INSTITUTE OF ENGINEERING

Pulchowk Campus, Lalitpur

Subject: C Programming

Lab Report 11

Tittle: **Structures**

Submitted by:
Susheel Thapa 077BCT090

Submitted to:
Department of Electronics and
Computer Engineering

Checked by

Content of Lab Report:

Background Information

C Programming

Editor Used

Compiler

Structures

Code and Output (Analysis)

Source Code

Output

Analysis

Conclusion

Background Information

What is C Programming?

C programming is a general-purpose, procedural, imperative computer programming language developed in 1972 by Dennis M. Ritchie at the Bell Telephone Laboratories to develop the UNIX operating system. C is the most widely used computer language.

Editor

Here, I have used Visual Studio Code as my editor. You can download the editor from Download Visual Studio Code - Mac, Linux, Windows . Select your operating system and download it.

Compiler

Here, I have used **gcc** as my compiler provided by MinGWw64. You can download it via Download MinGW-w64 - for 32 and 64 bit Windows from SourceForge.net. Your download will start automatically. Run the downloaded .exe file. After, you have install MinGW-w64, you need to configure it.

1. In the Windows search bar, type 'settings' to open your Windows Settings.

2. Search for Edit environment variables for your account.

3. Choose the Path variable and then select Edit.

4. Select New and add the Mingw-w64 destination folder path to the system path. The exact path depends on which version of Mingw-w64 you have installed and where you installed it. If you used the settings above to install Mingw-w64, then add this to the path: **C:\Program Files\mingw-w64\x86_64-8.1.0-posix-seh-rt_v6-rev0\mingw64\bin.**

5. Select OK to save the updated PATH. You will need to reopen any console windows for the new PATH location to be available.

Structure:

It is a heterogeneous user defined data type. It is also called constructed data type. It may contain different data types .Structure can also store non homogenous data type into a single collection. Structure may contain pointer, arrays, or even other structures other than the common data types such as int, float, long int etc. A structure provides a means of grouping variables under a single name for easier handling and identification. It can be defined as new named types. It is a convenient way of grouping several pieces of related information together. Complex hierarchies can be created by nesting structures. Structures may be copied to and assigned. They are also useful in passing groups of logically related data into structures. The declaration of structures is given below:

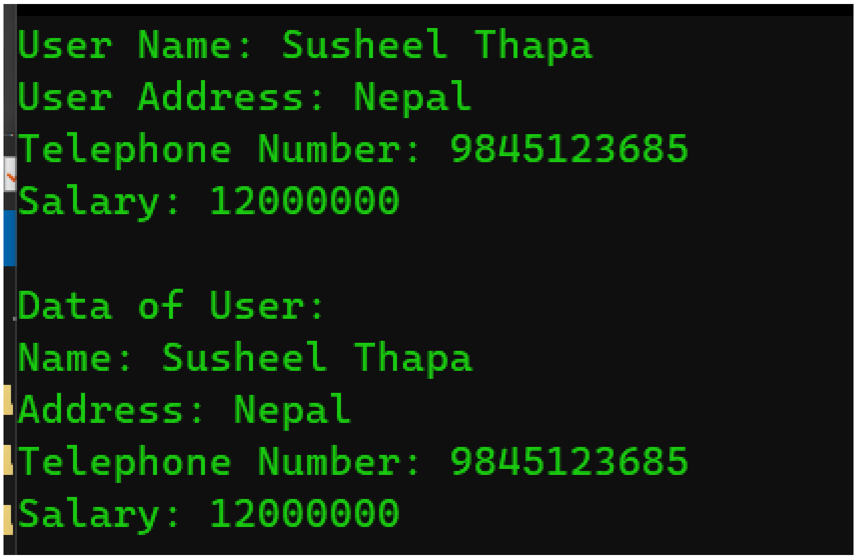
```
struct tag

{
  member 1;
  member 2;

  member n;
};
```

1.Write a program to create a structure having members: Name, Address, Telephone number and Salary of an employee. Read the values of members from the user and display.

