Green University of Bangladesh

Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering

Semester:2nd (Summer, Year:2022), B.Sc. in CSE (Day)

LAB REPORT NO: 07

Course Title: Structured Programming Lab

Course Code: CSE 104 / Section: DK

Lab Experiment Name: User Defined Function.

Student Details

Name	ID
Khondokar Saim	221902353

Lab Date : 08 / 08 / 2022

Submission Date : 13 / 09 / 2022

Course Teacher's Name : Monoshi Kumar Roy

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status

Marks: ………………………………… Signature:....................

Comments:.. Date:..............................

- **Tittle of the Lab experiment :** User Defined Function.

- **Objectives :**

In C programming functions are of two types. Such as,

1.Library Functions:
Library functions are built-in functions directly declared in C header files and placed in a common location called a library. such as scanf(), printf(), gets() etc. All functions have some specific task to be performed.
scanf () – used to get the input from the user
printf () – used to print the output to the screen

2.User-defined Functions:
User-defined functions are the functions that are created by the user, and the user can reuse them multiple times. A user-defined function mainly has three components.

- Function declaration
- Function call
- Function definition

➢ User-defined functions are functions that you use to organize your code in the body of a policy.

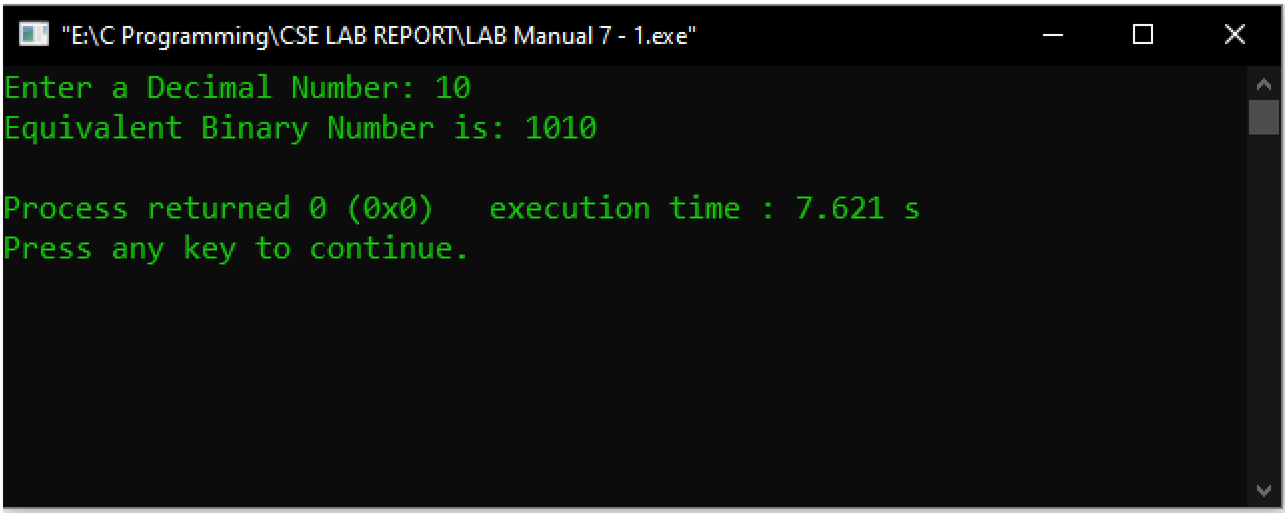
➢ Once you define a function, you can call it in the same way as the built-in action and parser functions. Variables that are passed to a function are passed by reference, rather than by value. This means that changing the value of a variable within a function also changes the value of the variable in the general scope of the policy.

1. Write a C Program to convert a decimal number to an equivalent binary number using function. [Decimal to Binary Conversion Method].

Code :

```
#include <stdio.h>
#include <math.h>

long decimalToBinary(int decimalnum)
{
    long binarynum = 0;
    int rem, temp = 1;

    while (decimalnum!=0)
    {
        rem = decimalnum%2;
        decimalnum = decimalnum / 2;
        binarynum = binarynum + rem*temp;
        temp = temp * 10;
    }
    return binarynum;
}

int main()
{
    int decimalnum;
    printf("Enter a Decimal Number: ");
    scanf("%d", &decimalnum);
    printf("Equivalent Binary Number is: %ld\n",
decimalToBinary(decimalnum));
    return 0;
}
```

Output :



```
Enter a Decimal Number: 10
Equivalent Binary Number is: 1010

Process returned 0 (0x0)   execution time : 7.621 s
Press any key to continue.
```

2. Write a C program to create menu driven calculator that performs basic arithmetic operations (add, subtract, multiply and divide) using functions.

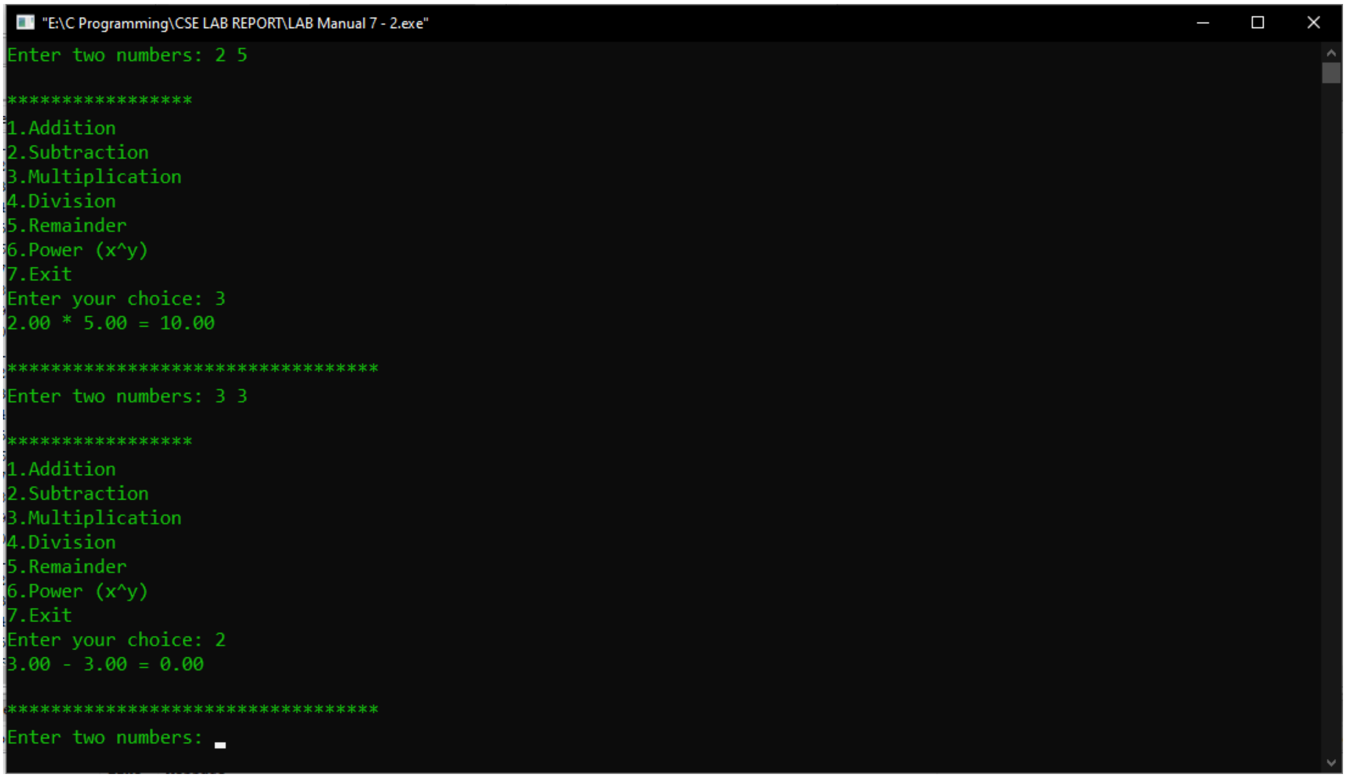
Code :

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

// function declarations
void display(float n1, float n2, char ch, float result);
void add(float n1, float n2);
void subtract(float n1, float n2);
void multiply(float n1, float n2);
void divide(float n1, float n2);
void rem(float n1, float n2);
void power(float n1, float n2);

// main function
int main()
{
  float n1, n2;
  int ch;

  do{
    printf("Enter two numbers: ");
    scanf("%f %f", &n1, &n2);

    printf("\n*****************");
    printf("\n1.Addition");
    printf("\n2.Subtraction");
    printf("\n3.Multiplication");
    printf("\n4.Division");
    printf("\n5.Remainder");
    printf("\n6.Power (x^y)");
printf("\n6.Power (x^y)");
  printf("\n7.Exit");
  printf("\nEnter your choice: ");
  scanf("%d", &ch);

  switch (ch) {
    case 1:
      add(n1,n2);
      break;
    case 2:
      subtract(n1,n2);
      break;
    case 3:
      multiply(n1,n2);
      break;
    case 4:
      divide(n1,n2);
      break;
    case 5:
      rem(n1,n2);
      break;
    case 6:
      power(n1,n2);
      break;
    case 7:
      printf("Thank You.");
      exit(0);
    default:
      printf("Invalid input.");
      printf("Please enter correct input.");
  }
```

