

CACCIA AGLI INSETTI

TASK

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

ANALISI E VALUTAZIONI

1. Iniziamo nell'analizzare il codice, andando a descrivere il suo funzionamento e successivamente l'intero codice:

FUNZIONAMENTO: Analizzando il codice la sua funzione sta proprio nell'essere un "assistente digitale" capace di effettuare una tra le tre opzioni proposte all'utente: Moltiplicazione tra due numeri, Divisione tra due numeri o Inserire una stringa.

CODICE:

- Int main (): Partendo dall'alto del codice ed esaminandolo la prima cosa che ci interessa ed è la prima funzione che viene letta in esecuzione è il "Main()" preceduto da "int" che sta ad indicare proprio il tipo di funzione "main", in questo caso NUMERO INTERO.
- char scelta: è un tipo di variabile simile all'INT ma che però sta ad indicare Carattere, pertanto verrà rilasciato in questo caso un output carattere.
- void (): All'interno del codice ritroviamo funzioni del tipo "Void()" le quali sono quelle funzioni che non restituiscono alcun valore di ritorno, sta di fatto che non si inserisce l'istruzione "return".
- Switch (): Tale funzione permette di eseguire diversi CASE (casi) nel caso in cui viene effettuata una scelta. Nel caso del codice se si dovesse scegliere A allora verrà eseguita una determinata istruzione, nel caso B un'altra e così via. La funzione Switch, per ciascun CASE termina con il comando BREAK che sta ad indicare "l'uscita" dalla funzione chiamante una volta che l'istruzione del CASE è stata eseguita
- printf(): è un tipo di funzione che permette di stampare a schermo i caratteri che vengono inclusi all'interno dei doppi apici.
- scanf(): è la funzione che permette di leggere l'input che viene inserito dall'utente via tastiera. È fondamentale andare a specificare di che tipo sarà l'input nel caso del codice

abbiamo più variabili “%c” che indica una variabile di tipo CHAR (Carattere) ed “%d” indica una variabile di tipo INT (intero). Il simbolo % sta ad indicare dove verrà inserita la variabile, mentre il simbolo & sta ad indicare che la variabile che abbiamo indicizzato all’inizio verrà associata a quel tipo di input.

- return 0: Indica il completamento con successo della funzione generata e il codice si chiude.

2. Andiamo ad individuare nel codice sorgente le casistiche non standard che il programma non gestisce nel tempo ed eventuali errori di sintassi e logici:

Char: Partendo dal primo scanf all’interno del char abbiamo che la variabile è di tipo “%d” ma per fare in modo che la scelta, impostata con caratteri (A,B,C) possa andare a buon fine c’è bisogno che inseriamo “%c”, altrimenti il programma si chiuderà sempre.

Successivamente andando a selezionare altri tasti all’infuori delle 3 opzioni il programma si sarebbe chiuso pertanto dobbiamo considerare questo aspetto per poi andare a modificarlo in fase di errori logici.

Void moltiplicazione: Per quanto riguarda il void della moltiplicazione lo short dell’int non ci darebbe la possibilità di inserire i numeri effettivi, o meglio che il numero inserito sarebbe stato sempre 0 errore dovuto anche al “%f” il quale dobbiamo cambiarlo in “%d” per dare possibilità che l’input e output sia effettivamente un numero.

Void divisione: All’interno di quest’ultimo ritroviamo un particolare errore dovuto all’operatore in quanto non sarà % bensì /.

Void stringa: In questa funzione invece non ritrovo nessun utilità in quanto se l’utente dovesse inserire la stringa, per come è stato realizzato il codice, non darebbe nemmeno possibilità di riprodurla.

3. Andiamo ora ad ottimizzare il codice:

Char: Partendo dal principio il programma non si sarebbe avviato se l’utente avesse inserito altre lettere o numeri al di fuori delle 3 opzioni, pertanto ho avviato una funzione “do” dal quale sarebbe apparso la scritta “inserisci i valori esatti” per dare possibilità di un nuovo inserimento il tutto tramite un default alla fine dei 3 case. Il “Rewind” mi permette di riproporre una nuova possibilità all’utente.

Void moltiplicazione: Al suo interno ho dato modo all’utente di inserire due numeri differenti in due stampe diverse in quanto provando ad inserire due numeri attaccati sarebbe risultato come unico e quindi l’utente avrebbe potuto non capire di dover inserire lo spazio, pertanto ho aggiunto un ulteriore printf con successivo scanf. Inoltre ho eliminato lo short del primo int per permettere di eseguire l’operazione.

Void dividi: In quest’ultimo non ho riscontrato nessun problema inserendo i numeri, ma dal momento in cui si andavano ad inserire numeri con la virgola o lettere il sistema mi dava errore.

Void ins_string: In quest’ultima ho scelto di inserire un ulteriore printf finale per far apparire a schermo la stringa di massimo 10 caratteri scritta dall’utente. Ho inserito anche nel primo printf un indicazione all’utente della lunghezza massima che la stringa avrebbe dovuto avere.

L'ESECUZIONE DEL CODICE SI PRESENTA COSÌ:

```
(kali㉿kali)-[~/Desktop]
$ ./Esercizio_10_Epicode
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
D
Inserisci una delle tre opzioni
Inserisci una delle tre opzioni
2
Inserisci una delle tre opzioni
Inserisci una delle tre opzioni
A
Inserisci i due numeri da moltiplicare
Numero uno: 4

Numero due: 7

Il prodotto tra 4 e 7 e': 28
```

```
(kali㉿kali)-[~/Desktop]
$ ./Esercizio_10_Epicode
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: 4

Inserisci il denominatore: 4

La divisione tra 4 e 4 e': 1
```

```
(kali㉿kali)-[~/Desktop]
$ ./Esercizio_10_Epicode
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 una stringa di massimo 10 caratteri:Epicode
La tua stringa è: Epicode
```