

The difference using structures and functions in C

Assignment 2

GROUP - 15 (IEC2021058, IEC2021059, IEC2021061, IEC2021062)

Department of Electronics and Communication Engineering

Indian Institute of Information Technology, Allahabad

Abstract— Given two decimal numbers, find the difference between them by passing structures to a function, for a certain number of times.

I. INTRODUCTION

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. To begin with, we take the required inputs from the user. The further program can be expressed by two approaches, the first one includes the usage of 'structures' and 'functions', and the second one is an amalgamation of 'structures with arrays' and 'functions'. Either approach would give us the difference between a set of two decimal numbers for a certain number of times, based on the user input. 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.

Definition of concepts used in the program:

- A **structure** is a user-defined data type, and creates a composite data type that can be used to group items of possibly different types into a single type under one name in a block of memory.
- **Arrays** are a collection of similar data elements stored at adjacent memory locations, wherein each data element can be accessed directly by only using its index number.
- **Functions** are a set of programming statements enclosed by {}, which can be called multiple times to provide reusability and modularity to the program.
- **Goto** statement is a jump statement, which is sometimes also referred to as an unconditional jump statement.

II. PROGRAM DESIGN

Program-1 (Standard Method)

- 1) Initially, we make a structure 'data' which will store two floating values 'a' and 'b'.
- 2) Then, we make the 'dif' function of the return data type- 'float'. This function takes the above structure as an argument and carries out the task of finding the difference between two decimal numbers. And, returns the answer.

- 3) Inside the main function, initially, we declare the structure 'data' as 't'.
- 4) Then, the program asks the user to enter two decimal numbers up to four decimal places that will be stored in 't'.
- 5) Then, the dif(t) function executes and the answer so calculated is printed on the screen.
- 6) Then, we ask the user whether the same task needs to be executed again or not.
- 7) If the user types 'y' or 'Y' for yes, the program executes again and the whole process repeats. To end the program the user can type 'n' or 'N'. If any other character is typed, it displays 'Wrong input' on the screen and again asks the user to type a character.
- 8) To carry out the above process, we have used the 'if else-if' ladder and the 'goto' statement. We have declared the 'goto' statement with the label name 'rp'.

Program-2 (Using structure with array)

- 1) Initially, we make a structure 'data', which can store three floating values 'a', 'b' and 'c'.
- 2) Then, we make the input() function of the 'struct data' type. This function takes the input of decimal numbers from the user and stores it in the element of an array of 'struct data' type.
- 3) Then, we make the dif() function of return data type- void. This function carries out the task of the difference of two decimal numbers and prints them. Inside the main function, initially, we declare two integer type variables 'i' and 'n'.
- 4) Then, we declare the array 't' of the 'struct data' type.

```
struct data t[25];
```
- 5) Now, we ask the user how many times the difference has to be carried out and store the input in a variable 'n'.
- 6) Then, we execute the 'for' loop from i=0 to i<n to carry out the difference 'n' number of times. Inside this loop, we take two decimal numbers as the input from the user and the compiler carries out the process of finding the difference between these two decimal numbers. The input and difference process repeats as many times as the user has stored in 'n'.

III. PROGRAM ILLUSTRATION

A. Standard method

Even before taking inputs from the user, we declare a structure 'data' that will store two floating values 'a' and 'b'. Thereafter, we define a function 'dif' that takes the structure as an argument and would give out the difference between the two decimal numbers that would be given as input.

The structure 'data' has been initially declared as 't' inside the main function.

Now we take the input from the user which are 55.6356 and 24.4512 respectively.

The value 55.6356 has been stored in 't.a'
And value 24.4512 has been stored in 't.b'

Then we call the function that has been defined above. The function 'dif' when executed performs the subtraction operation-

```
c = t.a - t.b
c = 55.6356 - 24.4512
c = 31.1844
```

The value 31.1844 gets stored in 'c' (this was the output delivered when $c \geq 0$)

If we interchange the values, then the new values will be :

```
t.a = 24.4512
t.b = 55.6356
c = 24.4512 - 55.6356
c = -31.1844
```

But the function 'dif' declared above will return -c whenever $c \leq 0$. Therefore, the final value that gets stored in c is $-(-31.1844)$ i.e; 31.1844.

After giving the difference as the output, the program asks the user "Do you want more". If the user enters Y/y the program will again ask the user to input two decimal numbers and the function is called yet again. After the execution of the function, we get the difference. Then the 'goto' statement is executed and it jumps back to the statement that asks "Do you want more". If the user enters 'N/n' the process gets finished and entering any other value displays "Wrong output" and leads to the execution of the 'goto' statement yet another time. This continues as long as the user does not enter 'N/n' and the process does not get finished.

B. Using Structure with Array

Even before taking inputs from the user, we declare a structure 'data' that will store three floating values 'a' and 'b' and 'c'. Thereafter, we define functions 'input' and 'dif'. 'input' function stores the input given by the user in the element of array 't' of 'struct data' type and returns the 't'. 'dif' function takes the structure as an argument and would

print the difference between the two decimal numbers that would be given as input.

Let's assume the user wants to run the program three times and the input given are as follows –

- 1) 2.4564 and 1.2658
- 2) 32.4568 and 84.3541
- 3) 4156.6865 and 4156.2359

All these inputs will be stored in the array 't' of 'struct data' type with the help of 'input' function, as shown in the first and second column of the following table:

t.a	t.b	t.c
2.4564	1.2658	1.1906
32.4568	84.3541	52.8973
4156.6865	4156.2359	0.4507

Table-1: Input stored in structure 'data' in the form of arrays.

After that, 'dif' function finds the difference of decimal numbers, in the following way-

```
t.c = t.b - t.a    ( if b>a )
t.c = t.a - t.b    ( if a>b )
```

All the answers depicted in the 3rd column of the above table get printed one by one, every time the user inputs the decimal numbers.

IV. ALGORITHM ANALYSIS

Time complexity:

Time complexity is the computational complexity that describes the amount of computer time it takes to run an algorithm. Time complexity is commonly estimated by counting the number of elementary operations performed by the algorithm, supposing that each elementary operation takes a fixed amount of time to perform.

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

As the second program contains a single looping statement, namely, the 'for' loop, its time complexity will be **O(n)**.

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

V. 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.

VI. REFERENCES

1. <https://www.tutorialspoint.com/c-program-for-program-for-array-rotation>
2. <https://www.geeksforgeeks.org/analysis-of-algorithms-set-4-analysis-of-loops/>
3. <https://www.javatpoint.com/functions-in-c>
4. <https://www.programiz.com/c-programming/c-structures>.

Code (Standard method) for implementation of this paper is given below:

```
#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));
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;
}
```

Listing 1. Code for this paper (standard method)

Code (using a structure with array) for implementation of this paper is given below:

```
#include <stdio.h>

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

// FUNCTION- 'input'
struct data input()
{
    struct data t;
    scanf("%f%f", &t.a, &t.b);
    return t;
};

// FUNCTION- 'dif'
void dif(struct data t)
{
    if ((t.a - t.b) >= 0)
        t.c = t.a - t.b;
    else
        t.c = t.b - t.a;
    printf("Difference= %.4f\n", t.c);
};

// MAIN FUNCTION
int main()
{
    int i, n;
    struct data t[25];
    printf("Enter number of times the difference has to be carried out: ");
    scanf("%d", &n);
    for (i = 0; i < n; i++)
    {
        printf("\nEnter two decimal numbers (upto 4 decimal places)- \n");
        t[i] = input();
        dif(t[i]);
    }

    return 0;
}
```

Listing 2. Code for this paper (using a structure with array)