```
printf("\n*************************************\n");
  }while(1);

  return 0;
}

// function for displaying the result
void display(float n1, float n2, char ch, float result)
{
  printf("%.2f %c %.2f = %.2f\n", n1, ch, n2, result);
}

// function for addition of two numbers
void add(float n1, float n2)
{
  float result = n1 + n2;
  display(n1, n2, '+', result);
}

// function for subtraction of two numbers
void subtract(float n1, float n2)
{
  float result = n1 - n2;
  display(n1, n2, '-', result);
}

// function for multiplication of two numbers
void multiply(float n1, float n2)
{
  float result = n1 * n2;
  display(n1, n2, '*', result);
}
```

```
// function for division of two numbers
void divide(float n1, float n2)
{
  float result = n1 / n2;
  display(n1, n2, '/', result);
}

// function for calculating remainder
void rem(float n1, float n2)
{
  //Modulus operator only works on int data type
  //Floating numbers are converted to int number
  int num1 = n1;
  int num2 = n2;
  int result = num1%num2;
  printf("%d %% %d = %d\n", num1, num2, result);
}

// function for calculating power
void power(float n1, float n2)
{
  if(n2<0) printf("Second number should be +ve.");
  else
  {
    float result=1.0;
    for(int i=1; i<=n2; i++)
    {
       result *= n1;
    }
    display(n1, n2, '^', result);
  }
}
```

Output :



"E:\C Programming\CSE LAB REPORT\LAB Manual 7 - 2.exe"

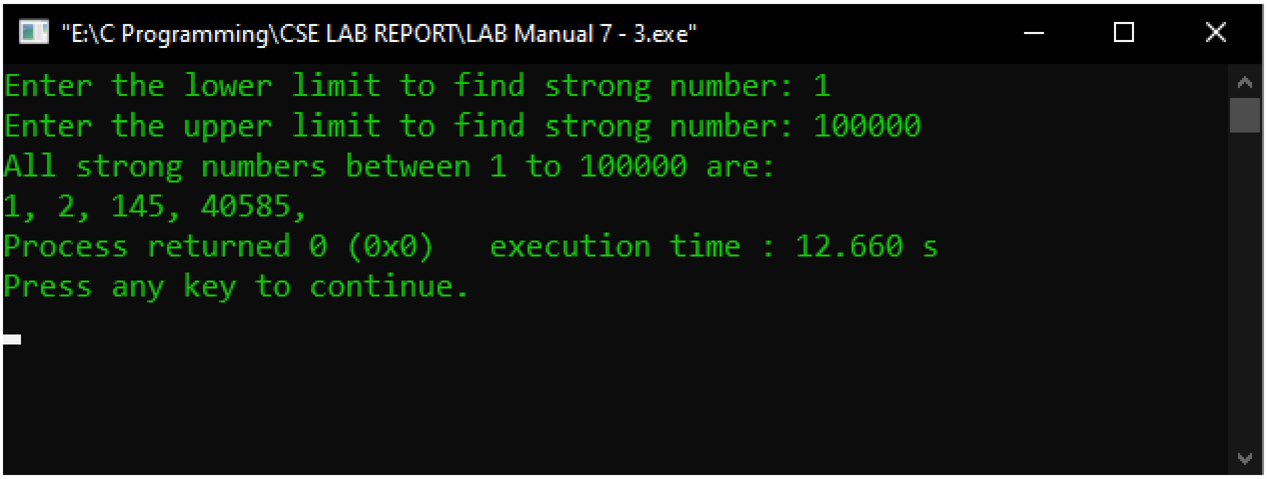
```
Enter two numbers: 2 5

*****************
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Remainder
6.Power (x^y)
7.Exit
Enter your choice: 3
2.00 * 5.00 = 10.00

**********************************
Enter two numbers: 3 3

*****************
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Remainder
6.Power (x^y)
7.Exit
Enter your choice: 2
3.00 - 3.00 = 0.00

**********************************
Enter two numbers: _
```

3. Write a C Program to print Strong Numbers between given interval using function.

Code :

```
#include <stdio.h>

long long fact(int num);
void printStrongNumbers(int start, int end);

int main()
{
    int start, end;

    //Input start and end range
    printf("Enter the lower limit to find strong number: ");
    scanf("%d", &start);
    printf("Enter the upper limit to find strong number: ");
    scanf("%d", &end);

    printf("All strong numbers between %d to %d are: \n", start, end);
    printStrongNumbers(start, end);

    return 0;
}
```

Output :



"E:\C Programming\CSE LAB REPORT\LAB Manual 7 - 3.exe"

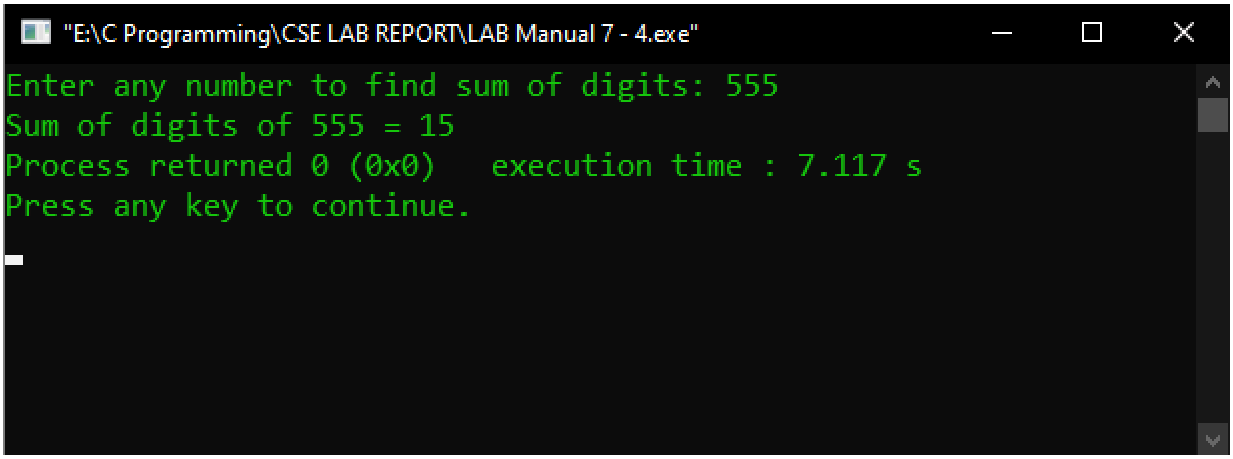
```
Enter the lower limit to find strong number: 1
Enter the upper limit to find strong number: 100000
All strong numbers between 1 to 100000 are:
1, 2, 145, 40585,
Process returned 0 (0x0)   execution time : 12.660 s
Press any key to continue.
```

3. Write a C program to calculate sum of all digits of a number using recursion.

Code :

```
#include <stdio.h>

// Function declaration
int sumOfDigits(int num);


int main()
{
    int num, sum;

    printf("Enter any number to find sum of digits: ");
    scanf("%d", &num);

    sum = sumOfDigits(num);

    printf("Sum of digits of %d = %d", num, sum);

    return 0;
}

//Recursive function to find sum of digits of a number
int sumOfDigits(int num)
{
    // Base condition
    if(num == 0)
        return 0;

    return ((num % 10) + sumOfDigits(num / 10));
}
```

Output :



