Pratica S2-L5

#include <stdio.h>

void menu ();

void moltiplica ();

void dividi ();

void ins\_string();

int main ()

{

char scelta = {'\0'};

menu ();

scanf ("%d", &scelta);

switch (scelta)

{

case 'A':

moltiplica();

break;

case 'B':

dividi();

break;

case 'C':

ins\_string();

break;

}

return 0;

}

void menu ()

{

printf ("Benvenuto, sono un assitente 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");

}

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);

}

void dividi ()

{

int a,b = 0;

printf ("Inserisci il numeratore:");

scanf ("%d", &a);

printf ("Inserisci il denumeratore:");

scanf ("%d", &b);

int divisione = a % b;

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

}

void ins\_string ()

{

char stringa[10];

printf ("Inserisci la stringa:");

scanf ("%s", &stringa);

}

Il programma fornito sembra essere un semplice programma console interattivo che funge da calcolatrice di base con funzionalità aggiuntive.

1-Lo scanf("%s", &stringa); nella funzione ins\_string() può portare a un overflow del buffer se l'utente immette una stringa più lunga di 9 caratteri (la dimensione dell'array stringa). Una soluzione per non causare un overflow sarebbe %9s per limitare il numero di caratteri letti.

Possiamo anche svuotare il buffer di input per evitare eventuali problemi con l'input successivo.

while (getchar() != '\n');

2 - L'operazione di divisione nella funzione dividi() dovrebbe utilizzare / anziché %. L'operatore % in C è per il resto, non per la divisione.

3 - L'array stringa non viene inizializzato prima della lettura dell'input. Dovresti inizializzarlo prima di usarlo. char stringa[10] = {'\0'};

4 - Le variabili a e b nella funzione moltiplica() sono dichiarate come short int ma vengono scansionate utilizzando rispettivamente %f e %d. un opzione è %f in %hd per la variabile a. Siccome è necessario scrivere solo due byte (sulla maggior parte delle macchine) invece dei 4 byte scritti da "%d"

5 - Dopo aver completato un'operazione, il programma termina. Potrebbe essere utile mettere il menu e l'elaborazione dell'input in un ciclo per consentire all'utente di eseguire più operazioni senza riavviare il programma.(se voluto)

while (1) {

menu();

scanf(" %c", &scelta);

switch (scelta) {

………………..

}

}