

Bug C-Hunting

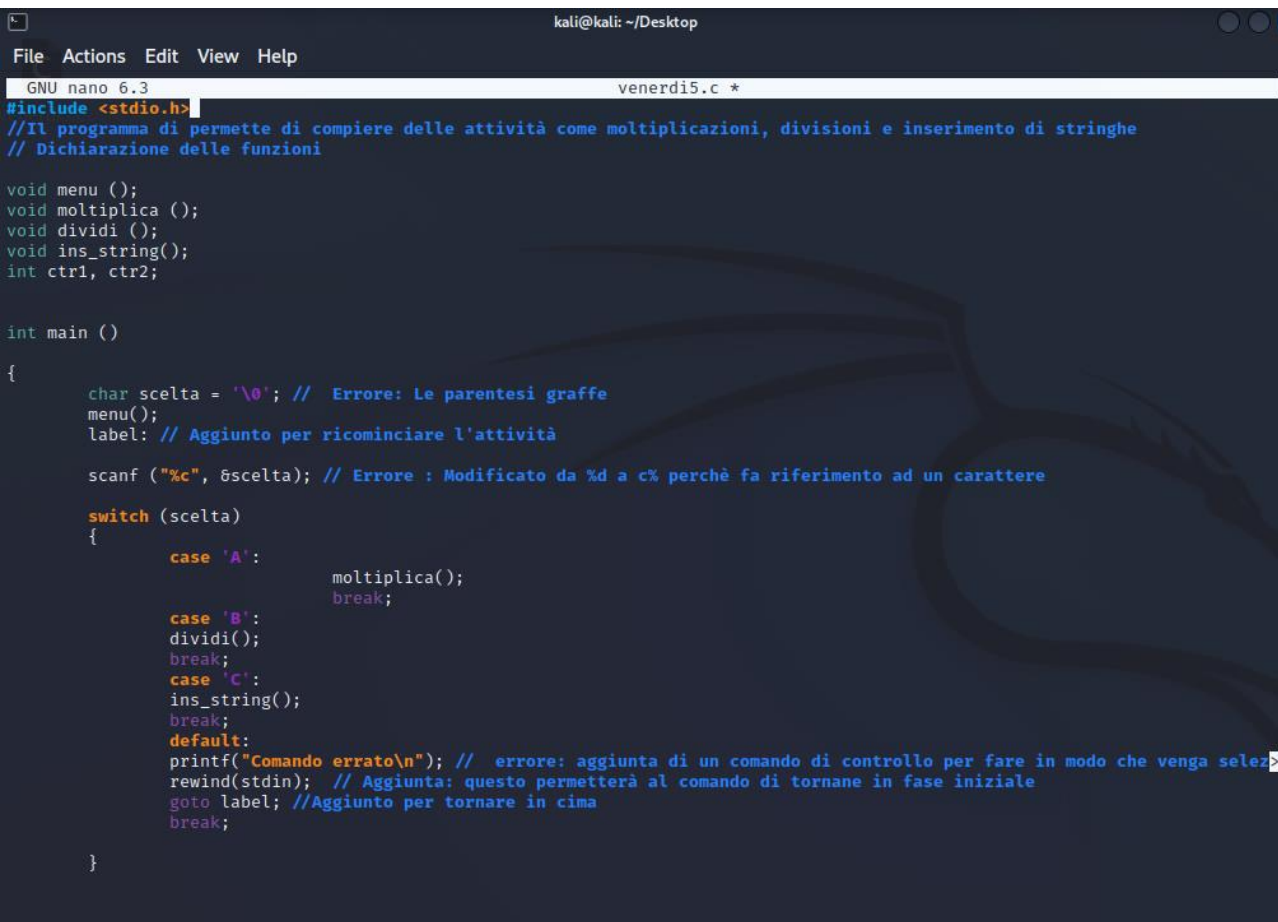
Questa applicazione ti dà la possibilità di compiere tre semplici scelte di cui:

- Moltiplicare
- Dividere
- Inserire una Stringa

Tramite le nostre richieste avremmo delle risposte corrette

Inizialmente si sono riscontrati diversi problemi riguardante l'inserimento dei dati (caratteri non ammessi e non segnalati, risposte alle funzioni incorrette) che sono state adeguatamente sistemate come qui sotto.