```
Enter any number to find sum of digits: 555
Sum of digits of 555 = 15
Process returned 0 (0x0)   execution time : 7.117 s
Press any key to continue.
```

• ANALYSIS AND DISCUSSION :

1. We got the exact result on output. Sometimes the result was wrong but we found the right implementation.

2. The problem of display anything in output is the easiest implementation. We solve that very easy.

3. In this assignment, we were faced some problem on last two question but by the teachers help we solve it.

4. Nothing is very/most difficult parts in my program to implement.

5. All program is easy to understand and these helped me a lot to remove my confusion about c programming.

6. I learnt-function call , function declaration, function definition, on program and many basic things about c programming.

Green University of Bangladesh

Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering

Semester:2nd (Summer, Year:2022), B.Sc. in CSE (Day)

LAB REPORT NO: 08

Course Title: Structured Programming Lab

Course Code: CSE 104 / Section: DK

Lab Experiment Name: String processing .

Student Details

Name	ID
Khondokar Saim	221902353

Lab Date : 15 / 08 / 2022

Submission Date : 13 / 09 / 2022

Course Teacher's Name : Monoshi Kumar Roy

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status

Marks: ………………………………… Signature:....................

Comments:.. Date:.............................

- **Tittle of the Lab experiment :** String processing.

- **Objectives :**

Strings are used for storing text/characters.

For example, "Hello World" is a string of characters.

Unlike many other programming languages, does not have a string type to easily create string variables. However, you we can use the char type and create an array of characters to make a string in C:

1. Access Strings

Since strings are actually arrays in C, you can access a string by referring to its index number inside square brackets [].

2. Modify Strings

To change the value of a specific character in a string, refer to the index number, and use single quotes:

3. Another Way Of Creating Strings

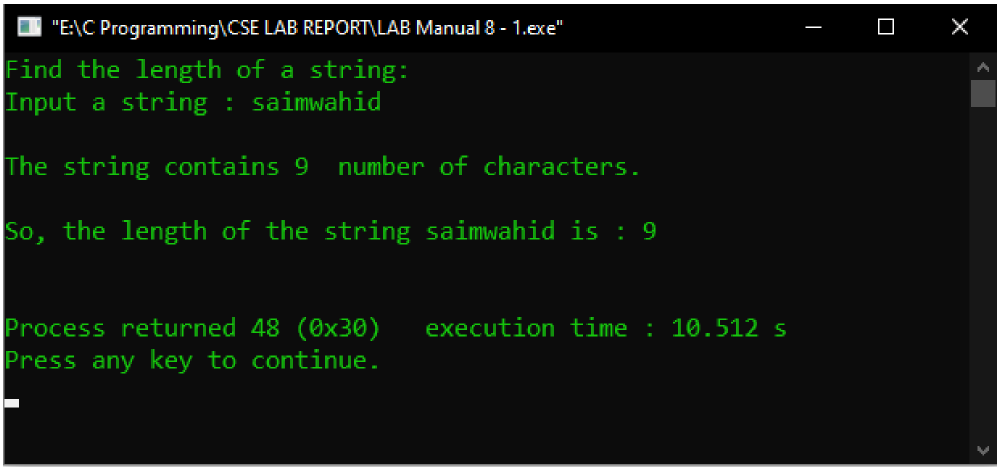
In the examples above, we used a 'string literal to create a string variable. This is the easiest way to create a

1. Write a program in C to find the length of a string without using library function.

Code :

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

 void main()
{
   char str1[50];
   int i, l = 0;

   printf("Find the length of a string:\n");

   printf("Input a string : ");
   scanf("%s", str1);

   for (i = 0; str1[i] != '\0'; i++)
   {
      l++;
   }
   printf("\nThe string contains %d  number of characters. \n",l);
   printf("\nSo, the length of the string %s is : %d\n\n", str1, l);
}
```

Output :



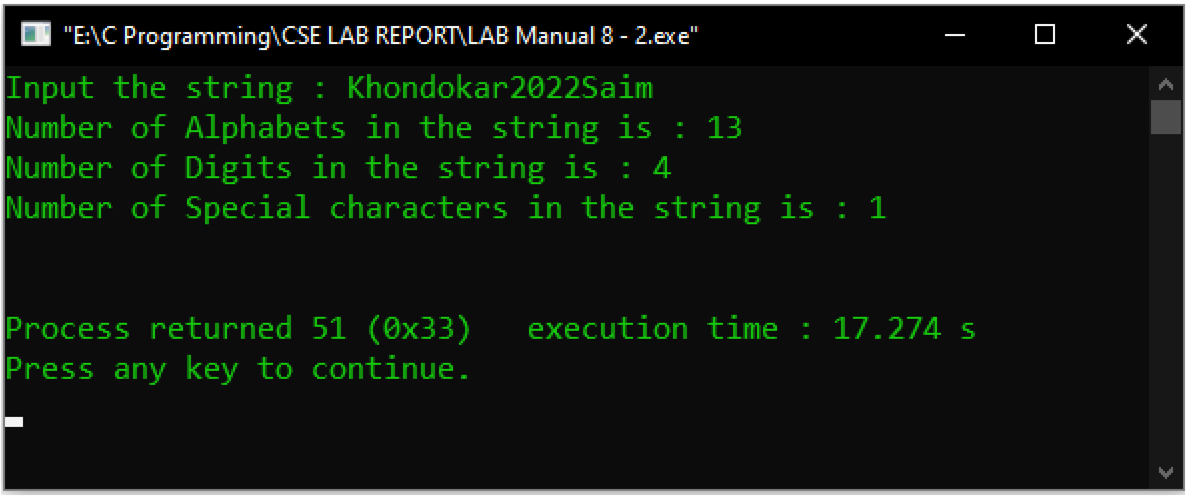
```
Find the length of a string:
Input a string : saimwahid

The string contains 9  number of characters.

So, the length of the string saimwahid is : 9


Process returned 48 (0x30)   execution time : 10.512 s
Press any key to continue.
```

2. Write a program in C to count total number of alphabets, digits and special characters in a string.

Code :

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define str_size 100 //Declare the maximum size of the string

void main()
{
    char str[str_size];
    int alp, digit, splch, i;
    alp = digit = splch = i = 0;

    printf("Input the string : ");
    fgets(str, sizeof str, stdin);

    //Checks each character of string

    while(str[i]!='\0')
    {
        if((str[i]>='a' && str[i]<='z') || (str[i]>='A' && str[i]<='Z'))
        {
            alp++;
        }
        else if(str[i]>='0' && str[i]<='9')
        {
            digit++;
        }
        else
        {
            splch++;
        }
        i++;
    }

    printf("Number of Alphabets in the string is : %d\n", alp);
    printf("Number of Digits in the string is : %d\n", digit);
    printf("Number of Special characters in the string is : %d\n\n",
splch);
}
```

Output :



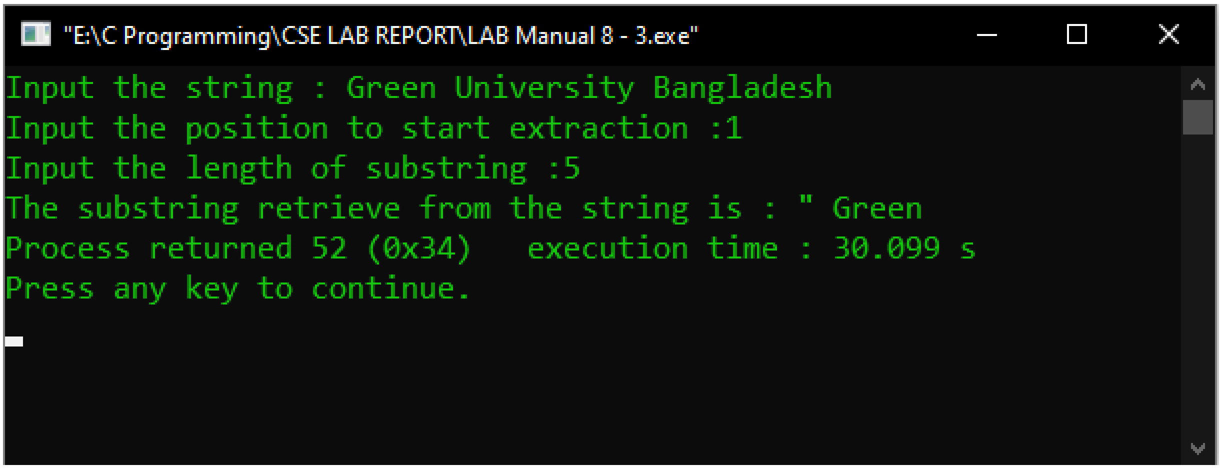
```
Input the string : Khondokar2022Saim
Number of Alphabets in the string is : 13
Number of Digits in the string is : 4
Number of Special characters in the string is : 1


Process returned 51 (0x33)   execution time : 17.274 s
Press any key to continue.
```

3. Write a program in C to extract a substring from a given string.

Code :

```
#include <stdio.h>
void main()
{
   char str[100], sstr[100];
   int pos, l, c = 0;

        printf("Input the string : ");
       fgets(str, sizeof str, stdin);

   printf("Input the position to start extraction :");
   scanf("%d", &pos);

   printf("Input the length of substring :");
   scanf("%d", &l);

   while (c < l)
   {
      sstr[c] = str[pos+c-1];
      c++;
   }
   sstr[c] = '\0';

   printf("The substring retrieve from the string is : \" %s\ ", sstr);