Source Code:

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

/*Creating structure*/
struct UserData
{
 /*Member of structure*/
 char user_name[100];
 char user_address[100];
 long long int user_telephone_number;
 long int user_salary;
};

int main()
{
 /*Creating Structure Variable*/
 struct UserData UD_one;

 system("cls");
 /*Taking input from user*/
 printf("User Name: ");
 scanf("%[^'\n']", UD_one.user_name);

 printf("User Address: ");
 scanf(" %[^'\n']", UD_one.user_address);

 printf("Telephone Number: ");
 scanf("%lld", &UD_one.user_telephone_number);

 printf("Salary: ");
 scanf("%ld", &UD_one.user_salary);

 /*Print the taken value*/
 printf("\nData of User:");
 printf("\nName: %s", UD_one.user_name);
 printf("\nAddress: %s", UD_one.user_address);
 printf("\nTelephone Number: %lld", UD_one.user_telephone_number);
   printf("\nSalary: %ld", UD_one.user_salary);
   getch();
   return 0;
}
```

Output:



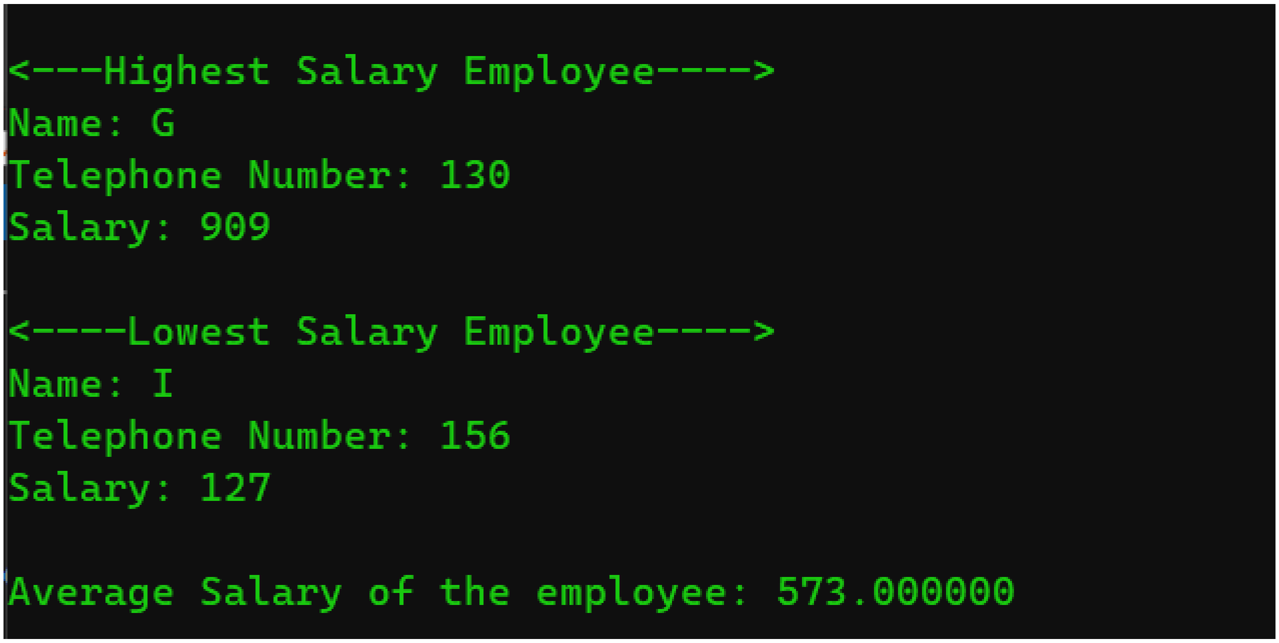
```
User Name: Susheel Thapa
User Address: Nepal
Telephone Number: 9845123685
Salary: 12000000

