

REPORT BUG HUNTING

Il programma simula il funzionamento di un assistente digitale.

All'avvio, viene presentato all'utente un menù con le funzioni disponibili e la possibilità di scegliere una di essa:

A) Per eseguire la moltiplicazione tra due numeri inseriti dall'utente

B) Per eseguire la divisione tra due numeri inseriti dall'utente

C) Per far inserire una stringa all'utente.

Una volta effettuata la scelta da parte dell'utente, viene eseguita la funzione corrispondente previo inserimento dei due numeri necessari per le operazioni o della stringa.

Vengono, infine, visualizzati a schermo i due numeri inseriti e il risultato dell'operazione, in caso di scelta A) o B) nel menù.

Andando più nel dettaglio del codice fornitoci, passo ora ad analizzare il suo funzionamento.

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

```
void menu ();  
void moltiplica ();  
void dividi ();  
void ins_string();
```

→ Caricamento in memoria della libreria <stdio.h> per le funzioni di I/O (printf/scanf)

→ Inizializzazione delle funzioni che si useranno nel programma

```
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;  
}
```

Funzione principale di un programma in linguaggio C:

Viene dichiarata ed inizializzata vuota la variabile "scelta" di tipo char, viene eseguita la funzione menù() e data la possibilità all'utente di inserire la propria scelta.

Viene gestita la scelta con lo switch, istruzione di controllo condizionale, in cui il caso 'A' porta all'esecuzione della funzione moltiplica(), il caso B) all'esecuzione della funzione dividi() ed il caso 'C' all'esecuzione della funzione ins_stringa.

Return 0 in quanto la funzione main è di tipo intero e quindi si aspetta come risultato un dato di quel tipo.

```

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");
}

```

Funzione menù:

Attraverso la funzione printf, viene stampato a schermo il menù che servirà all'utente per capire il funzionamento del programma e le opzioni fra le quali può scegliere.

```

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);
}

```

Funzione moltiplica:

Vengono dichiarate ed inizializzate a 0 le variabili a e b che rappresentano i due numeri che inserirà l'utente. Viene detto all'utente di inserire i due numeri che vuole moltiplicare e data la possibilità di farlo. Viene dichiarata la variabile prodotto come $a*b$ e, infine, vengono stampati i due numeri inseriti dall'utente ed il loro prodotto.

```

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);
}

```

Funzione dividi:

Vengono dichiarate ed inizializzate a 0 le variabili a e b che rappresentano i due numeri che inserirà l'utente. Viene detto all'utente di inserire i due numeri che vuole dividere e data la possibilità di farlo. Viene dichiarata la variabile divisione come a/b e, infine, vengono stampati i due numeri inseriti dall'utente e la loro divisione.

```

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

```

Funzione ins_string:

Viene dichiarato un array "stringa" di tipo char di dimensione 10. Viene detto all'utente di inserire la stringa e data la possibilità di farlo.

Passo ora ad analizzare il codice per individuare le **casistiche non standard** che il programma potrebbe non gestire:

- Nella funzione principale non viene gestito il caso in cui l'utente inserisca una lettera che non sia una di quelle presenti nel menù.
- Nella funzione dividi non viene gestito il caso in cui il denominatore sia uguale a 0.
- Nella funzione ins_string non viene gestito il caso in cui la stringa inserita abbia più caratteri della dimensione dell'array contenente la stringa.
- Nelle funzioni moltiplica e dividi non viene gestito il caso in cui si inseriscano delle lettere e non dei numeri.

Analizzo nuovamente il codice per individuare eventuali **errori di sintassi o errori logici**:

```
#include <stdio.h>
void menu ();
```

Manca `#include<string.h>` per caricare la libreria con le funzioni necessarie alla gestione delle stringhe

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

`%d` è per variabili di tipo `int`, essendo "scelta" una variabile di tipo `char`, nello `scanf` ci vuole `%c`

```
void menu ()
{
    printf ("Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti\n");
    printf ("Come posso aiutarti?\n");
}
```

Errore ortografico:
"assistente" non "assidente"

```
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;
```

`Void` è per funzioni che non restituiscono nessun tipo di risultato, in questo caso serve una funzione di tipo `float`

Il tipo di variabile "short int" appartiene al linguaggio C++, in C o è "short" o è "int". Trattandosi di una moltiplicazione, ci si potrebbe aspettare numeri molto grandi quindi meglio "int". Così facendo però, non si accetterebbero numeri decimali, quindi la soluzione migliore è avere tutte le variabili da utilizzare (a, b e prodotto) di tipo "float".

Decidendo di usare "float", ma scrivendo come nel codice "float a,b=0;" solo b è inizializzata a 0.

Per avere anche a=0 o si mettono su due righe separate "float a=0;" "float b=0;" oppure sulla stessa riga "float a=0, b=0;"

Infine, "%f" all'interno dello `scanf` è per variabili di tipo `float`, ma in questo caso si hanno due variabili di tipo `int`. Prendendo in considerazione, però, la soluzione in cui le variabili sono di tipo `float`, diventa sbagliato il `%d` nel secondo `scanf` in cui ci andrebbe `%f`.

```

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);
}

```

Void è per funzioni che non restituiscono nessun tipo di risultato, in questo caso serve una funzione di tipo float

Trattandosi di una divisione, ci si può aspettare sia numeri inseriti con la virgola. Per questo motivo il tipo delle variabili deve essere float. Scrivendo come nel codice “float a,b=0;” solo b è inizializzata a 0. Per avere anche a=0 o si mettono su due righe separate “float a=0;” “float b=0;” oppure sulla stessa riga “float a=0, b=0;”.