}
```

Output :



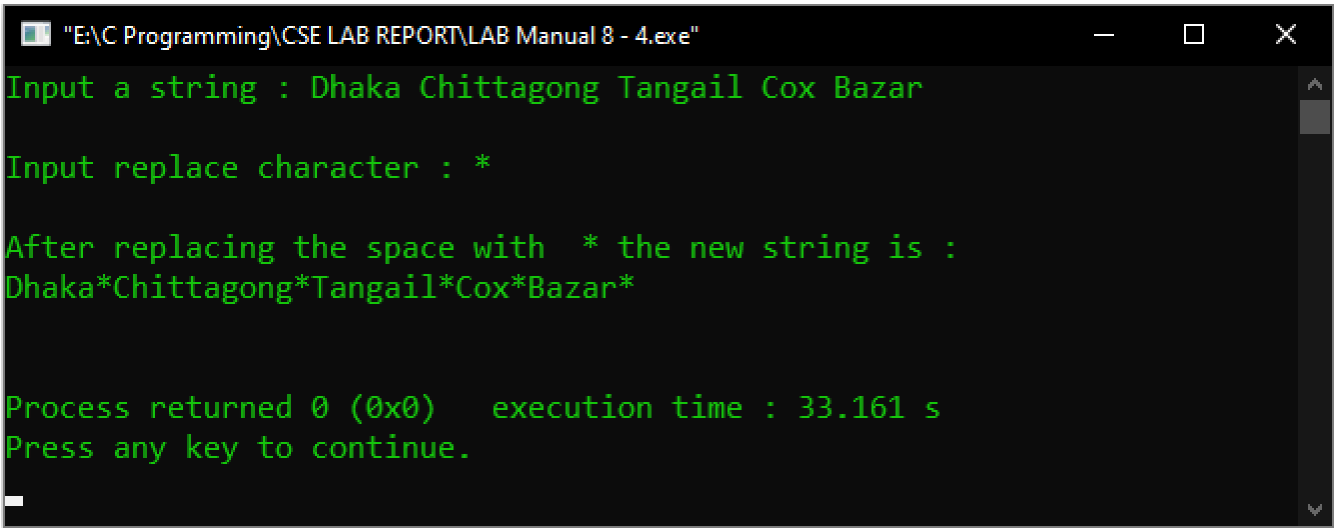
```
Input the string : Green University Bangladesh
Input the position to start extraction :1
Input the length of substring :5
The substring retrieve from the string is : " Green
Process returned 52 (0x34)   execution time : 30.099 s
Press any key to continue.
```

4. Write a program in C to replace the spaces of a string with a specific character.

Code :

```
#include<stdio.h>
#include<ctype.h>

int main()
{
        int new_char;
        char t;
        int ctr=0;
        char str[100];

    printf("Input a string : ");
        fgets(str, sizeof str, stdin);
    printf("\nInput replace character : ");
        scanf("%c",&t);
        printf("\nAfter replacing the space with  %c the new string is 
:\n",t);
        while (str[ctr])
        {
                new_char=str[ctr];
                if (isspace(new_char))
                new_char=t;
                putchar (new_char);
                ctr++;
        }
        printf("\n\n");
        return 0;
}
```

Output :



```
Input a string : Dhaka Chittagong Tangail Cox Bazar

Input replace character : *

After replacing the space with  * the new string is :
Dhaka*Chittagong*Tangail*Cox*Bazar*


Process returned 0 (0x0)   execution time : 33.161 s
Press any key to continue.
```

• ANALYSIS AND DISCUSSION :

1. We got the exact result on output. Sometimes the result was wrong but we found the right implementation.

2. The problem of display anything in output is the easiest implementation. We solve that very easy.

3. In this assignment, we were faced some problem on last two question but by the teachers help we solve it.

4. Nothing is very/most difficult parts in my program to implement.

5. All program is easy to understand and these helped me a lot to remove my confusion about c programming.

6. I learnt-string formatting ,string length, string comparison, replace character, extract string on program and many basic things about c programming.

Green University of Bangladesh

Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering

Semester:2nd (Summer, Year:2022), B.Sc. in CSE (Day)

LAB REPORT NO: 09

Course Title: Structured Programming Lab

Course Code: CSE 104 / Section: DK

Lab Experiment Name: Structure in C .

Student Details

Name	ID
Khondokar Saim	221902353

Lab Date : 22 / 08 / 2022

Submission Date : 13 / 09 / 2022

Course Teacher's Name : Monoshi Kumar Roy

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status

Marks: ………………………………… Signature:.....................

Comments:... Date:..............................

- **Tittle of the Lab experiment : Structure in C**

- **Objectives :**

Arrays allow to define type of variables that can hold several data items of the same kind. Similarly structure is another user defined data type available in C that allows to combine data items of different kinds.

Structures are used to represent a record. Suppose you want to keep track of your books in a library. You might want to track the following attributes about each book –

- Title
- Author
- Subject
- Book ID

To define a structure, you must use the struct statement. The struct statement defines a new data type, with more than one member.

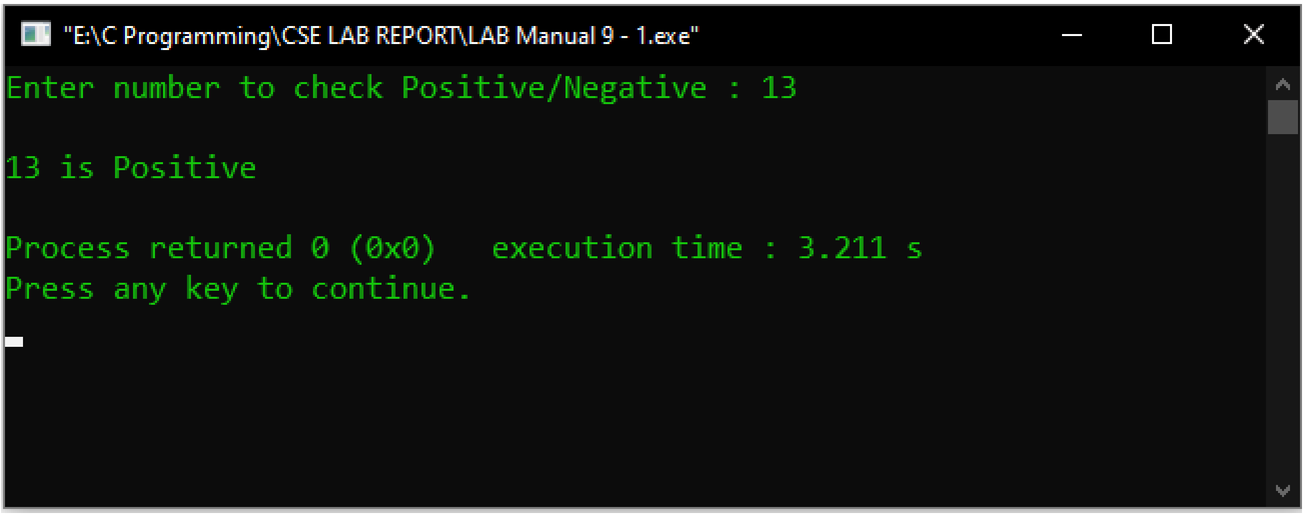
1. Write a C program to check a number is positive or negative.

Code :

```
#include <stdio.h>
struct checknumber
{
int num;
};

int main()
{
    struct checknumber test;
    printf("Enter number to check Positive/Negative : ");
    scanf("%d", &test.num);

    printf("\n");

    if(test.num >= 0){
        printf("%d is Positive\n", test.num);
    }
    else{
        printf("%d is Negative\n", test.num);
    }
}
```

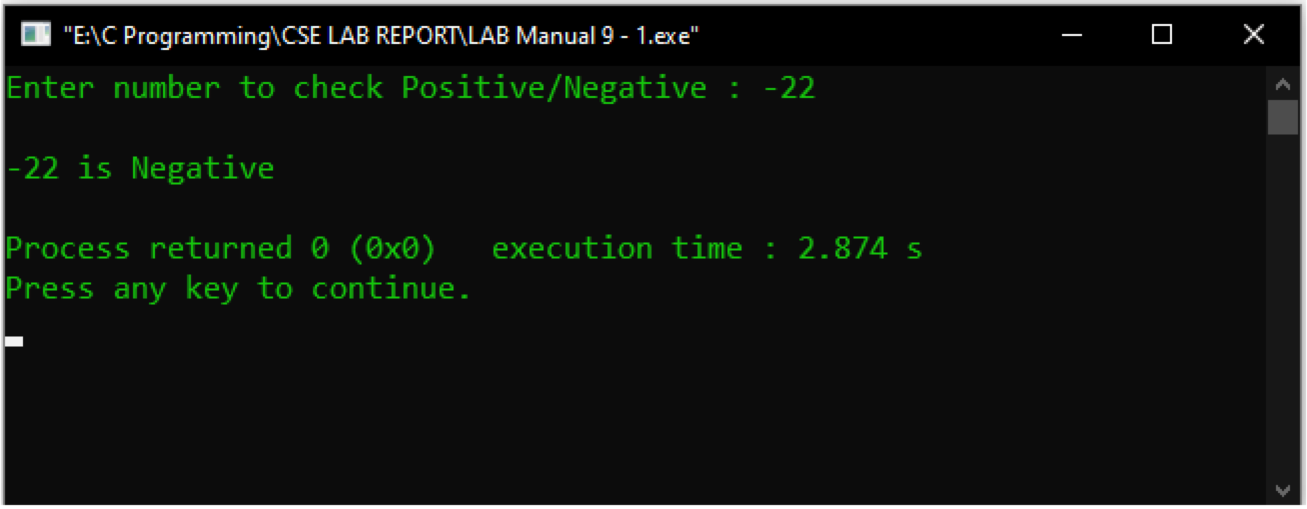
Output :

Part - 1


"E:\C Programming\CSE LAB REPORT\LAB Manual 9 - 1.exe"

