

## Analisi funzionalità programma da codice:

Il programma è un assistente digitale che presenta un menù all'utente, il menù presenta 3 possibili scelte tra cui: A moltiplicare due numeri, B dividere due numeri, C inserire una stringa.

Stranamente non è stato previsto un comando che l'utente possa premere per chiudere il programma, inoltre il programma non prevede la possibilità di eseguire più operazioni durante lo stesso lancio.

Non è chiaro che tipo di numeri si vogliano moltiplicare, in quanto la dichiarazione è sbagliata, inoltre quando si chiede in input il primo ci si aspetta un float.

## Gestione casistiche non standard:

```
scanf ("%d", &scelta);      1) In questa sezione di codice non viene gestito un eventuale
switch (scelta)           input non richiesto, come ad esempio un lettera non presente,
{                         una minuscola o un numero.
    case 'A':            case 'B':
        moltiplica();     dividi();
        break;             break;
    case 'C':            ins_string();
        break;             break;
}
void moltiplica ()          2) in questo metodo non viene gestito un eventuale input di stringhe, inoltre non si impone
{                           nessun limite alla lunghezza del numero inserito, ciò permetterebbe di superare la grandezza
    short int a,b = 0;    massima di memoria che il tipo di variabile può memorizzare causando errori.
    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);
}
```

2) in questo metodo non viene gestito un eventuale input di stringhe, inoltre non si impone nessun limite alla lunghezza del numero inserito, ciò permetterebbe di superare la grandezza massima di memoria che il tipo di variabile può memorizzare causando errori.

```

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);
}

```

3) Qui allo stesso modo del metodo precedente non viene gestito un eventuale input non richiesto come una stringa, e come sopra non si imposta nessun limite alla grandezza del numero.

```

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

```

4) in questo metodo invece si vuole limitare a 10 la dimensione della stringa, però poi si accetta in input qualsiasi lunghezza.

## Analisi e risoluzione errori logici/sintattici:

In tutto il programma gli scanf non hanno lo spazio davanti alla percentuale, questo fa sì che alcuni di essi verranno saltati dal 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;
}

```

< Questa dichiarazione è errata, sarebbe bastato: char scelta;

< Questo scanf si aspetta un int in input, che è errato dato che abbiamo dichiarato un char per la variabile "scelta", inoltre lo switch si aspetta una lettera maiuscola. Bisogna correggere con "%c", e possibilmente elevare a maiuscolo l'input o ripetere la richiesta fino a che non viene inserita la lettera richiesta in maiuscolo

< In questo switch un errore logico di progettazione è non prevedere un comando di uscita, inoltre come detto sopra non si prevede il riutilizzo del programma.

```

void menu () // questo metodo non presenta errori
{
    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");
}

void moltiplica ()
{
    short int a,b = 0; // dichiarazione errata
    printf ("Inserisci i due numeri da moltiplicare:");
    scanf ("%f", &a); // input errato, va usata la d per un int
    scanf ("%d", &b);

    short int prodotto = a * b; // dichiarazione errata di prodotto

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

```

Nel metodo moltiplica invece è errata la dichiarazione delle variabili, se vogliamo degli int la dichiarazione corretta sarebbe stata: int a,b; e poi eventualmente assegnare 0

Il primo scanf si aspetta un float, che non corrisponde al tipo di variabile che abbiamo dichiarato

Allo stesso modo di a e b, anche prodotto è dichiarata in modo errato, la dichiarazione corretta sarebbe stata: int prodotto = a\*b;

```

void dividi ()
{
    int a,b = 0; // anche questa dichiarazione è errata
    printf ("Inserisci il numeratore:");
    scanf ("%d", &a);
    printf ("Inserisci il denominatore:");
    scanf ("%d", &b);

    int divisione = a % b; // operazione errata

    printf ("La divisione tra %d e %d e': %d", a,b,divisione);
}

```

Anche qui la dichiarazione non è corretta, almeno non logicamente, in questo modo assegniamo 0 solo a b, avrebbe più senso farlo direttamente su righe separate anche per una questione di leggibilità.

La divisione in questo caso è logicamente errata in quanto l'operazione usata è modulo, che restituisce il resto della divisione.

```

void ins_string ()
{
    char stringa[10];
    printf ("Inserisci la stringa:");
    scanf ("%s", &stringa);
    // che scopo ha? nemmeno lo stampiamo
}

```

Qui invece l'errore è solo logico, facciamo inserire la stringa ma non ci facciamo niente, sarebbe sensato almeno stamparla.

## Programma con le risoluzioni applicate:

The screenshot shows a terminal window titled "alessio@kali: ~/Desktop/Esercizi/Week2/d5w2". The file being edited is "EsD5W2.c". The code is a menu-driven program in C. It includes headers for stdio.h, ctype.h, stdbool.h, string.h, and limits.h. The main function initializes a variable "first" to 1 and a character variable "scelta". It then enters a do-while loop where it calls "menu(first)". Inside the loop, it reads a character from the user and converts it to uppercase if it's lowercase. It then checks if the input is one of four letters ('A', 'B', 'C', or 'D'). If it's not, it prints an error message. If it is, it calls the corresponding function: "moltiplica()" for 'A', "dividi()" for 'B', "ins\_string()" for 'C', and prints "Arrivederci!" for 'D'. After each operation, it increments "first" by 1. The program ends with a return 0. The terminal also shows a series of keyboard shortcuts at the bottom.

