Green University of Bangladesh
Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering

Semester: (Spring, Year: 2022), B.Sc. in CSE (Day)

Course Title: STRUCTURE PROGRAMMING LAB

Course Code: 104 Section: DA

Lab Project Name: Library Management system.

Student Details

Name	ID
SAMIYA AKTER	213902044

Submission Date : 26/04/2022

Course Teacher's Name : Md. Solaiman Mia

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

Lab Project Status

Marks: …… Signature:.....

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

Table of Contents

Chapter 01
Introduction

1.1 Introduction

C is an imperative procedural language. It was designed to be compiled to provide low-level access to memory and language constructs that map efficiently to machine instructions, all with minimal runtime support. Despite its low-level capabilities, the language was designed to encourage cross-platform programming
Standards-compliant C program written with portability in mind can be compiled for a wide variety of computer platforms and operating systems with few changes to its source code. Since 2000, C has consistently ranked

1.2 **Objective**

It is an efficient programming language
The structure simplifies testing and debugging
It will be easier to learn other programming languages

Most programming languages can interface with it

Chapter 02

Design/Development /Implementation of the project

ALGORITHM 1:

step 1:start

step 2:The `#include` is a preprocessor command that tells the compiler to include the contents of the <stdio.h> (standard input and output) file in the program.

step 3:The `stdio.h` file contains functions such as `scanf()` and `printf()` to take input and display output respectively.

step 4:If you use the `printf()` function without writing `#include <stdio.h>`, the program will not compile.

step 5:The execution of a C program starts from the `main()` function.

step 6: `printf()` **is a library function to send formatted output to the screen. In this program,** `printf()` **displays** `Basic C programming!` **text on the screen.**

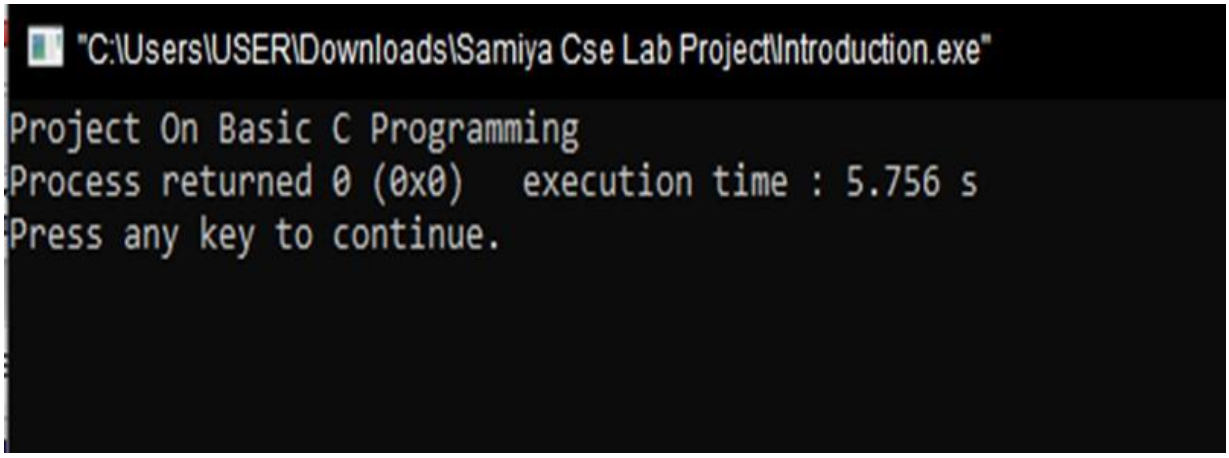
step 7: **The** `return 0;` **statement is the "Exit status" of the program. In simple terms, the program ends with this statement.**

step 8: **END**

CODE 1:

```
#include <stdio.h>
int main() {
  // printf() displays the string inside quotation printf(“Basic C Programming!"); return 0;
}
```

RESULT:



ALGORITHM 2:

Step 1:start
Step 2:Take a character as input
Step 3:Check if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z'))
Step 4:If the condition is true print the given character is an Alphabet.
Step 5:Else the character is not an Alphabet.
Step 6: END

CODE 2:

```
#include <stdio.h> int main() { char c; printf("Enter
a character: "); scanf("%c", &c); if ((c >= 'a' && c <=
'z') || (c >= 'A' && c <= 'Z')) printf("%c is an
alphabet.", c); else printf("%c is not an alphabet.",
c); return 0;
}
```

RESULT:

```
"C:\Users\USER\Downloads\Samiya Cse Lab Project\Decision Making.exe"

Enter a character: d
d is an alphabet.
Process returned 0 (0x0)   execution time : 8.845 s
Press any key to continue.
```

ALGORITHM 3:

Step 1: start

Step 2:Suppose the user entered 20.

Step 3:Initially, `addNumbers()` is called from `main()` with 20 passed as an argument.

Step 4:The number 20 is added to the result of `addNumbers(19)`.

Step 5:In the next function call from `addNumbers()` to `addNumbers()`, 19 is passed which is added to the result of `addNumbers(18)`. This process continues until `n` is equal to 0.