Data of User:
Name: Susheel Thapa
Address: Nepal
Telephone Number: 9845123685
Salary: 12000000
```

2.Create a structure employee containing names as character string , telephone as character string and salary as intger. Input the records of 10 employees. After that display the name,telephone and salary of employee with highest salary and lowest salary and display the average salary of all 10 employee

Source Code:

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>

struct Employee
{
 /*Member of structure*/
 char name[100];
 char telephone_number[100];
 int salary;
};

int main()
{
 struct Employee e[10];
    int highest_salary, lowest_salary;

    float average_salary = 0;

    system("cls");

    /*Taking input from user*/
    for (int i = 0; i < 10; i++)
    {
        printf("\nEmployee %d\n", i + 1);
        printf("Name: ");
        scanf(" %[^'\n']", e[i].name);

        printf("Telephone Number: ");
        scanf(" %[^'\n']", e[i].telephone_number);

        printf("Salary: ");
        scanf("%ld", &e[i].salary);
    }

    highest_salary = e[0].salary;
    lowest_salary = e[0].salary;

    system("cls");

    for (int i = 0; i < 10; i++)
    {
        (highest_salary < e[i].salary) ? (highest_salary = e[i].salary) : (highest_salary = highest_salary);
        (lowest_salary > e[i].salary) ? (lowest_salary = e[i].salary) : (lowest_salary = lowest_salary);

        average_salary = (average_salary * i + e[i].salary) / (i + 1);
    }

    for (int i = 0; i < 10; i++)
    {
        if (e[i].salary == highest_salary)
        {
            printf("\n<---Highest Salary Employee---->\n");
            printf("Name: %s\n", e[i].name);
            printf("Telephone Number: %s\n", e[i].telephone_number);
            printf("Salary: %d\n", e[i].salary);
        }
        if (e[i].salary == lowest_salary)
        {
            printf("\n<----Lowest Salary Employee---->\n");
            printf("Name: %s\n", e[i].name);
            printf("Telephone Number: %s\n", e[i].telephone_number);
            printf("Salary: %d\n", e[i].salary);
        }
    }

    printf("\nAverage Salary of the employee: %f\n", average_salary);

    getch();
    return 0;
}
```

Output:



```
<---Highest Salary Employee---->
Name: G
Telephone Number: 130
Salary: 909

<----Lowest Salary Employee---->
Name: I
Telephone Number: 156
Salary: 127

Average Salary of the employee: 573.000000
```

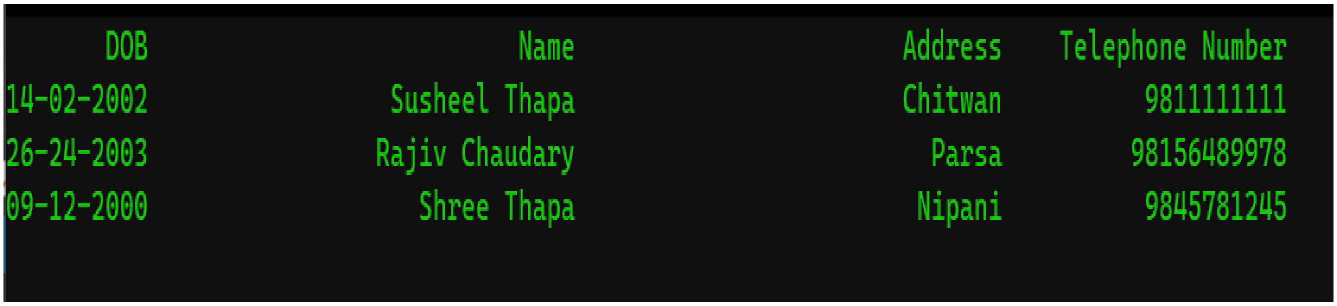
**2.Create a structure Data containing three members: int dd,int mm,int yy.
Create another member date of birth, create an object of structure data inside person.
Using these structure, Write a program to input records until user enter 'n' or 'N'.Then, dispaly the contents in tabular form.
Source Code:**

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>

/*A Structure*/
struct Date
{
   int dd;
   int mm;
   int yy;
};

/*Nested Structure*/
struct Person
{
   char name[100];
   char address[100];
   long long int telephone_number;
   struct Date date_of_birth;
};
```

```
int main()
{
   /*Creating the variable of srtucture*/
   struct Person p[100];
   int records = 0;

   system("cls");

   /*While loop to take data for infinite times till user enter n */
   while (1)
   {
      printf("Name: ");
      scanf(" %[^'\n']", p[records].name);

      if (strcmpi(p[records].name, "n") == 0)
      {
         break;
      }

      printf("Address: ");
      scanf(" %[^'\n']", p[records].address);

      printf("Telephone Number: ");
      scanf("%lld", &p[records].telephone_number);

      printf("Date of Birth:\n");
      printf(" Day: ");
      scanf("%d", &p[records].date_of_birth.dd);
      printf(" Month: ");
      scanf("%d", &p[records].date_of_birth.mm);
      printf(" Year: ");
      scanf("%d", &p[records].date_of_birth.yy);

      records++;
   }

   system("cls");

