

Report seconda settimana

Questo programma permette all'utente di scegliere tra 3 possibili operazioni, moltiplicazione di due numeri, divisione e inserimento e lettura di una stringa.

Di seguito ho commentato gli errori e le casistiche non contemplate in ogni funzione:

```
int main ()
{
    char scelta = {'\0'};           //parentesi graffe inutili
    menu ();
    scanf ("%d", &scelta);          //variabile di tipo char quindi %c al posto di %d
    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");
}
```

//l'utente potrebbe inserire caratteri diversi quindi bisogna effettuare un controllo, per esempio attraverso un ciclo do while o con la funzione fgets

```
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);
}
```

//le variabili a , b e prodotto sono dichiarate come short int e lette come float e int quindi bisogna modificare %f e %d con %hd, inoltre l'utente potrebbe inserire numeri reali o troppo grandi quindi la soluzione migliore sarebbe modificare tutto in float e %f

```

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);
}

```

//l'operazione scritta in questo modo calcola il resto della divisione, cambiare % con / e sarebbe opportuno modificare la variabile in float per gestire i numeri reali inoltre bisogna effettuare un controllo nel caso l'utente inserisca 0 come denominatore

```

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

```

//le stringhe non necessitano dell'operatore &, bisogna inserire un controllo per accertarsi che l'utente non inserisca una stringa troppo lunga e getchar per il carattere newline

Il codice con le opportune correzioni

```

int main ()
{
    char scelta = '\0';           //parentesi graffe rimosse
    do{                            //aggiunto ciclo do while per controllo scelta utente
        menu ();
        scanf ("%c", &scelta);    //modificato il %d in %c
    }while (scelta != 'A' && scelta != 'B' && scelta != 'C');
    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");
}

```

//ciclo while aggiunto in main ()

```

void moltiplica ()
{
    float a, b = 0;
    printf ("Inserisci i due numeri da moltiplicare:");
    scanf ("%f", &a);             scanf ("%f", &b);
    float prodotto = a * b;
    printf ("Il prodotto tra %f e %f e': %f", a,b,prodotto);
}
//modificati i tipi di variabile in float e %f per la lettura

```

```

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

```

```
scanf ("%f", &b);
} while ( b == 0);

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

//modificati i tipi di variabile e introdotto un ciclo while per impedire l'inserimento dello 0 come denominatore

```
void ins_string ()
{
    char stringa[10];
    printf ("Inserisci la stringa:");
    scanf ("%s", stringa);
    getchar();
    while(strlen(stringa)>10){
        printf("stringa troppo lunga, inserirne una piu corta\n");
        printf("Inserisci la stringa: ");
        scanf("%s", stringa);
        getchar();
    }
}
```

//rimosso l'operatore & , introdotto un ciclo per il controllo della lunghezza della stringa e aggiunto getchar per il carattere newline

