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:

```c
#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;

}
```

```c
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); // Perchè 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 denumeratore:"); // 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:

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

Dopo:

```c
#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:

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

Dopo:

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

Modifico la scelta errata del char in modo dar 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:

```c
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); // Perchè 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:

```c
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); // Perchè 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 stato 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:

```c
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);
}
```

Dopo:

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


 if(a == 0){
    printf ("\nNon puoi dividere per zero, riprova\n");  // controllo gli input inseriti dall'utente
    return dividi();

    } else if (a == 0){
      printf ("\nNon puoi dividere per zero, riprova\n");  // controllo gli input inseriti dall'utente
    return dividi();


    } else if (b == 0){
      printf ("\nNon puoi dividere per zero, riprova\n");  // controllo gli input inseriti dall'utente
    return dividi();

    } else {

        int divisione = a / b; // non si può dividere per zero il programma si blocca tolgo il % che da il resto non fa la divisione
        printf ("\nLa divisione tra %d e %d e': %d\n", a,b,divisione);
    }

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

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:

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

Dopo:

```c
void ins_string ()
{
    char stringa[20];
        printf ("Inserisci la stringa:");
        scanf("%s", stringa);
while(strlen(stringa) > 20 || strlen(stringa) < 1) // controllo lunghezza e se viene inserito l'username
{
printf("\nErrore, stringa non inserita oppure ha superato il massimo dei caratteri consentiti(2) .");
printf("\n Riprova inserisci il tuo nome(Massimo 20 caratteri): ");
scanf("%s", stringa);
}
 printf ("\nLa mia stringa e' : %s\n", stringa);
        //stringa da stampare forse?
        //Bufferoverflow, fuzz testing?
        main(); //Ritorno al inizio dopo l'operazione
}

//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