   /*Printing the data in tabular way*/
   printf("%10s%30s%30s%20s\n", "DOB", "Name", "Address", "Telephone Number");
   for (int i = 0; i < records; i++)
   {
      printf("%02d-%02d-%4d%30s%30s%20lld", p[i].date_of_birth.dd, p[i].date_of_birth.mm,
p[i].date_of_birth.yy, p[i].name, p[i].address, p[i].telephone_number);
      printf("\n");
```

```
    }

    getch();
    return 0;
}
```

Output:



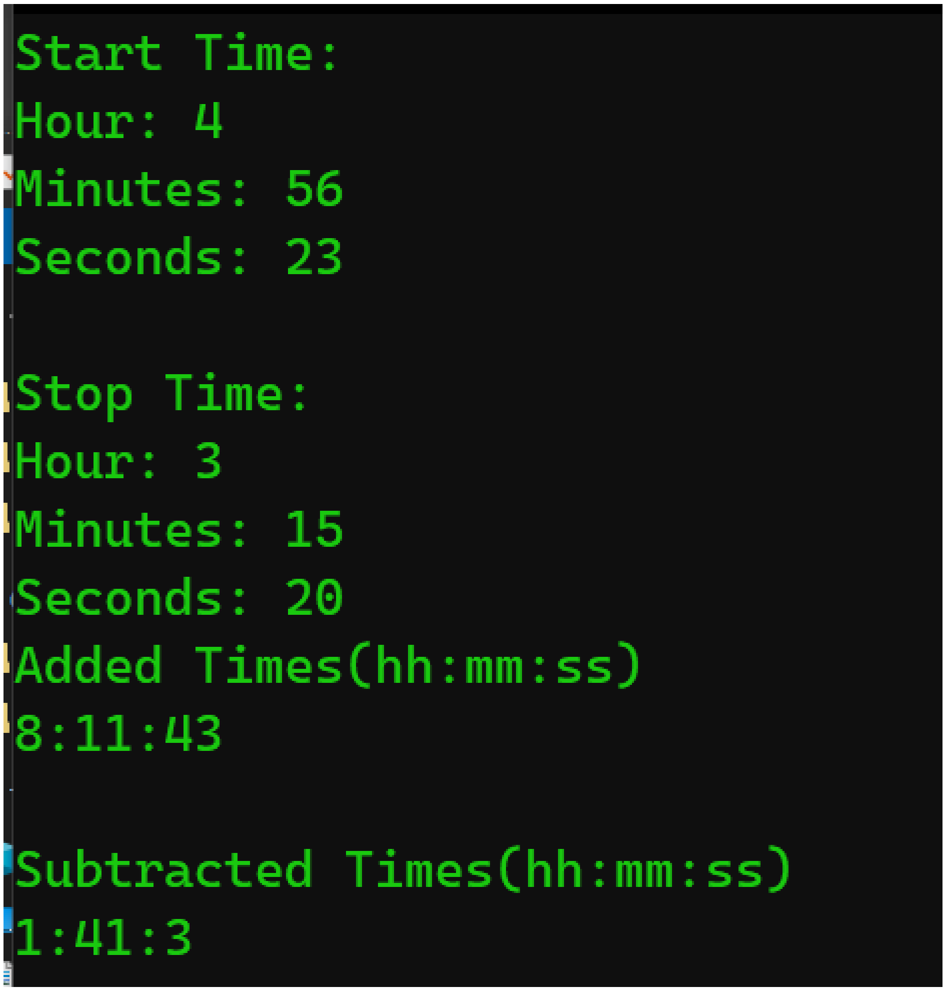
```
       DOB                         Name                      Address    Telephone Number
14-02-2002                Susheel Thapa                      Chitwan          9811111111
26-24-2003               Rajiv Chaudary                        Parsa         98156489978
09-12-2000                  Shree Thapa                       Nipani          9845781245
```

4.Create a structure TIME containing hour, minutes and seconds as its member. Write a program that uses this structure to input start time and stop time in main(). Pass the structure to a function that calculated the sum and diffrenece of start time and stop time.Display the sum and difference from main()

Source Code:

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>

/*Creating Structure*/
struct TIME
{
    int hh;
    int mm;
    int ss;
};

/*Function with type of created structure*/
struct TIME addTime(struct TIME first, struct TIME second)
{
    "Add the two time passed";

    struct TIME result;

    result.ss = first.ss + second.ss;
    result.mm = first.mm + second.mm;
    result.hh = first.hh + second.hh;

    if (result.ss >= 60)
    {
        result.ss = result.ss - 60;
        result.mm = result.mm + 1;
    }
    if (result.mm >= 60)
    {
        result.mm = result.mm - 60;
        result.hh = result.hh + 1;
    }
    return result;
}

struct TIME subtractTime(struct TIME first, struct TIME second)
{
    "Subtract the two time passes";

    struct TIME result;

    result.ss = first.ss - second.ss;
    result.mm = first.mm - second.mm;
    result.hh = first.hh - second.hh;

    if (result.ss < 0)
    {
        result.ss = result.ss + 60;
        result.mm = result.mm - 1;
    }
    if (result.mm < 0)
    {
        result.mm = result.mm + 60;
        result.hh = result.hh - 1;
    }
    return result;
}

int main()
{
    struct TIME start, stop, add, subtract;

    system("cls");

    /*Taking input*/
    printf("Start Time:\n");
    printf("Hour: ");
    scanf("%d", &start.hh);
    printf("Minutes: ");
    scanf("%d", &start.mm);
    printf("Seconds: ");
    scanf("%d", &start.ss);

    printf("\nStop Time:\n");
    printf("Hour: ");
    scanf("%d", &stop.hh);
    printf("Minutes: ");
    scanf("%d", &stop.mm);
    printf("Seconds: ");
    scanf("%d", &stop.ss);

    /*Function call*/
    add = addTime(start, stop);
    subtract = subtractTime(start, stop);

    /*Printing result*/
    printf("Added Times(hh:mm:ss)\n");
    printf("%d:%d:%d\n", add.hh, add.mm, add.ss);

    printf("\nSubtracted Times(hh:mm:ss)\n");
    printf("%d:%d:%d\n", subtract.hh, subtract.mm, subtract.ss);

    getch();
    return 0;
}
```

