni si può tornare al menù principale?

```

In questo caso è stato messo un filtro per i caratteri(andrebbe messo un filtro migliore per evitare attacchi di tipo bufferoverflow o resistere a fuzz testing) ma in questo caso va più che bene.

Il programma stampa una stringa di massimo 20 caratteri dichiarata dall'utente, se supera i 20 caratteri viene bloccato.

Successivamente ritorna al menù.

Allego questo PDF più il nuovo codice C.

**Mendolia Valerio**