```
GNU nano 6.4 EsD5W2.c *
#include <stdio.h>
#include <ctype.h>
#include <stdbool.h>
#include <string.h>
#include <limits.h>

void menu(bool firstRun);
void moltiplica ();
void dividi ();
void ins_string();

int main () {
    bool first = 1; // variabile che controlla se il programma ha già eseguito un'operazione
    char scelta; // dichiara la variabile che uso per scegliere
    do{ // questo ciclo permette di eseguire operazioni finché non si preme il tasto per uscire, che ferma il programma
        menu(first); // metodo per stampare il menù, esso è diverso se non è il primo ciclo
        do{ // questo ciclo si ripete finché l'input è diverso da una delle 4 lettere
            scanf (" %c", &scelta); // la lettera inserita potrebbe ancora essere minuscola
            if(islower(scelta)){
                scelta = toupper(scelta); // la trasformo in maiuscola
            }
            if(scelta!= 'A' && scelta != 'B' && scelta != 'C' && scelta != 'D'){
                printf("Solo le lettere del menù sono possibili \n");
            }
            if(isdigit(scelta)){
                printf("I numeri non sono ammessi \n");
            }
        } // la condizione di questo while fa sì che si possa uscire solo se si sceglie a, b, c o d
       }while(scelta!= 'A' && scelta != 'B' && scelta != 'C' && scelta != 'D' && isdigit(scelta));
        switch (scelta)
        {
            case 'A':
                moltiplica();
                break;
            case 'B':
                dividi();
                break;
            case 'C':
                ins_string();
                break;
            case 'D':
                printf("Arrivederci ! \n");
                return 0;
        }
        first = 0; // una volta qui il programma prima di tornare all'inizio aggiorna la variabile
    }while(1);
}

^G Help      ^O Write Out     ^W Where Is     ^K Cut      ^T Execute      ^C Location      M-U Undo
^X Exit      ^R Read File     ^\ Replace      ^U Paste     ^J Justify      ^/ Go To Line     M-E Redo
```

```

GNU nano 6.4                                         EsD5W2.c *
first = 0; // una volta qui il programma prima di toranre all'inzio aggiora la variabile
}while(1);

}

void menu (bool firstRun){ // questo metodo non presenta errori

    if(firstRun){
        printf ("Benvenuto, sono un assitente digitale, posso aiutarti a sbrigare alcuni compiti\n");
        printf ("Come posso aiutarti?\n");
    }else{
        printf ("Vuoi fare altro?\n");
    }
    printf ("A >> Moltiplicare due numeri \n");
    printf ("B >> Dividere due numeri \n");
    printf ("C >> Inserire una stringa \n");
    printf ("D >> Uscire dal programma \n");
}

void moltiplica (){
    int a, b, prodotto, check; // la variabile check controlla che il vaoler inserito sia un int
    do { // ripeto la richiesta finche l'utente non inserisce un int
        printf("Inserisci il primo numero: ");
        check = scanf(" %d", &a); // check = 1 se int
        if (check != 1) {
            printf("Inserisci un numero intero\n");
            getchar();
        }
    } while (check != 1);
    do {
        printf("Inserisci il secondo numero: ");
        check = scanf(" %d", &b);
        if (check != 1) {
            printf("Inserisci un numero intero\n");
            getchar();
        }
    } while (check != 1);
    prodotto = a * b;

    if (prodotto >= INT_MIN && prodotto <= INT_MAX) { // il prodotto dei numeri inseriti potrebbe essere troppo grande
        printf("Il prodotto dei due numeri è: %d\n", prodotto);
    } else {
        printf("Il prodotto non può essere rappresentato come int\n");
    }
}

void dividi (){
    int a, b, check; // la variabile check controlla che il vaoler inserito sia un int
    float divisione;
    do { // ripeto la richiesta finche l'utente non inserisce un int

```

```

        printf("Inserisci il numeratore: ");
        check = scanf(" %d", &a); // check = 1 se int
        if (check != 1) {
            printf("Inserisci un numero intero\n");
            getchar(); // svuoto il buffer
        }
    } while (check != 1);
    do {
        printf("Inserisci il denominatore: ");
        check = scanf(" %d", &b);
        if (check != 1) {
            printf("Inserisci un numero intero\n");
            getchar();
        }
    } while (check != 1);

    if(b!=0){ // eseguo la divisione solo se il numero inseritonon è 0
        divisione = (float) a / b;
        printf ("La divisione tra %d e %d e': %f", a,b,divisione);
    }else{
        printf("Impossibile eseguire la divisione per 0! farebbe Infinito \n");
    }
}

void ins_string() {

```

```

void ins_string() {
    char stringa[10]; // stringa di 10 caratteri, si considera anche il terminatore di stringa
    int i; // uso ancora il check per controllare che l'utente abbia inserito una stringa
    bool valid = 0;
    bool check = 0;
    while (!valid) {
        printf("Inserisci la stringa (max 10 caratteri):");
        check = scanf(" %s", &stringa);
        // Rimuove il carattere di a capo alla fine della stringa, che può causare errori
        if (stringa[strlen(stringa) - 1] == '\n') {
            stringa[strlen(stringa) - 1] = '\0';
        }
        // Verifica se la stringa è composta solo da caratteri alfabetici
        valid = true;
        for (i = 0; i < strlen(stringa); i++) {
            if (!isalpha(stringa[i])) {
                valid = false;
                break;
            }
        }
        if(check != 1){█
            printf("Inserisci una stringa, i numeri non sono validi! ");
        }

        if (!valid) {
            printf("La stringa inserita non è valida! \n");
        }
    }
    printf("La stringa inserita è: %s\n", stringa);
}

```

Il programma è testato e compila, durante l'esecuzione cattura tutti gli errori che sono stati trovati durante la fase di analisi.