

Dato il codice in allegato, si richiede allo studente di:

1. Capire cosa fa il programma senza eseguirlo.
2. Individuare dal codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati).
3. Individuare eventuali errori di sintassi / logici.
4. Proporre una soluzione per ognuno di essi.

Capire cosa fa il programma senza eseguirlo.

Ad una prima vista del codice si intuisce che lo scopo del programma sia quello di un possibile assistente digitale che sa svolgere 3 compiti: moltiplicare due numeri, dividere due numeri oppure inserire una stringa.

Si nota anche che il programma ha un menù iniziale dove, premendo corrispettivamente A, B o C si potrà scegliere il tipo di compito da fargli svolgere.

Individuare dal codice sorgente le casistiche non standard che il programma non gestisce

Si può notare subito che il programma non gestisce una possibile casistica che gli si venga inserita una lettera diversa da A, B e C questo fa sì che se io inserissi una qualsiasi altra lettera dell'alfabeto, il programma non fa niente.

Un altro caso è che il programma permette all'utente di dividere per 0 quando questa operazione è impossibile.

Ultimo problema visibile è che il codice permette all'utente di inserire una stringa. Quest'ultima registra solo dieci caratteri, quindi potrebbe andrebbe in errore se l'utente dovesse inserirne più di 10. In oltre il programma carica solo la stringa ma non stampa niente all'utente.

Individuare eventuali errori di sintassi / logici

Errori di sintassi e logici se ne notano parecchi, primo fra tutti è:

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

Io lo cambierei semplicemente in

```
char scelta = '0'
```

per far sì che il nostro ciclo do-while funzioni (lo vedremo più avanti)

Il prossimo errore che possiamo notare è:

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

Infatti, scelta è un carattere char (come 'A' e 'B') ma è stato scritto in formato %d che serve per i numeri interi quindi la modifica da fare sarebbe sostituire %d con %c

```
scanf ("%c", &scelta)
```

Altro errore lo notiamo sulle righe di codice

```
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)
```

Secondo me la cosa migliore da fare su questa parte di codice e usare int al posto di short int. Quindi sostituire int al posto di short int e poi di modificare il formato di "a" "b" e "prodotto" %d

```
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)
```

Andando avanti vediamo che in “dividi” la divisione è:

```
int divisione = a % b
```

L'errore qui è molto chiaro: è stato inserito il % al posto del / per dividere i 2 numeri

```
int divisione = a / b
```

Ulteriore modifica sarebbe quello di mettere tutto in float, visto che parliamo di divisioni e, visto che spesso portano risultati con la virgola, cambiare da int a float e sostituire i %d con %f. In tal caso avremmo una situazione del genere

```
float a,b = 0;
```

```
printf ("Inserisci il numeratore: ");
```

```
scanf("%f", &a);
```

```
printf ("Inserisci il denominatore: ");
```

```
scanf("%f", &b);
```

```
if (b == 0) {
```

```
    printf("È impossibile dividere per zero\n");
```

```
    return;
```

```
}
```

```
float divisione = a / b;
```

```
printf ("La divisione tra %0.2f e %0.2f è:  
%0.2f\n", a, b, divisione);
```

Per evitare di avere parecchi numeri dopo la virgola si potrebbe usare %0.2f come sopra

Proporre una soluzione per ognuno di essi

In allegato ho riscritto il programma con le modifiche sopra elencate e con qualche aggiunta che vado a spiegare.

Prima di tutto ho inserito un do-while nel menù così da poter risolvere il problema che se l'utente dovesse inserire un carattere diverso da "A, B o C" il programma lo rimanderebbe al menù iniziale chiedendogli se vuole fare un'operazione.

Altra aggiunta è aver aggiunto un if nella sezione "dividi" per far sì che se viene inserito come denominatore il numero 0 l'operazione è impossibile.

Nella ultima sezione di stringa ho aumentato la lunghezza della stringa possibile a 50 caratteri e ho inserito un printf per mostrare la stringa inserita.