

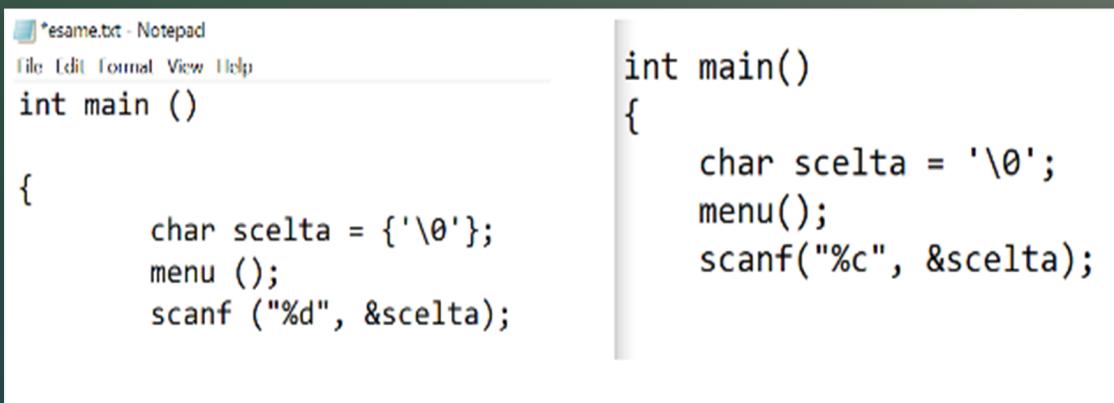


Bug Hunting
Francesco Emulo

Relazione

Durante la lettura del codice fornito dalla traccia appare abbastanza evidente che si tratti di un'assistente digitale che permette all'utente, tramite un piccolo menù a risposta multipla, di effettuare una divisione tra due numeri, una moltiplicazione tra due numeri, o inserire una stringa. Analizzando il codice, tuttavia, sono stati riscontrati diversi errori; per rendere la lettura del report più leggera, verranno allegati a sinistra (o in alto) gli screen del codice pre-correzione ed a destra (o in basso) quelli inerente il codice corretto (seguiti da un commento volto a giustificare le scelte intraprese).

Errori Di Sintassi e Logici



The screenshot shows two versions of a C program side-by-side in a Notepad window. The left column contains the original code with several syntax errors, while the right column contains the corrected code.

Original Code (Left):

```
int main ()
{
    char scelta = {'\0'};
    menu ();
    scanf ("%d", &scelta);
```

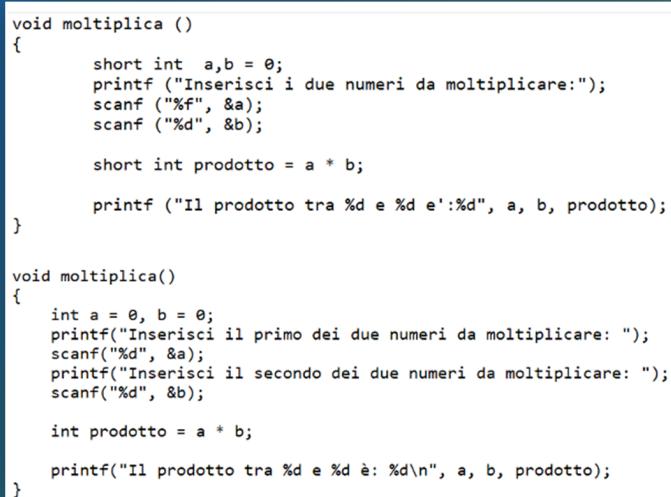
Corrected Code (Right):

```
int main()
{
    char scelta = '\0';
    menu();
    scanf("%c", &scelta);
```

Partendo dalla funzione main, lo scanf della variabile scelta deve essere %c dato che la variabile sarà di tipo char (carattere). %d sarebbe stato opportuno nel caso in cui avessimo trattato numeri interi.
NB: Lo switch è trattato nelle implementazioni.

Moltiplica

Passando alla funzione “moltiplica”, invece, possiamo notare come siano stati utilizzati una volta %f, (notazione riservata alle variabili float), ed una volta %d (riferito alle variabili int).