Essendo quindi di tipo float negli scanf ci va %f.

Errore ortografico: “denominatore” non “denumeratore”.

Anche la variabile “divisione” deve essere di tipo float e quindi nel printf ci devono essere i %f e non i %d che si riferiscono a variabili di tipo int.

Il segno % restituisce il resto della divisione, se si vuole il risultato della divisione bisogna usare /.

```

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

```

Nello scanf non ci va la & prima di stringa in quanto si tratta di un array di caratteri.

Manca, inoltre, un printf alla fine (prima della parentesi graffa) per stampare la stringa inserita.

Come ultimo passaggio, vado a correggere il codice per risolvere le casistiche non standard che il programma non gestisce e sistemare tutti gli errori sintattici/logici riscontrati in precedenza.

Aggiungo un caso 'D' per far sì che se l'utente avesse attivato l'assistente senza bisogno, possa uscire senza dover necessariamente svolgere una delle sue funzioni. Per fare ciò implemento una nuova funzione che chiamo uscita() e la chiamo nel caso 'D' dello switch.

Le parti corrette o aggiunte sono evidenziate con dei commenti.

```
#include <stdio.h>
#include <string.h> //aggiungo la libreria string.h per gestire le funzioni per la stringa

void menu();
float dividi(); // inizializzo la funzione dividi() di tipo float
float moltiplica(); // inizializzo la funzione moltiplica() di tipo float
void ins_string();
void uscita(); // funzione aggiunta per gestire il caso 'D' dello switch in caso l'utente voglia uscire dal programma

int main()
{
    char scelta ={'\0'};

    menu();
    scanf(" %c", &scelta);

    /* Implemento un ciclo while per il controllo sulla lettera inserita.
     * Quando viene inserita una lettera diversa da quelle nel menu', fa ripetere l'inserimento.
     */
    while(scelta != 'A' && scelta != 'B' && scelta != 'C' && scelta != 'D')
    {
        printf("\nInserimento errato. Scegliere A, B, C o D: ");
        scanf(" %c", &scelta);
    }

    switch (scelta)
    {
        case 'A':
            moltiplica();
            break;
        case 'B':
            dividi();
            break;
        case 'C':
            ins_string();
            break;
        case 'D':
            uscita(); //aggiungo il caso 'D' per far uscire l'utente dall'assistente
                    // chiamo la nuova funzione per l'uscita dal programma
    }

    return 0;
}

void menu()
{
    printf("Benvenuto, sono un assistente digitale, posso aiutarti a sbrigare alcuni compiti\n"); //errore di ortografia corretto
    printf("Come posso aiutarti?\n");
    printf("A >> Moltiplicare due numeri\nB >> Dividere due numeri\nC >> Inserire una stringa\nD >> Esci\n"); //aggiunta della scelta D
}

float moltiplica() //funzione di tipo float
{
    float a=0, b=0; //variabili di tipo float
    float prodotto=0;

    printf("\nInserisci i due numeri da moltiplicare:\n");
    scanf(" %f", &a);
    scanf(" %f", &b);

    prodotto=a*b;

    printf("\n\nIl prodotto tra %.2f e %.2f e': %.2f\n", a,b, prodotto);
    // metto %f perche' variabili float. Aggiungo .2 dopo il segno % per avere solo due numeri dopo la virgola

    return 0; //aggiunta del return perche' non e' piu' una funzione di tipo void
}

float dividi() //funzione di tipo float
```

```

float dividi() //funzione di tipo float
{
    float a=0, b=0;          // variabili di tipo float
    float divisione=0;

    printf("\nInserisci il numeratore: ");
    scanf(" %f", &a);

    printf("\nInserisci il denominatore: ");          // corretto errore ortografico
    scanf(" %f", &b);

    /* implemento un ciclo while per il controllo sul denominatore che non puo' essere 0
    * ogni volta che e' 0, fa ripetere l'inserimento.
    */
    while(b==0)
    {
        printf("\nIl denominatore non puo' essere 0!!\nInserisci nuovamente il denominatore: ");
        scanf(" %f", &b);
    }

    divisione=a/b; // corretto il simbolo % con / per eseguire la divisione

    printf("\nLa divisione tra %.2f e %.2f e': %.2f\n", a,b, divisione);
    // metto %f perche' variabili float. Aggiungo .2 dopo il segno % per avere solo due numeri dopo la virgola

return 0;          //aggiunta del return perche' non e' piu' una funzione di tipo void
}

void ins_string ()
{
    char stringa[10];

    printf("\nInserisci la stringa: ");
    scanf(" %s", stringa); // tolto & prima di stringa

    /* Aggiungo un ciclo while per il controllo sulla lunghezza della stringa inserita.
    * Fa ripetere l'inserimento della stringa fino a quando la dimensione non e' pari alla dimensione dell'array
    */
    while(strlen(stringa)>10)
    {
        printf("\nStringa inserita troppo lunga. Riprova.");
        printf("\nInserisci la stringa: ");
        scanf(" %s", stringa);
    }

    printf("\nStringa inserita: ' %s ', stringa); //aggiungo questo printf per far stampare a schermo la stringa inserita dall'utente
}

void uscita() //nuova funzione per far uscire l'utente dal programma direttamente dal menu'
{
    printf("\nGrazie di avermi usato. Alla prossima :)\n");
}

```