CL118 Programming Fundamentals

Lab 07

Recursion & Arrays

Instructors: Ms. Maham Mobin Sheikh
Ms. Atiya jokhio

NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES

LAB 07

maham.mobin@nu.edu.pk|atiya.jokhio@nu.edu.pk

Learning Objectives

This lab will cover the following topics:

- Recursion
- Arrays
- 1D Array

Recursion

When a function invokes itself, the call is known as a recursive call. Recursion (the ability of a function to call itself) is an alternative control structure to repetition (looping). Rather than use a looping statement to execute a program segment, the program uses a selection statement to determine whether to repeat the code by calling the function again or to stop the process.

Flowchart for recursion:

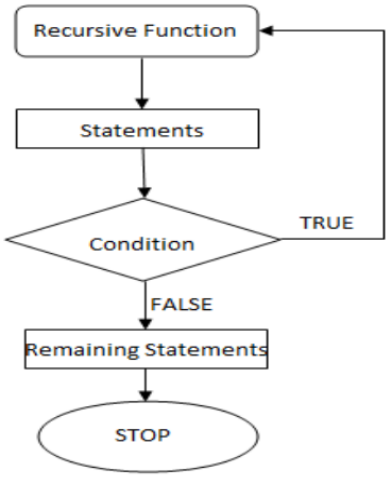
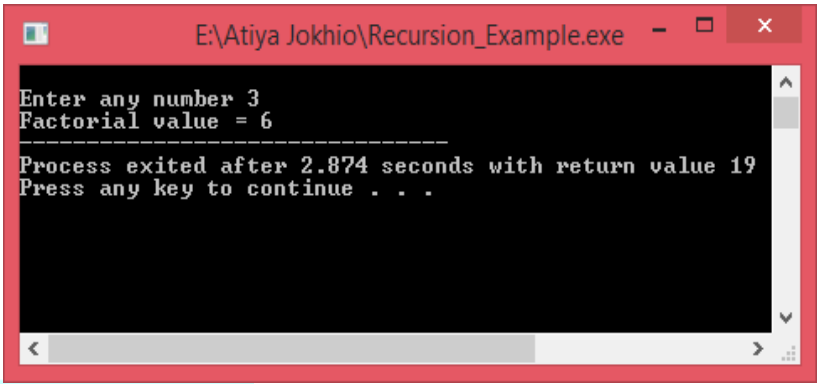


Fig: Flowchart showing recursion

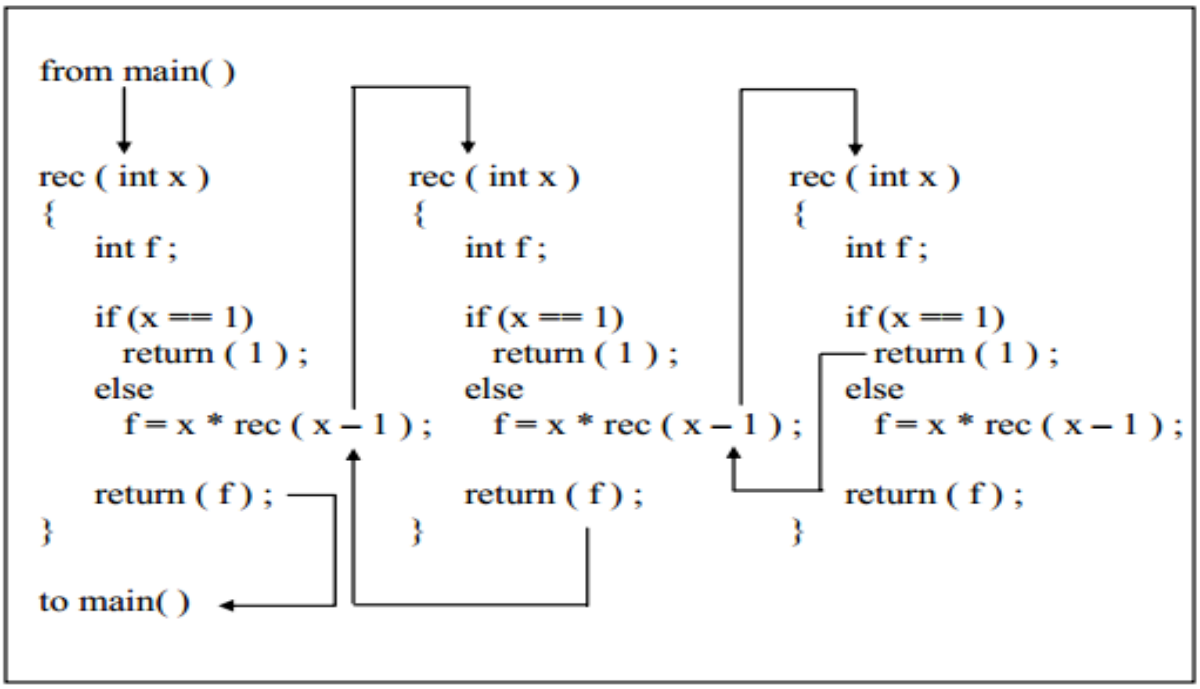
Each recursive solution has at least two cases: the base case and the general case.
The **base case** is the one to which we have an answer; the **general case** expresses the solution in terms of a call to itself with a smaller version of the problem. Because the general case solves a smaller and smaller version of the original problem, eventually the program reaches the base case, where an answer is known, and the recursion stops.
For example, a classic recursive problem is the factorial. The factorial of a number is defined as the number times the product of all the numbers between itself and 0: N! = N * (N-1)! The factorial of 0 is 1. We have a base case, Factorial (0) is 1, and we have a general case, Factorial (N) is N * Factorial (N-1). An if statement can evaluate N to see if it is 0 (the base case) or greater than 0 (the general case). Because N is clearly getting smaller with each call,