Output:



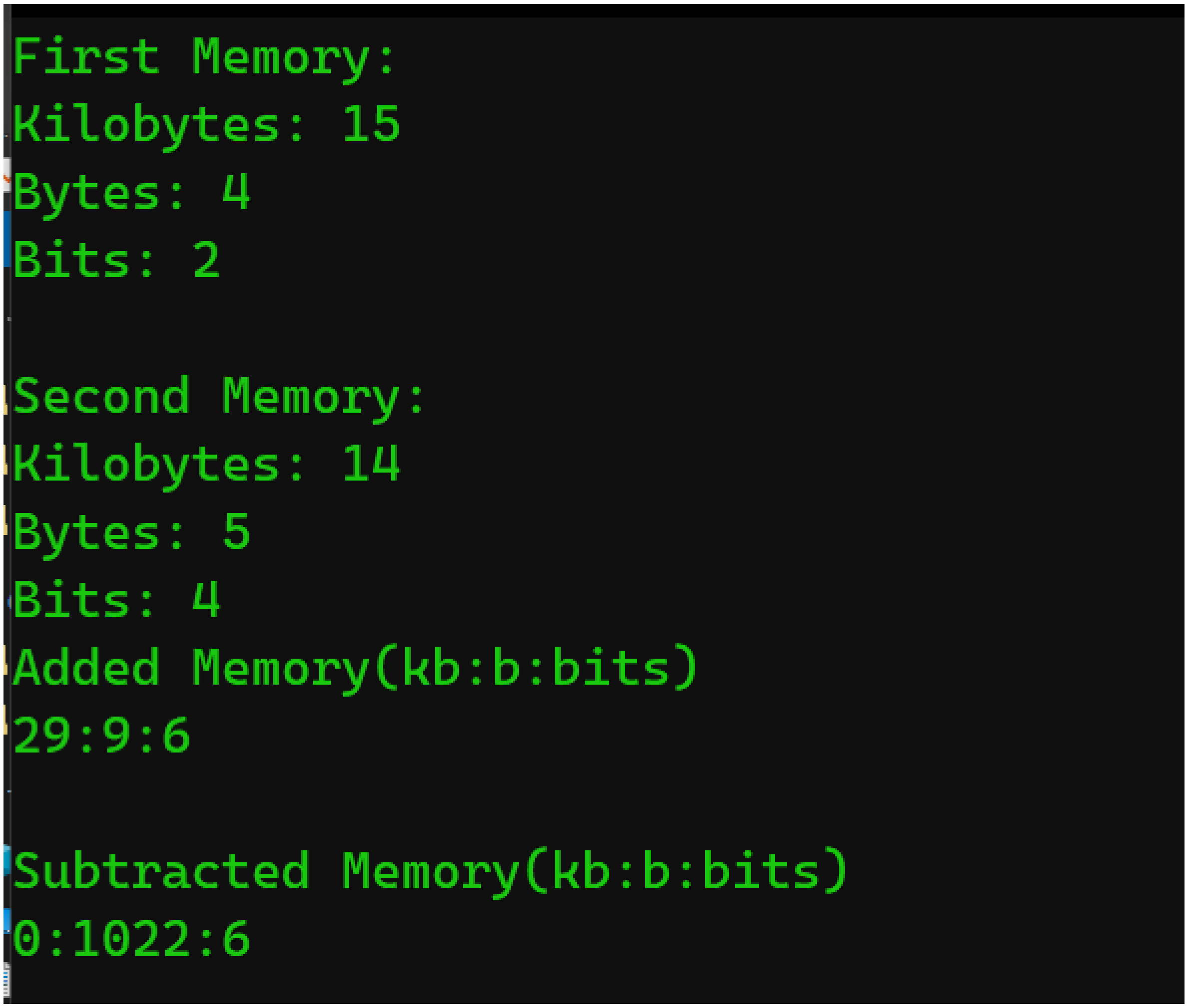
```
Start Time:
Hour: 4
Minutes: 56
Seconds: 23

Stop Time:
Hour: 3
Minutes: 15
Seconds: 20
Added Times(hh:mm:ss)
8:11:43

Subtracted Times(hh:mm:ss)
1:41:3
```

