

Progetto S2L5 Luca Danelli

Il programma propone all'utente una scelta tra tre diverse casistiche, denominate A,B e C, che rispettivamente corrispondono a

A: Moltiplicare due numeri

B: Dividere due numeri

C: Inserire una stringa

Successivamente all'interno del programma vengono richiamate 3 funzioni, ossia `moltiplica()`, `dividi()` e `ins_string()` codificate per il lavoro preposto. Viene usata l'istruzione **switch()** per permettere all'utente di scegliere quale funzione utilizzare.

Il primo errore è nella lettura della variabile **scelta**, che sarà poi il parametro dell'istruzione switch:

codice esercizio:

```
scanf ("%d", &scelta);
```

il codice corretto:

```
scanf ("%s", &scelta);
```

In quanto scelta è una stringa, non un intero.

Voglio poi controllare che l'utente possa scegliere solo tra A,B o C. Lo posso fare modificando il codice nel modo seguente, informando anche l'utente dell'errore:

codice esercizio:

```
int main ()
```

```
{
```

```
    char scelta = {'\0'};
```

```
    menu ();
```

```
    scanf ("%d", &scelta);
```

```
    switch (scelta)
```

codice corretto:

```
int main ()
```

```
{
```

```
    char scelta = {'\0'};
```

```
    menu ();
```

```
    do {
```

```
        scanf ("%s", &scelta);
```

```
        if ((scelta!='A')&&(scelta!='B')&&(scelta!='C')) {
```

```
            printf("scegli tra A, B o C\n");
```

```
        }
```

```
    }
```

```
    while ((scelta!='A')&&(scelta!='B')&&(scelta!='C')); // se la scelta non è fra quelle proposte  
    il ciclo verrà ripetuto a partire dal do
```

Errore simile al primo nella funzione `moltiplica()`, in particolare nella lettura della variabile `a`.

Inoltre dovremmo dichiarare le variabili come `int` per leggerle con `%d` nello `scanf`:

codice esercizio:

```
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);
}

```

il codice corretto:

```

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

    int prodotto = a * b;

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

```

Poi, nella funzione dividi():

codice esercizio:

```

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);
}

```

Qui ho trovato più di un errore. Innanzi tutto nella divisione l'operatore dovrebbe essere / e non %, perché quest'ultimo serve a restituire il resto di una divisione intera. Poi, le variabili a,b e divisione dovrebbe essere definita come **float** in quanto ci aspettiamo che dividendo due interi possiamo ottenere anche risultati decimali, ed operando fra due interi non possiamo aspettarci un risultato intero. Possiamo poi usare la notazione %.2f per stampare a schermo solo i primi due decimali dopo la virgola nel printf() finale. Inoltre non viene fatto un controllo nel caso l'utente inserisca 0 come denominatore. Per risolvere l'ultimo punto ho usato un ciclo **do while** che ripete la lettura da tastiera di b da parte dell'utente se questo inserisce 0 come denominatore. Ho poi aggiunto un printf all'interno dell'istruzione if per informare l'utente dell'errore

codice corretto:

```

void dividi ()
{
    float a,b = 0;
    printf ("Inserisci il numeratore:");

```

```

scanf ("%f", &a);
printf ("Inserisci il denominatore:");
do {
scanf ("%f", &b);
    if (b==0) {
        printf("il denominatore non può essere zero!\ninseriscilo di nuovo:");
    }
}
while (b==0); //se b è uguale a 0 il ciclo riprende dal do
float divisione = a / b;

printf ("La divisione tra %.2f e %.2f e': %.2f", a,b,divisione);
}

```

Veniamo alla funzione `ins_string()`.

codice esercizio:

```

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

```

Per prima cosa inizializzo a zero il valore della variabile stringa. C'è poi un errore nella lettura della stringa con `scanf`, va rimosso il `&`. Sarebbe poi opportuno fare un controllo sulla lunghezza della stringa inserita dall'utente in modo da evitare problemi di overflow. Lo posso fare utilizzando la funzione **`strlen()`**, che restituisce la lunghezza di una stringa, avendo cura di importare la libreria `<string.h>` in cima al codice. Inoltre, ho aggiunto un `printf()` per informare l'utente sulla stringa inserita stampandola a schermo:

codice corretto:

```

void ins_string ()
{
    char stringa[10]={'\0'};
    do {
        printf ("Inserisci la stringa (MAX 10 caratteri):");
        scanf ("%s", stringa);
        if (strlen(stringa) > 10) {
            printf("inserisci massimo 10 caratteri!\n");
        }
    }
    while (strlen(stringa) > 10); // se la lunghezza è maggiore di 10 il ciclo viene ripetuto
    printf("la stringa che hai inserito: %s", stringa);
}

```