```
Enter number to check Positive/Negative : 13

13 is Positive

Process returned 0 (0x0)   execution time : 3.211 s
Press any key to continue.
```

Part - 2


"E:\C Programming\CSE LAB REPORT\LAB Manual 9 - 1.exe"

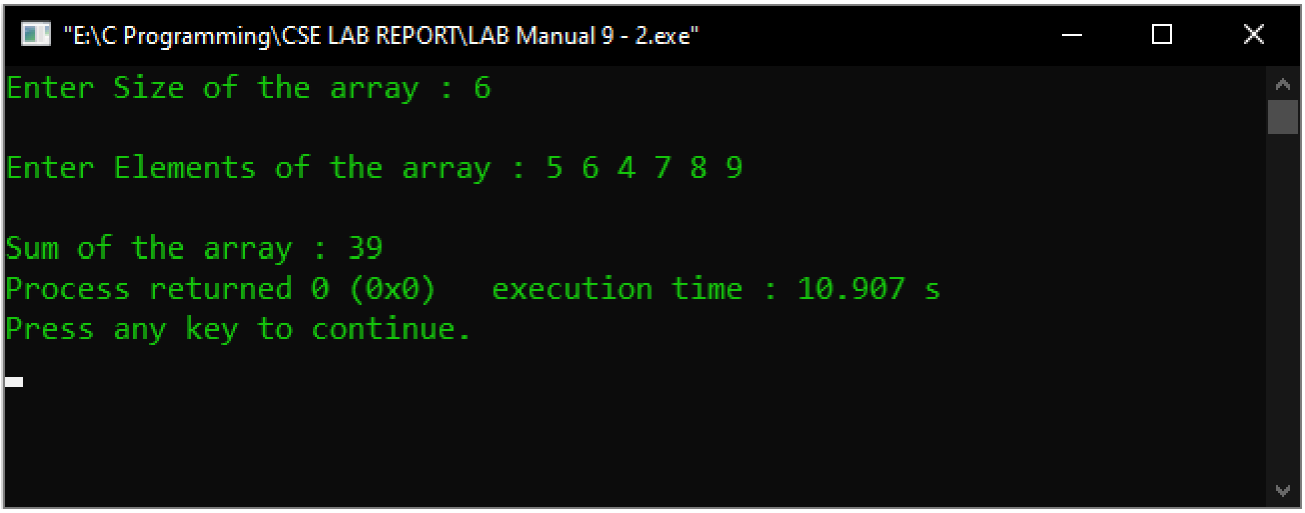
```
Enter number to check Positive/Negative : -22

-22 is Negative

Process returned 0 (0x0)   execution time : 2.874 s
Press any key to continue.
```

2. Write a C program for input N numbers in array and print the summation of all value in array.

Code :

```
#include <stdio.h>
struct store_array
{
    int array[10];
};

int main()
{

    struct store_array arr;
    int size, sum=0;
    printf("Enter Size of the array : ");
    scanf("%d", &size);

    printf("\n");

    printf("Enter Elements of the array : ");
    for(int i=0; i<size; i++)
    {
        scanf("%d", &arr.array[i]);
        sum = sum + arr.array[i];
    }
    printf("\n");
    printf("Sum of the array : %d", sum);

    return 0;
}
```

Output :



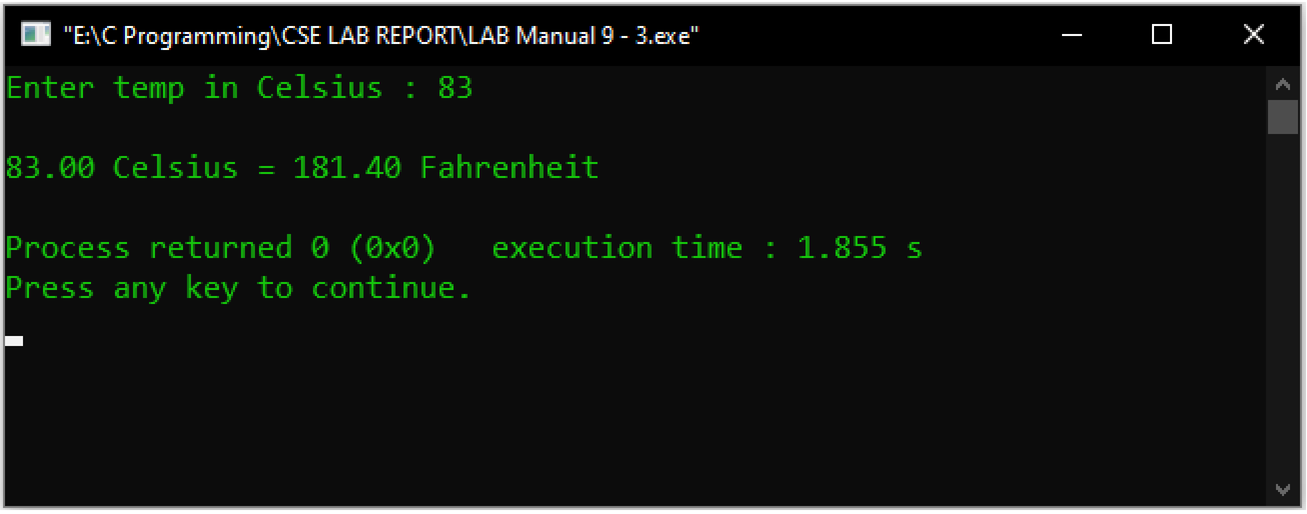
```
Enter Size of the array : 6

Enter Elements of the array : 5 6 4 7 8 9

Sum of the array : 39
Process returned 0 (0x0)   execution time : 10.907 s
Press any key to continue.
```

3. Write a C program to calculate Celsius to Fahrenheit .

Code :

```
#include <stdio.h>
struct celToFahren
{
    float celsius;
};

int main()
{
    struct celToFahren degree;
    float fahrenheit;

    printf("Enter temp in Celsius : ");
    scanf("%f", &degree.celsius);

    printf("\n");

    fahrenheit = (degree.celsius * 9 / 5) + 32;

    printf("%.2f Celsius = %.2f Fahrenheit\n", degree.celsius,
fahrenheit);

    return 0;
}
```

Output :



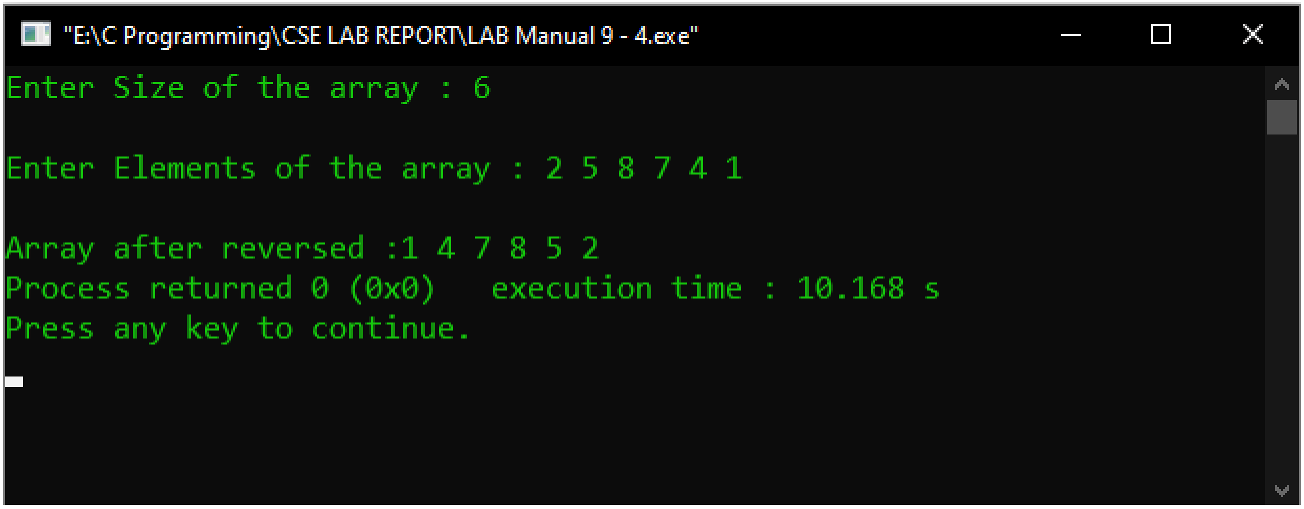
```
Enter temp in Celsius : 83

83.00 Celsius = 181.40 Fahrenheit

Process returned 0 (0x0)   execution time : 1.855 s
Press any key to continue.
```

4. Write a C Program to take input N size array and print reverse array.

Code :

