Esercizio S2/L5 Bug Hunting

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

Vincenzo Colletta 06/10/2023

Codice di partenza

```
#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);
}
```

Leggendo il codice, possiamo apprendere che questo programma è composto da 4 funzioni void, nella quale è richiesto all'utente, di scegliere tramite "void menù", che operazione svolgere tra quelle proposte, e poi appunto, eseguire l'operazione scelta con l'inserimento di due numeri.

Comportamenti non contemplati

Le alternative non contemplate sono ad esempio l'inserimento di una lettera che sia diversa da 'a' 'b' o 'c', nel void menù, nel caso infatti in cui dovessimo inserire una lettere diversa, il programma terminerà. Avremmo potuto risolvere questa eventuale possibilità inserendo una struttura di controllo condizionale* 'if', stabilendo che se la lettera inserita non è una di quelle, allora il programma visualizza una scritta con ad esempio "Selezionare una delle opzioni disponibili", e rimandare l'utente sul menù di scelta.

Un altro comportamento non contemplato è la possibilità di avere dei numeri decimali, nel caso in cui il risultato di una divisione non sia un numero intero.

In quel caso avremmo dovuto cambiare il tipo di variabile, da intera a decimale, e anche la lettura non sarà piu %d ma sarà %lf.

**Una struttura di controllo condizionale permette al programma di eseguire istruzioni diverse a seconda di una condizione booleana(vero/falso)*

Individuazione e correzione errore 1

```c
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;

}
```

Errore 1

Dichiarazione errata di scelta:

La variabile scelta è dichiarata come un carattere (char), ma nel switch viene confrontata con caratteri tra apici singoli ('A', 'B', 'C'). La dichiarazione di tipo char, è corretta nella prima immagine, ma l'input dovrebbe essere letto come carattere con %c anziché %d. Ecco come dovrebbe essere dichiarata e letta:

```c
int main ()

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

    switch (scelta)
    {
        case 'A':
        moltiplica();
        break;
        case 'B':
                dividi();
                break;
        case 'C':
                ins_string();
                break;
    }

return 0;

}
```

Individuazione e correzione errore 2

```c
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);
}
```

```c
void moltiplica ()
{
    int  a,b = 0;
    printf ("Inserisci i due numeri da moltiplicare:");
    scanf ("%d", &a);
    scanf ("%d", &b);

    int prodotto = a * b;

    printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);
}
```

Nella funzione moltiplica(), stiamo leggendo le due variabili, a e b, come ‘short int’ ma stiamo usando: ‘%f’ per leggere a, e ‘%d’ per leggere b.
Dovremmo invece utilizzare lo stesso tipo di dato per la dichiarazione e la lettura.
Quindi utilizzare **‘%d’** per entrambi.

Individuazione e correzione errore 3

```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);
}
```

```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);
}
```

Nella funzione dividi(), stiamo calcolando la divisione tra a e b, ma nell'immagine a sinistra vediamo che il simbolo %, non è corretto per la nostra operazione, lo modifichiamo quindi con /.

Individuazione e correzione errore 4

```
81  void ins_string ()
82  {
83      char stringa[10];
84          printf ("Inserisci la stringa:");
85          scanf ("%s", &stringa);
86  }
```

```
81  void ins_string ()
82  {
83      char stringa[10];
84          printf ("Inserisci la stringa:");
85          scanf ("%s", stringa);
86  }
```

Nella funzione ‘ins_string ()’, stiamo leggendo utilizzando ‘scanf’ con ‘%s’, ma ‘stringa’ è gia un array di caratteri, quindi non è necessario utilizzare l’operatore di riferimento’&’.

Fin.