The screenshot shows a C program with two implementations of the 'moltiplica' function. The first implementation uses %f for both inputs, while the second uses %d.

```
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 è:%d", a, b, prodotto);
}

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

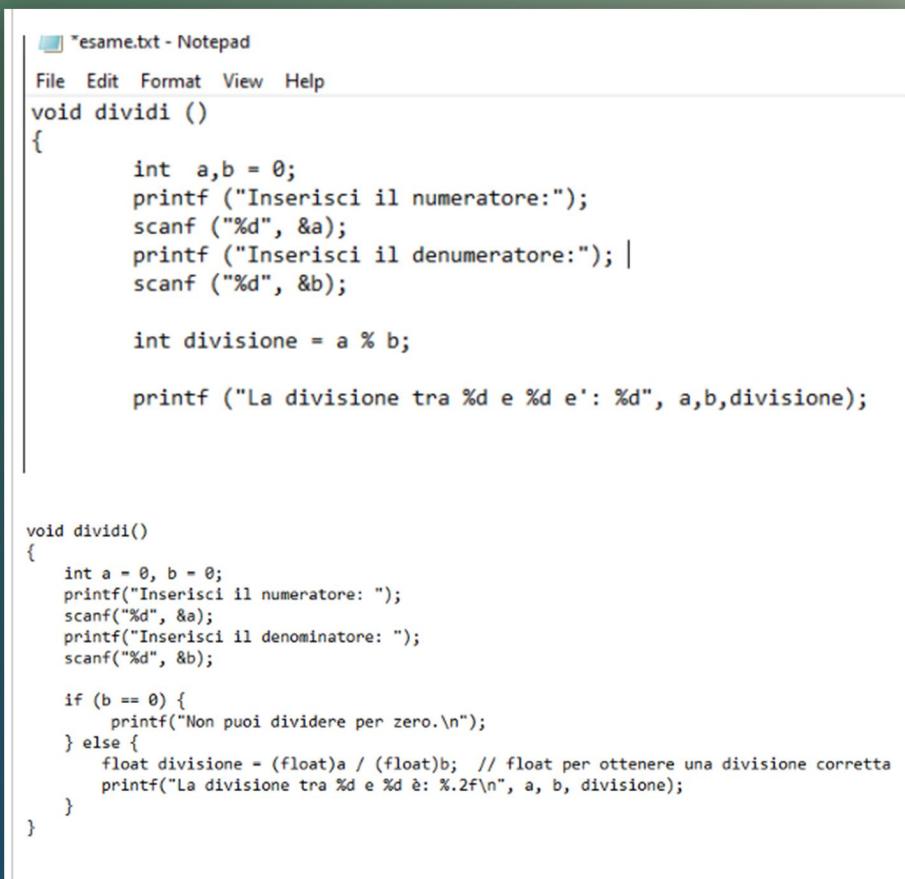
    int prodotto = a * b;

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

Inoltre, era stata utilizzata la funzione short int; questa poteva essere lasciata così com'era dato che non è un'errore (magari, in questo caso, si potevano sostituire lo specificatore di formato %d con %hd); tuttavia short int è un dato rappresentante un intero di dimensioni generalmente inferiori ad int (le sue dimensioni, infatti, sono di 2byte rispetto ad i 4byte di int)

Dividi

Arriviamo alla funzione “dividi”, purtroppo anche qui sono stati riscontrati errori logici. Analizziamo gli screen:



```
*esame.txt - Notepad
File Edit Format View Help
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 è: %d", a,b,divisione);
}

void dividi()
{
    int a = 0, b = 0;
    printf("Inserisci il numeratore: ");
    scanf("%d", &a);
    printf("Inserisci il denominatore: ");
    scanf("%d", &b);

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

Seguendo un flusso crano-caudale, il primo errore che possiamo notare è sito nella stringa del secondo printf. L'autore ha richiesto di inserire il denominatore, doveva, invece, richiedere il denominatore. Ancora è stato modificato l'operatore aritmetico % (utilizzato per ottenere il resto di una divisione) con l'operatore aritmetico / (utilizzato per ottenere il rapporto tra due numeri). Ulteriore accorgimento è stato quello di modificare la variabile divisione da int a float poichè non è detto che non vi sia resto.

NB: Il condizionale è trattato nelle implementazioni.

Ins_string

```
File Edit Format View Help
void ins_string ()
{
    char stringa[10];
    printf ("Inserisci la stringa:");
    scanf ("%s", &stringa);
}
void ins_string()
{
    char stringa[10];
    printf("Inserisci la stringa: ");
    scanf("%s", stringa);
    printf("Hai inserito: %s\n", stringa);
}
```

Generalmente, l'operatore & (operatore di indirizzo o di riferimento) viene utilizzato per ottenere l'indirizzo di memoria di una variabile. In questo caso, tuttavia, se consideriamo lo scanf della prima immagine, è superfluo tale operatore dato che "stringa" è un'array. A seguire lo scanf è stato implementato un printf per stampare l'output della stringa precedente.

Menù

Errore di battitura al primo printf: è stato scritto assistente invece di assistente.

Implementazioni al programma

La principale implementazione riguarda lo Switch sito in int main()

```
switch (scelta)
{
    case 'A':
        moltiplica();
        break;
    case 'B':
        dividi();
        break;
    case 'C':
        ins_string();
        break;
}
switch (scelta)
{
    case 'A':
        moltiplica();
        break;
    case 'B':
        dividi();
        break;
    case 'C':
        ins_string();
        break;
    default:
        printf("Inserisci una delle
               scelte possibili.\n");
        break;
}
```

E' stato fatto in modo che quando l'utente sbaglia ad inserire l'input richiesto, il programma lo segnala stampandolo a schermo. Ulteriori implementazioni sono commentate nel codice.

Tornando alla funzione dividi, trattiamo il condizionale implementato:

```
*esame.txt - Notepad
File Edit Format View Help
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 è: %d", a,b,divisione);

}

void dividi()
{
    int a = 0, b = 0;
    printf("Inserisci il numeratore: ");
    scanf("%d", &a);
    printf("Inserisci il denominatore: ");
    scanf("%d", &b);

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

Abbiamo fatto sì da distinguere due casistiche possibili:

- L'utente inserisce un qualsiasi numero diverso da 0;
- L'utente inserisce 0.

Quando il denominatore è 0 il programma stamperà che tale denominatore non è accettabile evitando di incorrere in errori di runtime.

Ulteriori errori ed implementazioni sono state descritte nel codice.