```
#include <stdio.h>
struct store_array
{
    int array[10];
};

int main()
{
    struct store_array arr;
    int size;
    printf("Enter Size of the array : ");
    scanf("%d", &size);

    printf("\n");

    printf("Enter Elements of the array : ");
    for(int i=0; i<size; i++)
    {
        scanf("%d", &arr.array[i]);
    }
    printf("\n");

    printf("Array after reversed :");

    for(int i=size-1; i>=0; i--)
    {
        printf("%d ", arr.array[i]);
    }

    return 0;
}
```

Output :



```
Enter Size of the array : 6

Enter Elements of the array : 2 5 8 7 4 1

Array after reversed :1 4 7 8 5 2
Process returned 0 (0x0)   execution time : 10.168 s
Press any key to continue.
```

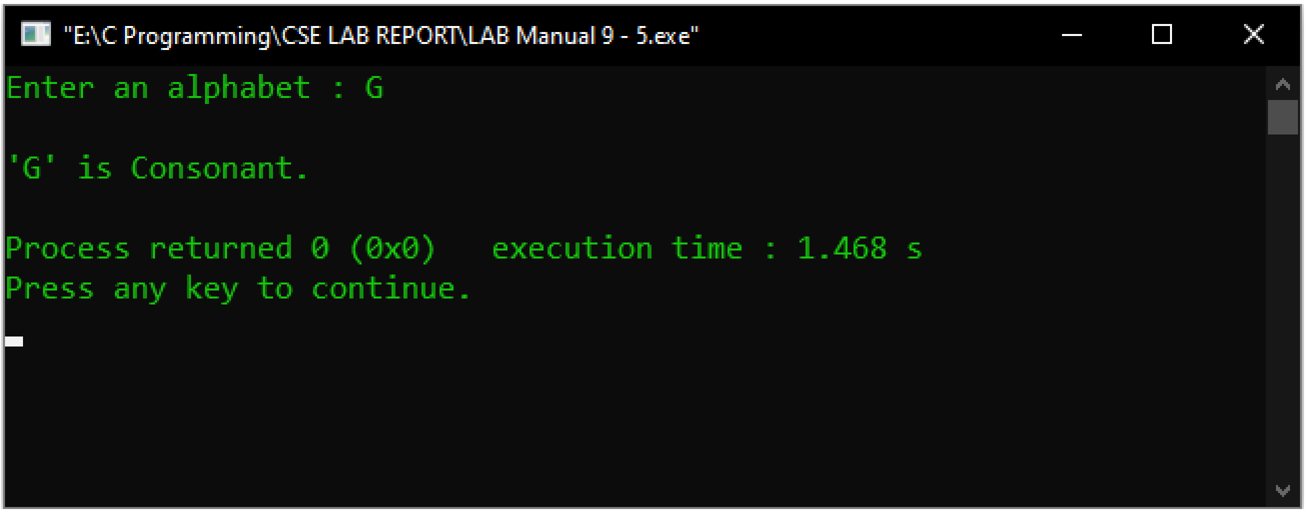
5. Write a C program to check a character is vowel or not.

Code :

```
#include <stdio.h>
struct alphabet
{
    char ch;
};

int main()
{
    struct alphabet vowel;

    printf("Enter an alphabet : ");
    scanf("%c", &vowel.ch);

    printf("\n");
    if(vowel.ch=='a' || vowel.ch=='e' || vowel.ch=='i' || vowel.ch=='o' ||
vowel.ch=='u' ||
       vowel.ch=='A' || vowel.ch=='E' || vowel.ch=='I' || vowel.ch=='O' ||
vowel.ch=='U')
    {
        printf("'%c' is Vowel.", vowel.ch);
    }
    else if((vowel.ch >= 'a' && vowel.ch <= 'z') || (vowel.ch >= 'A' &&
vowel.ch <= 'Z'))
    {
        printf("'%c' is Consonant.\n", vowel.ch);
    }
    else
    {
        printf("'%c' is not an alphabet.\n", vowel.ch);
    }
 return 0;
}
```

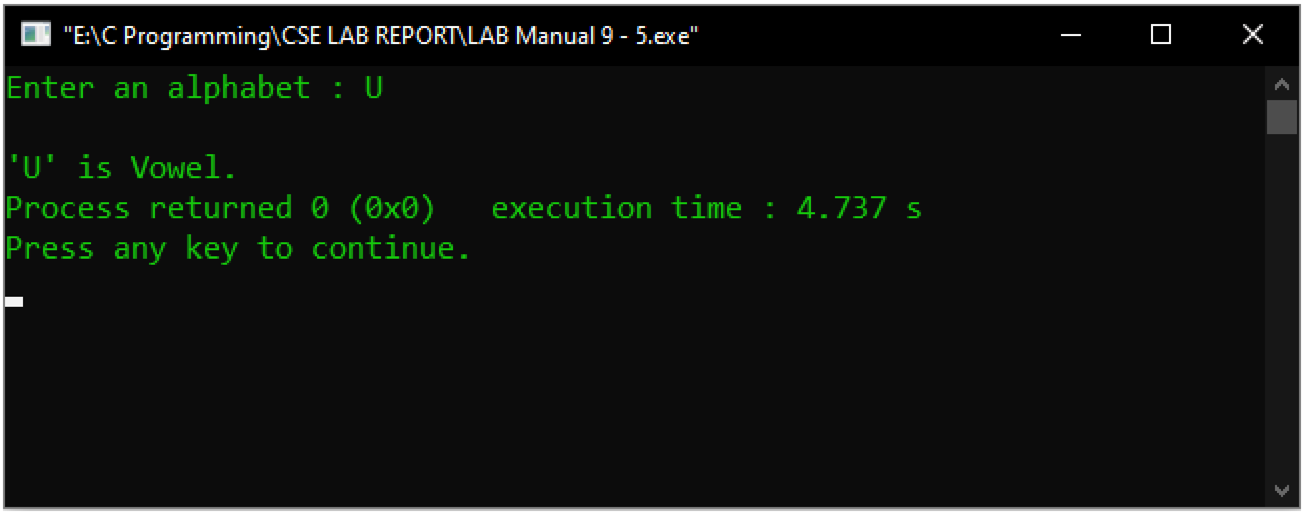
Output :

Part - 1



```
Enter an alphabet : G

'G' is Consonant.

Process returned 0 (0x0)   execution time : 1.468 s
Press any key to continue.
```

Part - 2



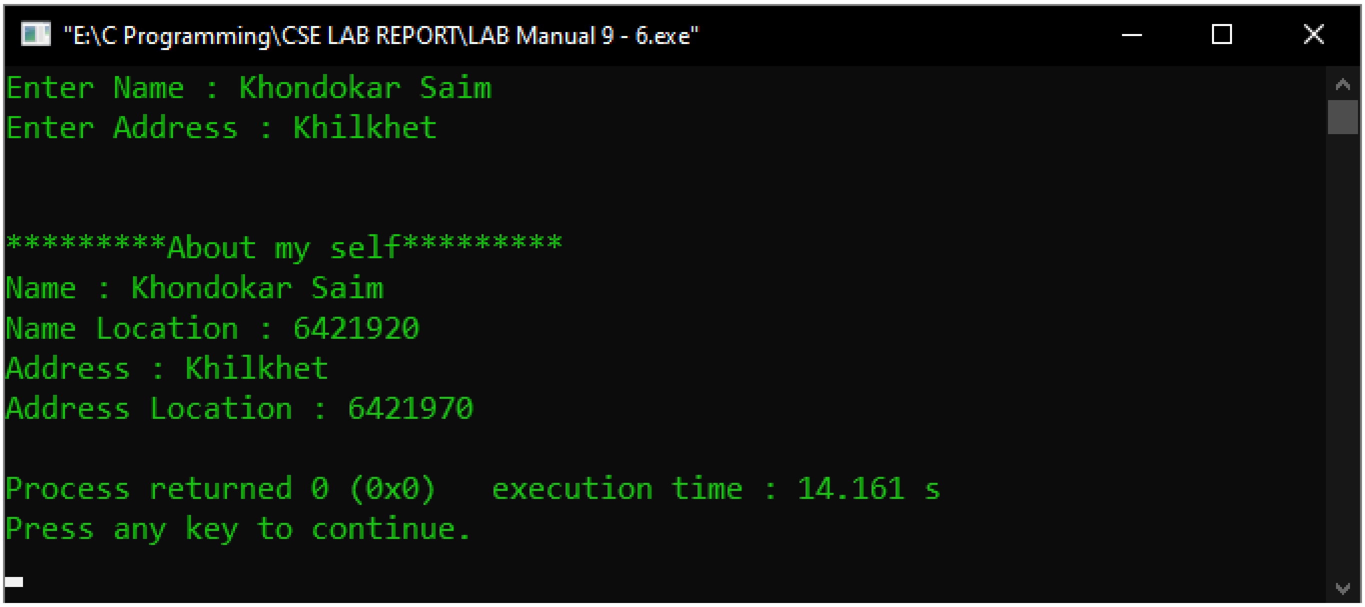
```
Enter an alphabet : U

'U' is Vowel.
Process returned 0 (0x0)   execution time : 4.737 s
Press any key to continue.
```

6. Write a C program using nested structure to print your name and address with location.

Code :