the base case is reached.
Following is the recursive version of the function to calculate the factorial value.

```
#include <stdio.h>
int main( )
{
    int a, fact ;
    printf ( "\nEnter any number " ) ;
    scanf ( "%d", &a ) ;
    fact = rec ( a ) ;
    printf ( "Factorial value = %d", fact ) ;
    }
    rec ( int x )
    {
    int f ;
    if ( x == 1 )
    return ( 1 ) ;
    else
    f = x * rec ( x - 1 ) ;
    return ( f ) ;
    }
```



Assume that the number entered through scanf( ) is 3. The figure below explains what exactly happens when the recursive function rec( ) gets called.



Disadvantages of recursion

- Recursive programs are generally slower than non-recursive programs. This is because, recursive function needs to store the previous function call addresses for the correct program jump to take place.
- Requires more memory to hold intermediate states. It is because, recursive program requires the allocation of a new stack frame and each state needs to be placed into the stack frame, unlike non-recursive (iterative) programs.

Arrays:

An array is a compound data type or a named collection of homogeneous items in which individual items are accessed by their place within the collection. The place within the collection is called an index. All the item in the array must be of the same type, but you can create arrays of integers, arrays of characters and the like.

Array Declaration:

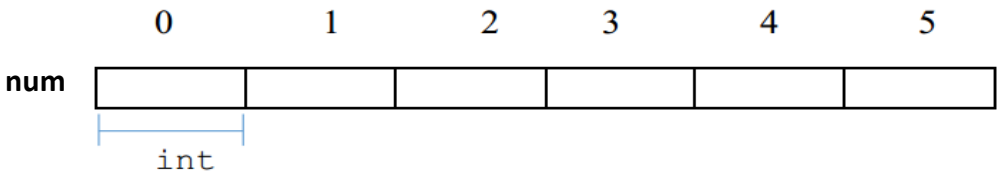
An array declaration specifies the name of the array, the type of elements stored in the array, and the number of array elements (the size of the array).

type name [elements];

For example, six values of type int can be declared as an array without having to declare 6 different variables (each with its own identifier). Instead, using an array, the six int values are stored in contiguous memory locations, and six can be accessed using the same identifier, with the proper index.

Int num[6];

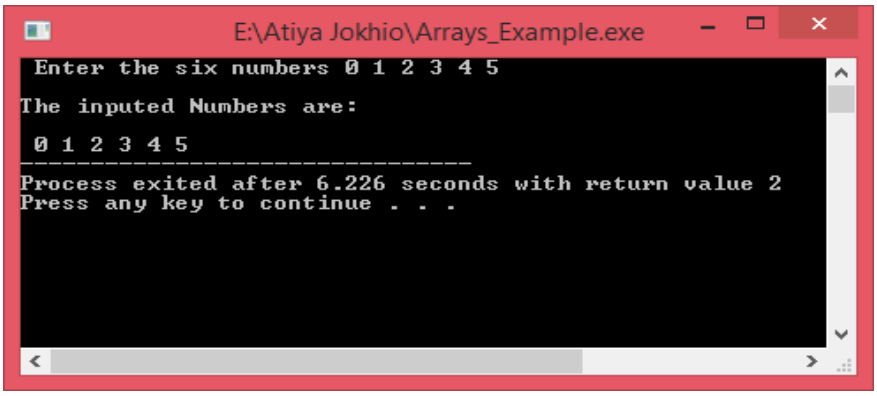
The declaration creates an array of integers named num and reserves enough memory to store six integers.