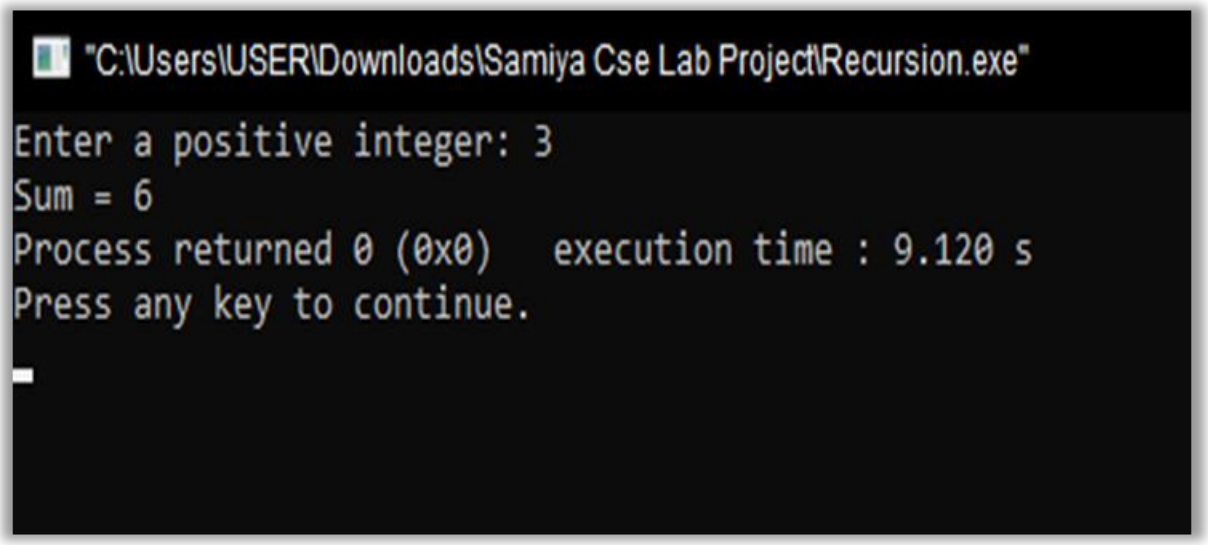
Step 6:When `n` is equal to 0, there is no recursive call. This returns the sum of integers ultimately to the `main()` function. Step 7: END.

CODE 3:

```
#include <stdio.h> int

addNumbers(int n); int

main() {
```

```
  int num;
  printf("Enter a positive integer: ");
  scanf("%d", &num); printf("Sum = %d",
  addNumbers(num));
  return 0;
}

int addNumbers(int n) {
  if (n != 0)
    return n + addNumbers(n - 1);
  else
    return n;
}
```

RESULT:



ALGORITHM 4:

Step 1:start

Step 2:In this program, the elements are stored in the integer array

data[].

Step 3:Then, the elements of the array are accessed using the pointer notation

Step 4:END.

CODE 4:

```
#include <stdio.h>

int main() { int

data[5];



    printf("Enter elements: ");

    for (int i = 0; i < 5; ++i)

    scanf("%d", data + i);



    printf("You entered: \n");

    for (int i = 0; i < 5; ++i)

    printf("%d\n", *(data + i));

    return 0;

}
```

RESULT:

```
"C:\Users\USER\Downloads\Samiya Cse Lab Project\pointer and arrays.exe"
Enter elements: 3
4
3
2
3
You entered:
3
4
3
2
3

Process returned 0 (0x0)   execution time : 17.242 s
Press any key to continue.
```

ALGORITHM 5:

Step 1: start

Step 2:Here, using a `for` loop, we iterate over characters of the string from `i = 0` to until `'\0'` (null character) is encountered. In each iteration, the value of `i` is increased by 1.

Step 3:When the loop ends, the length of the string will be stored in the `i` variable.

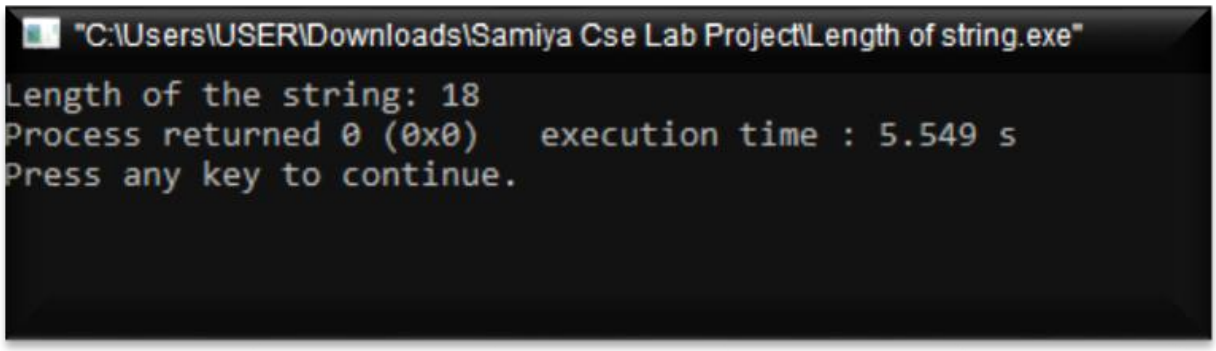
Step 4: END.

CODE 5:

```
#include <stdio.h> int main() { char
s[] = "Programming is fun"; int i;
for (i = 0; s[i] != '\0'; ++i);
```

```
    printf("Length of the string: %d", i);
    return 0;
}
```

RESULT:



Chapter 03

Conclusion

Practical implementation\Scope for future

In this article, we'll explain what C programming is, list ways that you can use it, detail just a few of the many benefits that can be gained from learning this foundational programming language, and provide a simple explanation of how C works.