```
#include <stdio.h>
#include <string.h>
struct location
{
    int *namelocation;
    int *addlocation;
};

struct myinfo{

    char name[50];
    char address[50];
    struct location findLocation;

};

int main(){
    struct myinfo my;
    printf("Enter Name : ");
    fflush(stdin);
    gets(my.name);

    printf("Enter Address : ");
    fflush(stdin);
    gets(my.address);

    my.findLocation.namelocation = &my.name;
    my.findLocation.addlocation = &my.address;
    printf("\n\n");
    printf("*********About my self*********\n");
    printf("Name : %s\n", my.name);
    printf("Name Location : %d\n", my.findLocation.namelocation);
    printf("Address : %s\n", my.address);
    printf("Address Location : %d\n", my.findLocation.addlocation);

}
```

Output :



```
Enter Name : Khondokar Saim
Enter Address : Khilkhet


*********About my self*********
Name : Khondokar Saim
Name Location : 6421920
Address : Khilkhet
Address Location : 6421970

Process returned 0 (0x0)   execution time : 14.161 s
Press any key to continue.
```

• ANALYSIS AND DISCUSSION :

1. We got the exact result on output. Sometimes the result was wrong but we found the right implementation.

2. The problem of display anything in output is the easiest implementation. We solve that very easy.

3. In this assignment, we were faced some problem on last two question but by the teachers help we solve it.

4. Nothing is very/most difficult parts in my program to implement.

5. Structure is a user-defined datatype in C language which allows us to combine data of different types together

6. structure helps to construct a complex data type which is more meaningful. It is somewhat similar to an Array, but an array holds data of similar type only.

7. I learnt from structure –

   - Declaring Structure Variables
     i. Declaring Structure variables separately
     ii. Declaring Structure variables with structure definition

   - Accessing Structure Members
   - Structure Initialization
   - Array of Structure
   - Nested Structures

Green University of Bangladesh

Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering

Semester:2nd (Summer, Year:2022), B.Sc. in CSE (Day)

LAB REPORT NO: 10

Course Title: Structured Programming Lab

Course Code: CSE 104 / Section: DK

Lab Experiment Name: Pointers and Arrays .

Student Details

Name	ID
Khondokar Saim	221902353

Lab Date : 29 / 08 / 2022

Submission Date : 13 / 09 / 2022

Course Teacher's Name : Monoshi Kumar Roy

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status

Marks: ………………………………… Signature:....................

Comments:.. Date:.............................

- ## Tittle of the Lab experiment : **Pointers and Arrays .**

- ## Objectives :

The Pointer in C, is a variable that stores address of another variable. A pointer can also be used to refer to another pointer function. A pointer can be incremented/decremented, i.e., to point to the next/ previous memory location. The purpose of pointer is to save memory space and achieve faster execution time.

Types of Pointers in C -

- Null Pointer
- Void Pointer
- Wild pointer

Other types of pointers in C are as follows:

- Dangling pointer
- Complex pointer
- Near pointer
- Far pointer
- Huge pointer

What You Will Learn:

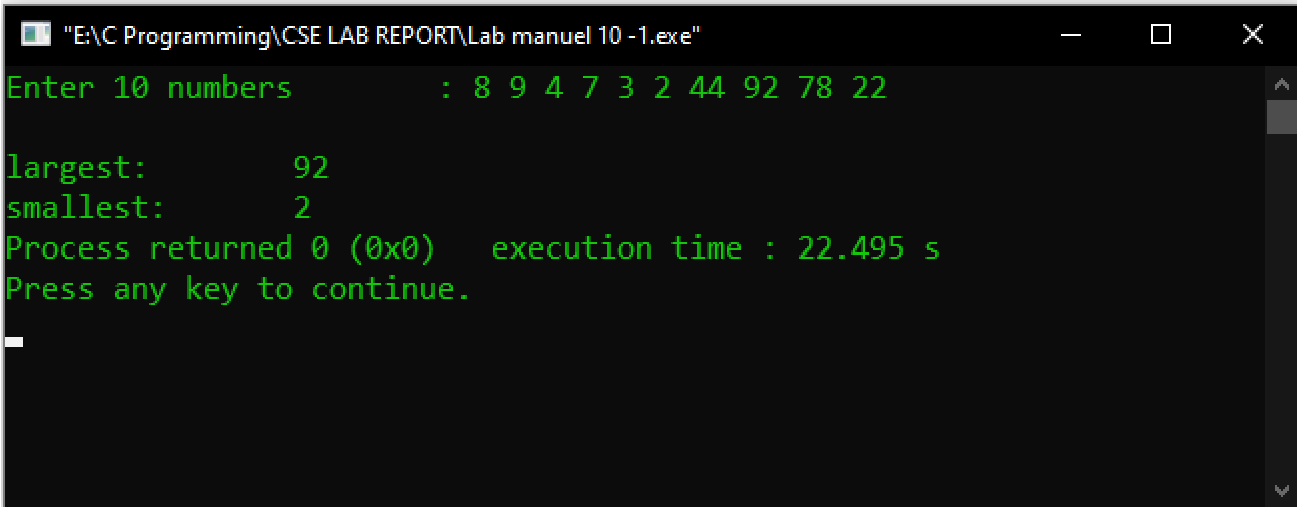
- Declaring a Pointer
- Initialize a pointer
- Direct and Indirect Access Pointers
- Pointer Arithmetic in C
- Priority operation (precedence)
- C Pointers & Arrays

1. Write a program in C to find the maximum and minimum element in an array using pointer

Code :

```
#include <stdio.h>
#define N 10

void max_min(int a[], int n, int *max, int *min);
int main(void)
{
int b[N], *i, big, small;

printf("Enter %d numbers\t:",N);
for(i=&b[0];i<&b[N];i++)
    scanf("%d",i);

max_min(b, N, &big, &small);

printf("\nlargest:\t%d",big);
printf("\nsmallest:\t%d",small);

return 0;
}
void max_min(int a[], int n, int *max, int *min)
{
int *i;
*max = *min = a[0];
for (i=&a[0]; i<&a[n]; i++)
{
    if(*i>*max)
    *max=*i;
    else if (*i<*min)
    *min=*i;
}}
```

Output :



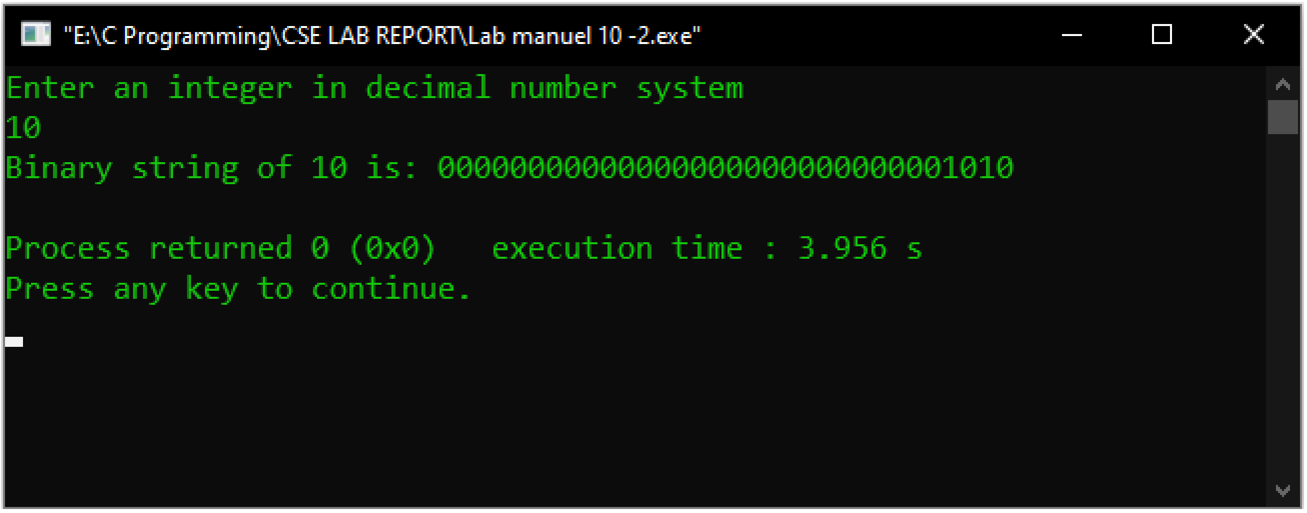
```
Enter 10 numbers        : 8 9 4 7 3 2 44 92 78 22

largest:        92
smallest:       2
Process returned 0 (0x0)   execution time : 22.495 s
Press any key to continue.
```

2. Write a C program to convert Decimal to Binary number system pointer.

Code :

```
#include <stdio.h>
#include <stdlib.h>
char *decimal_to_binary(int);

int main()
{

int n;
char *p;

printf("Enter an integer in decimal number system\n");
scanf("%d", &n);

p = decimal_to_binary(n);
printf("Binary string of %d is: %s\n", n, p);
 free(p);
 return 0;