Where each blank panel represents an element of the array. In C, the first element in an array is always numbered with a zero (not a one), no matter its length. You refer to an individual array element by placing its number (called its subscript or array index) within brackets immediately after the array name. in the num array you denote the first element num[0], the second element num [1] , and so on.

// a program that reads six integers into array and displays them

```
#include <stdio.h>
void main(void)
{
int num[6]; // array of six numbers
int j;       // loop control index
// input six integers into the num array.
printf(" Enter the six numbers ");
for (j=0 ;j<6 ;j++)
scanf("%d",&num[j]);
//Display the numbers
printf("\nThe inputed Numbers are:\n\n ");
for (j=0 ;j<6 ;j++)
{
printf("%d ",num[j]);
}
}
```



In the program ,numbers are read by means of a loop in which the loop counter ,j , ranges from 0 to 5 .for each value of j, the statement

scanf("%d",&num[j]);

read in numbers and stores it in array element num[j].

Array Initialization

An array initialization specifies the name of the array, the type of elements stored in the array, and the number of array elements (the size of the array) and the elemets also.
Int num[6]={ 2, 4, 6, 8, 10};

Characters Array or Strings

A string constant is a one-dimensional array of characters terminated by a null (‘\0’). The terminator define end of string.
For example,
char name[5] = { ‘F’, ‘A’, ‘S’, ‘T’, ‘\0’ };
char name[5]= “FAST”;

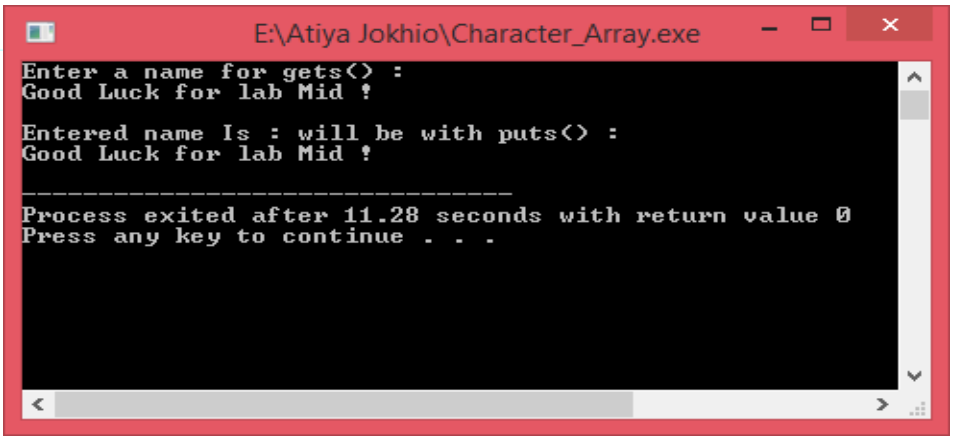
Input function scanf() can be used with %s format specifier to read a string input from the terminal. But there is one problem with scanf() function, it terminates its input on the first white space it encounters. Therefore if you try to read an input string "Hello World" using scanf() function, it will only read Hello and terminate after encountering white spaces.

The String I/O Function gets() & puts()

Sanf() and printf() is not versatile for string I/O we can use gets() and puts() function from stdio library.

For Example:

```c
#include<stdio.h>
void main(void)
{
char name[20];
printf("Enter a name for gets() :");
//get string input using gets() function
gets(name);
printf("Entered name Is : will be with puts() :");
//print string using puts() function
puts(name);
}
```