5.Write a program to compute any two instant memory spaces in format (Kilobytes:Bytes:Bits) using structure. Build function to add and subtract given memory spaces where 1kb = 1-24b and 1b = 8bits and display the result from main()
Source Code:

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>

/*Creating Structure*/
struct Memory
{
 int kb;
 int b;
 int bits;
};
```

```
/*Function with type of created structure*/
struct Memory addMemory(struct Memory first, struct Memory second)
{
    "Add the two memory passed";

    struct Memory result;

    result.bits = first.bits + second.bits;
    result.b = first.b + second.b;
    result.kb = first.kb + second.kb;

    if (result.bits >= 8)
    {
        result.bits = result.bits - 8;
        result.b = result.b + 1;
    }
    if (result.b >= 1024)
    {
        result.b = result.b - 1024;
        result.kb = result.kb + 1;
    }
    return result;
}

struct Memory subtractMemory(struct Memory first, struct Memory second)
{
    "Subtract the two Memory passed";

    struct Memory result;

    result.bits = first.bits - second.bits;
    result.b = first.b - second.b;
    result.kb = first.kb - second.kb;

    if (result.bits < 0)
    {
        result.bits = result.bits + 8;
        result.b = result.b - 1;
    }
    if (result.b < 0)
    {
        result.b = result.b + 1024;
        result.kb = result.kb - 1;
    }
    return result;
}

int main()
{
    struct Memory first, second, add, subtract;

    system("cls");

    /*Taking input*/
    printf("First Memory:\n");
    printf("Kilobytes: ");
    scanf("%d", &first.kb);
    printf("Bytes: ");
    scanf("%d", &first.b);
    printf("Bits: ");
    scanf("%d", &first.bits);

    printf("\nSecond Memory:\n");
    printf("Kilobytes: ");
    scanf("%d", &second.kb);
    printf("Bytes: ");
    scanf("%d", &second.b);
    printf("Bits: ");
    scanf("%d", &second.bits);

    /*Function call*/
    add = addMemory(first, second);
    subtract = subtractMemory(first, second);

    /*Printing result*/
    printf("Added Memory(kb:b:bits)\n");
    printf("%d:%d:%d\n", add.kb, add.b, add.bits);

    printf("\nSubtracted Memory(kb:b:bits)\n");
    printf("%d:%d:%d\n", subtract.kb, subtract.b, subtract.bits);

    getch();
    return 0;
}
```

Output:



```
First Memory:
Kilobytes: 15
Bytes: 4
Bits: 2

Second Memory:
Kilobytes: 14
Bytes: 5
Bits: 4
Added Memory(kb:b:bits)
29:9:6

Subtracted Memory(kb:b:bits)
0:1022:6
```

Analysis
We learn about how we can declare new data types, use it , take data and print out the taken data and many more.

Conclusion:
From this we have learn about structures