Qui sono stati trovati dei caratteri aggiunti "{}" che non servivano ed è stata aggiunta un'etichetta (**Label**) che funzionerà da richiamo (**Goto**) per ritornare all'interna. E' stato anche aggiunto un comando di controllo per l'inserimento di un carattere non richiesto



```
GNU nano 6.3 venerdì5.c *
#include <stdio.h>
//Il programma di permette di compiere delle attività come moltiplicazioni, divisioni e inserimento di stringhe
// Dichiarazione delle funzioni

void menu ();
void moltiplica ();
void dividi ();
void ins_string();
int ctr1, ctr2;

int main ()
{
    char scelta = '\0'; // Errore: Le parentesi graffe
    menu();
    label: // Aggiunto per ricominciare l'attività

    scanf ("%c", &scelta); // Errore : Modificato da %d a %c perchè fa riferimento ad un carattere

    switch (scelta)
    {
        case 'A':
            moltiplica();
            break;

        case 'B':
            dividi();
            break;
        case 'C':
            ins_string();
            break;
        default:
            printf("Comando errato\n"); // errore: aggiunta di un comando di controllo per fare in modo che venga selezionato
            rewind(stdin); // Aggiunta: questo permetterà al comando di tornare in fase iniziale
            goto label; //Aggiunto per tornare in cima
            break;
    }
}
```

Nel **Void Moltiplica** è stata resa più efficiente la risposta e sono stati sostituiti i valori **&d** per la corretta risposta, sono stati aggiunti anche **ctr1\ctr2** di richiamo (int ctr1, ctr2) così che venga bloccato qualsiasi carattere che non sia un numero, racchiudi nel **Do-While**

```

void moltiplica ()
{
    do{

        int a,b, prodotto; // Errore: in questa maniera le operazioni saranno più efficienti
        printf ("Inserisci i due numeri da moltiplicare:");
        ctr1 = scanf ("%d", &a); // 5° Errore: %f viene sostituito in %d
        ctr2 = scanf ("%d", &b); // Qui viene inserito un blocco nel caso si digiti un carattere e non un numero
        if(ctr1 == 0 || ctr2 == 0) {
            printf ("Solo numeri \n\n");
            rewind(stdin);
        }

        else{
            prodotto = a * b;
            printf ("Il prodotto tra %d e %d e': %d", a,b,prodotto);
            rewind(stdin);} // aggiunta: Ci permetterà di tornare allo stadio iniziale
    } while (ctr1 == 0 || ctr2 == 0);
}

```

Void Dividi;

Qui sono stati corretti i valori %d con %f per rendere la risposta corretta, con in precedenza è stato racchiuso tutto nel Do-While, in maniera da rimanere nel programma, ed è stata impletata anche qui la funziona per il corretto inserimento dei dati.

```

void dividi ()
{
    do{

        float a,b = 0;

        printf ("Inserisci il numeratore:");
        ctr1 = scanf ("%f", &a); //qui abbiamo corretto il fattore %d con il fattore %f
        printf ("Inserisci il denominatore:");
        ctr2 = scanf ("%f", &b); //qui anche la correzione di %d in %f
        if(ctr1 == 0 || ctr2 == 0) { // Blocco per la digitare correttamente un numero
            printf ("Solo numeri \n\n");
            rewind(stdin); //ci permetterà di tornare in cima
        }

        else{
            float divisione = a / b;
            printf ("La divisione tra %0.1f e %0.1f e': %0.4f", a,b,divisione);
            rewind(stdin);
        } }while (ctr1 == 0 || ctr2 == 0);
}

```

Nel **void ins_string** è stato aggiunto il segnale di immissioni dei caratteri richiesti.

```

void ins_string ()
{
    char stringa[10];
    printf ("Inserisci la stringa:");
    fgets (stringa,10, stdin);
    printf("Hai inserito la sequenza stringa: %s",stringa); // qui è il inserito il blocco dei caratteri
    rewind(stdin);
}

```