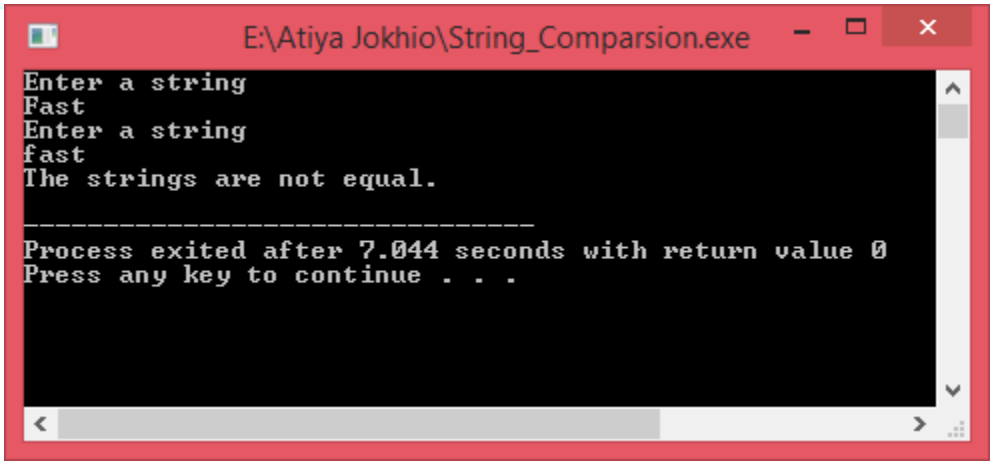


C supports a wide range of functions that manipulate null-terminated strings –

Sr.No.	Function & Purpose
1	**strcpy(s1, s2);** Copies string s2 into string s1.
2	**strcat(s1, s2);** Concatenates string s2 onto the end of string s1.
3	**strlen(s1);** Returns the length of string s1.
4	**strcmp(s1, s2);** Returns 0 if s1 and s2 are the same; less than 0 if s1<s2; greater than 0 if s1>s2.

Example:

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

int main()
{
   char str1[100], str2[100];

   printf("Enter a string\n");
   gets(str1);

   printf("Enter a string\n");
   gets(str2);

   if (strcmp(str1,str2) == 0)
      printf("The strings are equal.\n");
   else
      printf("The strings are not equal.\n");

