

PROGETTO SETTIMANA 2

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

1- CAPIRE COSA FA IL PROGRAMMA:

Il programma simula un assistente virtuale che può svolgere 3 funzioni, l'utente può scegliere dal menù:

- A-Eseguire la moltiplicazione tra due numeri e stamparne il risultato
- B-Eseguire la divisione tra due numeri e stamparne il resto
- C-Inserimento di una stringa formata da max[10] caratteri

2-INDIVIDUARE DAL CODICE SORGENTE LE CASISTICHE NON STANDARD CHE IL PROGRAMMA NON GESTISCE:

- Il programma non gestisce il caso in cui l'utente, all'interno del menù di scelta della funzione, inserisca un carattere diverso da A, B o C e non ne prevede un messaggio di errore. Inoltre non prevede.
- Il programma non gestisce la situazione in cui ci sia 0 come denominatore della divisione.
- Il programma non gestisce il caso in cui ci sia una stringa più lunga di [10] caratteri.
- Il programma non gestisce il caso in cui l'utente inserisca valori non numerici nel caso della moltiplicazione o divisione.

3-INDIVIDUARE EVENTUALI ERRORI DI SINTASSI/LOGICI, PROPORRE UNA SOLUZIONE.

Per prima cosa la scelta della funzione dal menù in quanto la variabile char scelta vuole %c invece che %d (usata per gli int):

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

-Nello switch manca un default case nel caso l'utente inserisca un carattere sbagliato

```
char scelta;

while (1) {
    menu();
    if (scanf(" %c", &scelta) != 1 || (scelta != 'A' && scelta != 'B' && scelta != 'C')) {
        printf("Scelta non valida. Riprova.\n");
        continue;
    }

    switch (scelta) {
        case 'A':
            moltiplica();
            break;
        case 'B':
            dividi();
            break;
        case 'C':
            ins_string();
            break;
        default:
            printf("Scelta non valida. Riprova.\n");
            break;
    }

    break;
}

return 0;
}
```

- Nella funzione moltiplica() cambiamo short int in "int" e per leggere utilizziamo %d così da essere sicuri di leggere il carattere giusto

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

    short int prodotto = a * b;

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

void dividi ()
```

-La funzione divisione con l'operatore % restituisce il resto della divisione intera ma non il risultato della divisione, quindi sostituiamo "%" con "/" e gestiamo anche il caso in cui l'utente inserisca uno 0 al denominatore

```
void dividi ()
{
    int a, b;
    printf("Inserisci il numeratore: ");
    scanf("%d", &a);
    printf("Inserisci il denominatore diverso da zero: ");
    scanf("%d", &b);

    if (b != 0) {
        float divisione = (float)a / b;
        printf("La divisione tra %d e %d è: %.2f\n", a, b, divisione);
    } else {
        printf("Impossibile dividere per zero.\n");
    }
}
```

-Nella funzione stringa aggiungiamo un printf per stampare a video il contenuto di stringa

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

-Tornando alla moltiplicazione gestiamo il caso in cui l'utente non inserisca un numero, inseriamo un messaggio di errore e richiediamo il numero

```

void moltiplica (){
int a, b;
char input[20];

printf("Inserisci il primo numero da moltiplicare:\n");

while (1) {
    printf("Numero 1: ");
    if (fgets(input, sizeof(input), stdin) != NULL) {
        if (sscanf(input, "%d", &a) == 1) {
            break; // Esci dal ciclo se l'input è un numero valido
        } else {
            printf("Input non valido. Inserisci un numero valido.\n");
        }
    }
}

printf("Inserisci il secondo numero da moltiplicare:\n");

while (1) {
    printf("Numero 2: ");
    if (fgets(input, sizeof(input), stdin) != NULL) {
        if (sscanf(input, "%d", &b) == 1) {
            break; // Esci dal ciclo se l'input è un numero valido
        } else {
            printf("Input non valido. Inserisci un numero valido.\n");
        }
    }
}

int prodotto = a * b;
printf("Il prodotto tra %d e %d è: %d\n", a, b, prodotto);
}

```

- Ora facciamo la
stessa cosa per la

funzione dividi

```

void dividi(){
int a, b;
char input[20];

printf("Inserisci il numeratore:\n");

while (1) {
    printf("Numeratore: ");
    if (fgets(input, sizeof(input), stdin) != NULL) {
        if (sscanf(input, "%d", &a) == 1) {
            break; // Esci dal ciclo se l'input è un numero valido
        } else {
            printf("Input non valido. Inserisci un numero valido.\n");
            input[0] = '\0'; // Resetta l'input
        }
    }
}

printf("Inserisci il denominatore diverso da zero:\n");

while (1) {
    printf("Denominatore: ");
    if (fgets(input, sizeof(input), stdin) != NULL) {
        if (sscanf(input, "%d", &b) == 1 && b != 0) {
            break; // Esci dal ciclo se l'input è un numero valido e diverso da zero
        } else {
            printf("Input non valido o denominatore zero. Inserisci un numero valido e diverso da zero.\n");
            input[0] = '\0'; // Resetta l'input
        }
    }
}

float divisione = (float)a / b;
printf("La divisione tra %d e %d è: %.2f\n", a, b, divisione);
}

```

- Infine ho riscritto il codice con un'indentazione migliore per renderlo più leggibile (vedi prova.c)