

Questo è il codice dato sbagliato nel progetto di oggi

```
#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 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;
    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);
}
```

Questa è la correzione come richiesta nell'Esercizio

Analizziamo il codice fornito :errore di dichiarazione di scelta: La dichiarazione della variabile scelta è incorretta. È dichiarata come un array di caratteri, ma dovrebbe essere una singola variabile di tipo char. **char scelta = '\0'; // Corretta dichiarazione** errore nell'Input della Scelta: nell'istruzione **scanf**, stai cercando di acquisire un carattere, quindi dovresti utilizzare **%c** anziché **%d**. **scanf(" %c", &scelta); // corretta acquisizione del carattere** errore nei parametri di scanf nella Funzione moltiplica: Nella funzione moltiplica, stai utilizzando il formato **%f** per acquisire un intero. Dovresti usare **%d** per acquisire un intero. **scanf("%d", &a); // Corretta acquisizione di un intero** Errore nella Funzione dividi: L'operazione di divisione dovrebbe essere eseguita con l'operatore **/**, non con **%**. Quindi, correggi l'operazione di divisione. **int divisione = a / b; // Corretta operazione di divisione** problema nella Funzione ins_string: La funzione scanf con **%s** acquisirà una stringa senza limiti, ma la dimensione dell'array stringa è di 10.. È consigliabile **utilizzare scanf("%9s", stringa);** per specificare la lunghezza massima della stringa acquisita. **scanf("%9s", stringa); // Corretta acquisizione della stringa** miglioramenti generali: per migliorare la leggibilità, puoi utilizzare formati più esplicativi per la stampa, come **"%d"** per gli interi anziché **"%d"**. **printf("Il prodotto tra %d e %d è: %d", a, b, prodotto);** mancata gestione di Input Non Validi: Non c'è alcuna gestione degli input non validi dell'utente. Potresti considerare l'aggiunta di controlli per garantire che l'utente inserisca input corretti.

Esercizio 2

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

#define MAX_NOME 50

int main() {
    int scelta, punteggio = 0;
    char nome[MAX_NOME];

    printf("Benvenuto nel gioco di domanda/risposta!\n");
    printf("Lo scopo del gioco è rispondere correttamente alle domande per ottenere il massimo punteggio.\n");

    do {
        printf("\nMenu di scelta:\n");
        printf("A) Iniziare una nuova partita\n");
        printf("B) Uscire dal gioco\n");
        printf("Scegli: ");
        scanf(" %c", &scelta);

        if (scelta == 'A' || scelta == 'a') {
            printf("Inserisci il tuo nome: ");
            // Utilizziamo fgets per leggere una linea intera, evitando buffer overflow
            fgets(nome, sizeof(nome), stdin);
            nome[strcspn(nome, "\n")] = '\0'; // Rimuove il carattere newline eventualmente presente

            // Controllo lunghezza nome
            if (strlen(nome) >= MAX_NOME) {
                printf("Nome troppo lungo. Riprova con un nome più breve.\n");
                continue;
            }
        }
    } while (scelta != 'B' && scelta != 'b');
```

```
punteggio = 0; // Resetta il punteggio per una nuova partita
```

```
// Domande
```

```
printf("\n%s, rispondi alle seguenti domande:\n", nome);
```

```
// Domanda 1
```

```
printf("1. Qual è la capitale dell'Italia?\n");
```

```
printf("A) Roma\nB) Parigi\nC) Berlino\nScelta: ");
```

```
char risposta1;
```

```
scanf(" %c", &risposta1);
```

```
if (risposta1 == 'A' || risposta1 == 'a') {
```

```
    punteggio++;
```

```
}
```

```
// Domanda 2
```

```
printf("\n2. Quale pianeta è più grande? \n");
```

```
printf("A) Marte\nB) Giove\nC) Venere\nScelta: ");
```

```
char risposta2;
```

```
scanf(" %c", &risposta2);
```

```
if (risposta2 == 'C' || risposta2 == 'c') {
```

```
    punteggio++;
```

```
}
```

```
// Domanda 3
```

```
printf("\n3. Quanti continenti ci sono nel mondo?\n");
```

```
printf("A) 5\nB) 6\nC) 7\nScelta: ");
```

```
char risposta3;
```

```
scanf(" %c", &risposta3);
```

```
if (risposta3 == 'C' || risposta3 == 'c') {
```

```
    punteggio++;
```

```
}
```

```
// Domanda 4
```

```
printf("\n4 In che anno fu scoperta l'America?\n");
```

```
printf("A) 1700\nB) 1600\nC) 1492\nScelta: ");
```

```
char risposta4;
```

```
scanf(" %c", &risposta4);
```

```
if (risposta4 == 'C' || risposta4 == 'c') {
```

```
    punteggio++;
```

```
}
```

```
// Domanda 5
```

```
printf("\n5 Chi ha inventato Tesla?\n");
```

```
printf("A) Gates\nB) Bezos\nC) Musk\nScelta: ");
```

```
char risposta5;
```

```
scanf(" %c", &risposta5);
```

```
if (risposta5 == 'C' || risposta5 == 'c') {
```

```
    punteggio++;
```

```
}
```

```
// Mostra il punteggio
```

```
printf("\n%s, hai totalizzato %d punti.\n", nome, punteggio);
```

```
} else if (scelta != 'B' && scelta != 'b') {
```

```
    printf("Scelta non valida. Riprova.\n");
```

```
printf("Scelta non valida. Riprova!\n");  
// Pulisce il buffer per evitare problemi con l'input successivo  
while (getchar() != '\n');  
}  
} while (scelta != 'B' && scelta != 'b');  
printf("Grazie per aver giocato!\n");  
return 0;  
}
```

In questo codice:

1. Abbiamo sostituito `scanf("%s", nome)` con `fgets(nome, sizeof(nome), stdin)` per leggere una linea intera, evitando così il buffer overflow.
2. Abbiamo rimosso il carattere newline che `fgets` potrebbe includere alla fine della stringa.
3. Aggiunto un controllo sulla lunghezza del nome.
4. Aggiunto un loop per pulire il buffer dell'input in caso di scelta non valida, evitando così problemi di input successivi.