}

char *decimal_to_binary(int n)

{
 int c, d, t;
 char *p;
 t = 0;
 p = (char*)malloc(32+1);
 if (p == NULL)
 exit(EXIT_FAILURE);
 for (c = 31 ; c >= 0 ; c--)
{
 d = n >> c;
 if (d & 1)
 *(p+t) = 1 + '0';
 else
 *(p+t) = 0 + '0';
 t++;

}
 *(p+t) = '\0';
 return p;

}
```

Output :



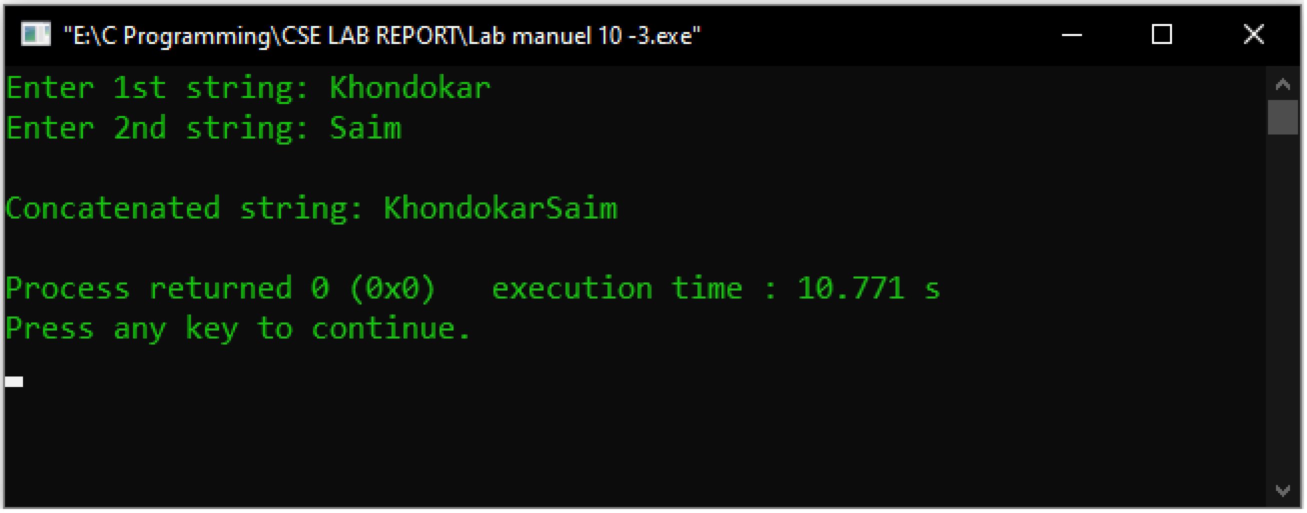
```
Enter an integer in decimal number system
10
Binary string of 10 is: 00000000000000000000000000001010

Process returned 0 (0x0)   execution time : 3.956 s
Press any key to continue.
```

3. Write a C program to concatenate two strings using pointers

Code :

```
#include <stdio.h>
#define MAX_SIZE 100

int main()
{
    char str1[MAX_SIZE], str2[MAX_SIZE];
    char * s1 = str1;
    char * s2 = str2;

    // Inputting 2 strings from user
    printf("Enter 1st string: ");
    gets(str1);
    printf("Enter 2nd string: ");
    gets(str2);

    // Moving till the end of str1
    while(*(++s1));

    // Coping str2 to str1
    while(*(s1++) = *(s2++));

    printf("\nConcatenated string: %s\n", str1);

    return 0;
}
```

Output :



```
Enter 1st string: Khondokar
Enter 2nd string: Saim

Concatenated string: KhondokarSaim

Process returned 0 (0x0)   execution time : 10.771 s
Press any key to continue.
```

4. Write a C program to add two matrix using pointers

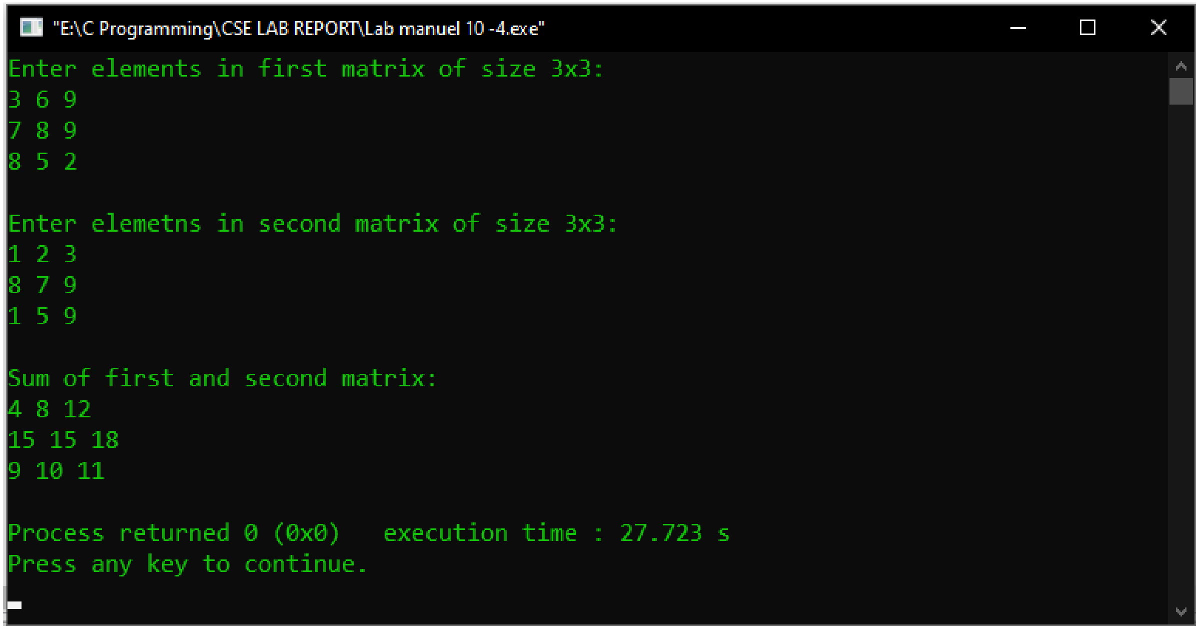
Code :

```
#include <stdio.h>
#define ROWS 3
#define COLS 3

void matrixInput(int mat[][COLS]);
void matrixPrint(int mat[][COLS]);
void matrixAdd(int mat1[][COLS], int mat2[][COLS], int res[][COLS]);


int main()
{
    int mat1[ROWS][COLS], mat2[ROWS][COLS], res[ROWS][COLS];

    printf("Enter elements in first matrix of size %dx%d: \n", ROWS, COLS);
    matrixInput(mat1);

    printf("\nEnter elemetns in second matrix of size %dx%d: \n", ROWS, COLS);
    matrixInput(mat2);

    matrixAdd(mat1, mat2, res);

    printf("\nSum of first and second matrix: \n");
    matrixPrint(res);


    return 0;
}
```

```
void matrixInput(int mat[][COLS])
{
    int i, j;


    for (i = 0; i < ROWS; i++)
    {
        for (j = 0; j < COLS; j++)
        {
            scanf("%d", (*(mat + i) + j));
        }
    }
}

void matrixPrint(int mat[][COLS])
{
    int i, j;

    for (i = 0; i < ROWS; i++)
    {
        for (j = 0; j < COLS; j++)
        {
            printf("%d ", *(*(mat + i) + j));
        }
        printf("\n");
    }
}

void matrixAdd(int mat1[][COLS], int mat2[][COLS], int res[][COLS])
{
    int i, j;

    for (i = 0; i < ROWS; i++)
    {
```

```
for (j = 0; j < COLS; j++)
        {
            *(*(res + i) + j) = *(*(mat1 + i) + j) + *(*(mat2 + i) + j);
        }
    }
}
```

Output :



"E:\C Programming\CSE LAB REPORT\Lab manuel 10 -4.exe"

```
Enter elements in first matrix of size 3x3:
3 6 9
7 8 9
8 5 2

Enter elemetns in second matrix of size 3x3:
1 2 3
8 7 9
1 5 9

Sum of first and second matrix:
4 8 12
15 15 18
9 10 11

Process returned 0 (0x0)   execution time : 27.723 s
Press any key to continue.
```

- ## ANALYSIS AND DISCUSSION :

1. We got the exact result on output. Sometimes the result was wrong but we found the right implementation.

2. The problem of display anything in output is the easiest implementation. We solve that very easy.

3. In this assignment, we were faced some problem on last two question but by the teachers help we solve it.

4. Nothing is very/most difficult parts in my program to implement.

5. All program is easy to understand and these helped me a lot to remove my confusion about pointer and array in c programming.

6. I learnt- Declaring a Pointer , Initialize a pointer, Direct and Indirect Access Pointers , Pointer Arithmetic in C Priority operation (precedence), C Pointers & Arrays on program and many basic things about c programming.