

Il test di oggi consiste in:

1. Analizzare e comprendere cosa fa il programma dato in allegato senza eseguirlo
2. Individuare dal codice casistiche che il programma non gestisce
3. Individuare eventuali errori
4. Ricollegandoci al punto 3, se ce ne sono risolverli

1.

Iniziamo dall'intestazione del nostro programma

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

```
void menu ();
void moltiplica ();
void dividi ();
void ins_string();
```

Ogni programma in C per poter funzionare ha bisogno dell'inclusione delle librerie, come in questo caso con la libreria "stdio.h" che sta per "standard input-output header" con la quale si includono le definizioni standard per poter far funzionare il programma. Sotto possiamo notare la dichiarazione delle quattro diverse funzioni "void" le quali avranno ognuna la propria mansione che vedremo più avanti. Il tipo di funzione "void" è una funzione alla quale non interessa un valore di ritorno.

```
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");
}
```

Questa funzione una volta richiamata dal programma avrà il compito di stampare l'intestazione appunto della nostra applicazione con la funzione "printf"

```
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 funzione al suo interno ha le istruzioni per far sì che una volta richiamata dal programma possa svolgere la moltiplicazione tra due numeri

```
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);
}
```

Presenta analoghe istruzioni della funzione moltiplica ma questa una volta richiamata farà la divisione tra due numeri

```

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

```

Questa funzione se richiamata farà inserire una stringa di massimo 10 caratteri dall'utente

Infine tutte queste funzioni per funzionare devono essere richiamate nella funzione principale del programma, senza la quale il programma non compilerebbe neanche, ossia la funzione "main"

```

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

    switch (scelta)
    {
        case 'A':
            multiplica();
            break;
        case 'B':
            dividi();
            break;
        case 'C':
            ins_string();
            break;
    }

    return 0;
}

```

La funzione "main" a differenza delle altre è di tipo "int", questo vuol dire che alla fine del programma ci dovrà essere un valore di ritorno di tipo "int", come in questo caso "return 0".

Lo scopo di questa funzione sarà quello di stampare per prima cosa un menù da cui scegliamo cosa fare attraverso la funzione "menu()", una volta selezionata l'azione desiderata con lo "scanf" in base all'azione che abbiamo scelto verranno richiamate le varie funzioni "multiplica()", "dividi()" o "ins_string()" che eseguiranno le istruzioni che gli sono state date. Una volta terminate le istruzioni della funzione il programma uscirà da solo e si chiude l'applicazione

2.

Analizzando il codice possiamo notare che le migliorie da poter apportare sono:

- Fare un "case D" nello switch dove c'è l'opzione di uscire subito dal programma
- Controlli sullo switch affinché se l'utente sbaglia a digitare non esce dal programma ma compare un messaggio dove lo invita a digitare i valori consentiti e solo dopo può partire con l'operazione desiderata
- Nella funzione dividi() cambiare il tipo di dato da int a float così da poter fare anche le divisioni con il resto
- Sempre nella funzione dividi() mettiamo un controllo per evitare che venga messo 0 come denominatore
- Aggiungere un printf alla fine della funzione ins_string() per far vedere all'utente la stringa che ha inserito
- Una volta finito con un operazione si fa decidere all'utente se serve altro o meno. In caso voglia fare un'altra operazione il programma ripartirà, altrimenti uscirà

3.

```

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

    switch (scelta)
    {
        case 'A':
            multiplica();
            break;
        case 'B':
            dividi();
            break;
        case 'C':
            ins_string();
            break;
    }

    return 0;
}

```

L'errore principale di natura logica che troviamo all'interno del main è il "%d" utilizzato nello scanf. Avendo assegnato il tipo "char" alla variabile "scelta" invece di "%d" che si usa per i tipi di dati "int", bisognerà mettere il "%c" che si usa per i tipi di dati "char". È un errore logico perché in fase di compilazione non darà errore ma solo warning, ma lanciando la nostra applicazione e andando ad inserire un valore char, il valore non viene riconosciuto e l'applicazione terminerà. Un altro errore di natura estetica per avere un codice più pulito, leggibile e omogeneo è la mancata indentazione nel "case A" dello switch.

```
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");
}
```

Qui c'è un errore di sintassi, un errore ortografico al momento della scrittura

```
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);
}
```

Qui ci sono due errori logici nei printf poiché in entrambi manca "\n" per chiudere la riga ed andare a capo. Dico errore logico perché ci accorgiamo solo nel momento di esecuzione del codice che scriverà tutto sulla stessa riga. Altro errore è sul primo scanf poiché per prendere in input un numero bisogna mettere "%d" e non "%f". Altro errore è mettere due tipi di dati per la stessa variabile: o è "int" o è "short"

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

    int divisione = a % b;

    printf ("La divisione tra %d e %d e': %d", a,b,divisione);
}
```

Anche nella funzione dividi() come nella funzione moltiplica() mancano gli "\n" di fine riga nel printf, e inoltre il simbolo per fare la divisione non è questo "%" ma questo "/".

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

Manca il "\n" alla fine del printf

4.

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

Aggiunto un printf a cui passiamo la stringa scritta dall'utente cosicché quest'ultimo possa controllare cosa ha scritto

```

int main ()
{
    char scelta = {'\0'};
    char sceltafinale = {'\0'};

    do{
        menu ();
        scanf ("%c", &scelta);

        while(scelta != 'A' && scelta != 'B' && scelta != 'C' && scelta != 'D'){
            printf("Inserisci un carattere valido\n");
            scanf ("%c", &scelta);
        }

        switch (scelta)
        {
            case 'A':
                multiplica();
                break;
            case 'B':
                dividi();
                break;
            case 'C':
                ins_string();
                break;
            case 'D':
                printf("Sei uscito dal programma\n");
                break;
        }

        printf("Ti serve altro?\nS >> Si\nN >> No\n");
        scanf ("%c", &sceltafinale);

        while(sceltafinale != 'S' && sceltafinale != 'N'){
            printf("Inserisci un carattere valido\n");
            scanf ("%c", &sceltafinale);
        }
    } while(sceltafinale == 'S');

    return 0;
}

```

Le nuove modifiche apportate al main, compresi di controlli per digitare gli input giusti e “case D” per uscire subito dal programma

```

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\nD >> Esci dal programma\n");
}

```

Funzione menù rimasta quasi uguale tranne per qualche correzione ortografica e l’aggiunta dell’opzione D

```

void multiplica ()
{
    int a,b = 0;
    printf ("Inserisci il primo numero da moltiplicare:\n");
    scanf ("%d", &a);
    printf ("Inserisci il secondo numero da moltiplicare:\n");
    scanf ("%d", &b);

    int prodotto = a * b;
    printf ("Il prodotto tra %d e %d e': %d\n", a,b,prodotto);
}

```

Corretto il “%f” con il “%d” per acquisire i valori int che vengono passati dallo scanf

```

void dividi ()
{
    float a,b = 0;
    printf ("Inserisci il numeratore:\n");
    scanf ("%f", &a);
    printf ("Inserisci il denominatore:\n");
    scanf ("%f", &b);
    while(b == 0){
        printf ("Inserisci il denominatore maggiore di 0:\n");
        scanf ("%f", &b);
    }

    float divisione = a / b;
    printf ("La divisione tra %f e %f e': %f\n", a,b,divisione);
}

```

Aggiunto il controllo sul denominatore maggiore di zero, e cambiato il tipo di dato da int a float per fare le divisioni con il resto

