Program for finding Difference using structures and functions

Assignment no.2

Submitted by- Group-15

Guided by: MD Meraz Sir

Professor: Dr. Mohammed Javed

Table of Contents

PROBLEM STATEMENT

Write a C program to find difference between decimal numbers by structures to a function.

HOW WE HAVE SOLVED THE PROBLEM

To rightfully obtain the difference between two decimal numbers for a stipulated number of times, we have used the concept of structures, arrays and functions. We have taken two approaches, namely, ‘the standard method’, that includes the usage of structures and functions, and the second one is an amalgamation of ‘structure with array’ and functions. Also, the output has been curated not to give the difference of the numbers as positive or negative values, rather to give out the absolute values.

Program 1 (Standard Method)

1. Initially, we make a struct ‘data’, which will store two floating values ‘a’ and ‘b’. We also make a ‘dif’ function which takes ‘struct data’ as an argument and returns the difference of two decimal numbers.
2. Inside the main function, we declare struct data ‘t’, and stores the two decimal numbers given by the user inside it.
3. Then, we call the ‘dif’ function using ‘t’ as an argument and the difference is printed on the screen.
4. Then, we ask the user whether the process need to be carried out again or not. Following are the three cases that arise, which we have managed using ‘if else-if’ ladder -

- If the user says ‘Yes’, then the program again ask the user to input two decimal numbers and carries out the difference process and then the goto statement transfers the control to the beginning, at the position where label ‘rp’ is defined and again the system asks ‘Do you want more’. The same process repeats again and again until the user says ‘No’.
- If the user says ‘No’, then, the program exits.
- For any other input, goto statement executes again and asks the user ‘Do you want more’.

Program 2 (Using structure with array)

1. Initially, we make a struct ‘data’, which will store three floating values ‘a’, ‘b’ and ‘c’. Also, declare ‘input’ function which will store decimal numbers inside the struct data ‘t’. Also, declare the ‘dif’ function which carries out the process of finding the difference between two decimal numbers.
2. Inside the main function, we declare the ‘struct data t[25]’.
3. Then, we ask the user how many number of times the difference has to be carried out and stores it in variable ‘n’.
4. Then, we execute ‘for’ loop from i=0 to i<n, this loop stores the decimal numbers in ‘struct data t’ array with the help of ‘input’ function and carries out the process of difference between two decimal numbers with the help of ‘dif’ function.
   t.c = t.b - t.a (if b>a) t.c = t.a - t.b (if a>b)
5. In this way, t.c stores the difference between two decimal numbers.

t.a	t.b	t.c
2.4564	1.2658	1.1906
32.4568	84.3541	52.8973

CODE FOR THE PROGRAM:

(The Standard Method)

```c
#include <stdio.h>

// STRUCTURE
struct data
{
    float a, b;
};

// FUNCTION- 'dif'
float dif(struct data i)
{
    float c;
    c = i.a - i.b;
    if (c >= 0)
        return c;
    else
        return -c;
};

// MAIN FUNCTION
int main()
{
    struct data t;
    char c;
    printf("Enter two decimal numbers (upto 4 decimal places)-\n");
    scanf("%f%f", &t.a, &t.b);
    printf("Difference=%.4f\n", dif(t));
```

```c
rp:
    printf("\nDo you want more (Y/N): ");
    fflush(stdin);
    c = getchar();

    if (c == 'Y' || c == 'y')
    {
        printf("Enter two decimal numbers (upto 4 decimal places)-\n");
        scanf("%f%f", &t.a, &t.b);
        printf("Difference=%.4f\n", dif(t));
        goto rp;
    }
    else if (c == 'N' || c == 'n')
        printf("\nFinish.");
    else
    {
        printf("Wrong input\n");
        goto rp;
    }
    return 0;
}
```

TIME COMPLEXITY

The first program does not involve any loop or recursive statements, therefore, the time complexity for the standard method will be **O(1)**.

The second program contains a single looping statement, namely, the ‘for’ loop, therefore, its time complexity will be **O(n)**.

O(1) < O(n) (Running time of the program)

CONCLUSION:

Therefore, both of our approaches for the program solves the problem statement. But, clearly, the standard method is more efficient than the method that uses structure with arrays. From the user's point of view, the only difference is that the first program asks the user every time whether they want to run it again or not. On the other hand, the second program initially asks the user about the number of times the difference needs to be obtained so that the user need not tell the system again and again to get the difference of various sets of two decimal numbers.

THANK YOU

GROUP-15

(IEC2021058, IEC2021059, IEC2021061, IEC2021062)