   return 0;
}
```



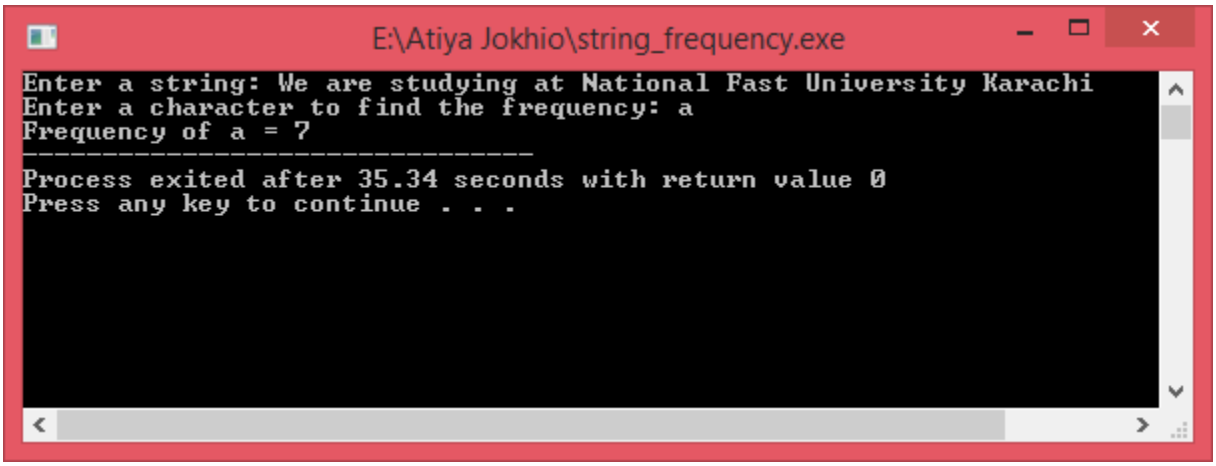
Lab Activity

Task# 01

Searching has different techniques when it comes to programming. Finding any certain value from a given range can be done by the help of different algorithms. Generate a random list consisting of 10 numbers and search for a value taken from user. Stop only when user finds specified value. Also mention at which position it was found.

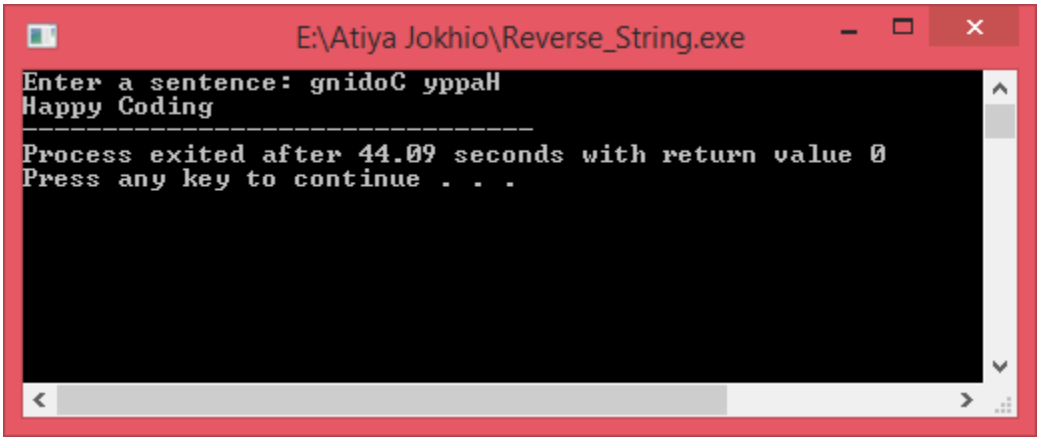
Task# 02

Write a program in c to show following output



Task# 03

Bit wants to send his message (Password of Account) to byte, in order to avoid that no one can understand his message he used the codewords. He only says to byte whatever will be written in given sentence, if you flip it right to left that will be its answer. Write a program in c (as displayed in output) using recursion.



Task# 04

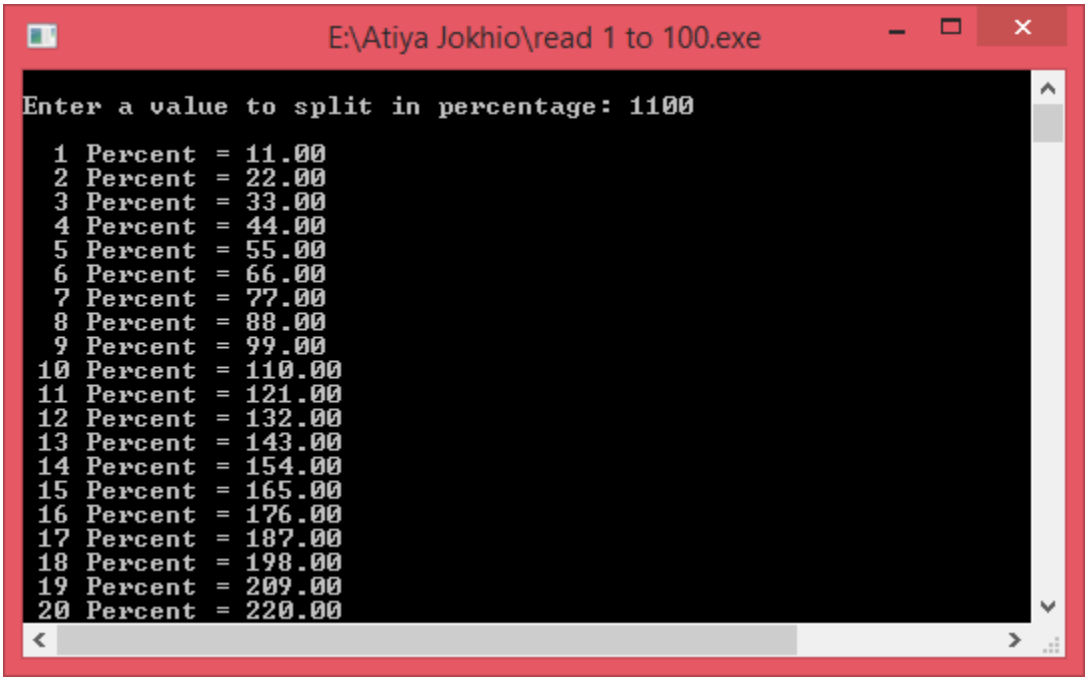
Waheed Ali, a student of 2nd year studying in university is suffering from fever since last week. Doctor wants to diagnose his temperature values for whole week. Write a program in C that inputs the temperature values of entire week and displays the day on which he had maximum and minimum temperature.

Task# 05

Write a program in C that inputs the elements of an integer array containing 08 numbers from the user and displays the number of positive, negative and zero values and also find the sum of inputted positive and negative numbers present in an array.

Task# 06

Write a program in C to read and print its corresponding percentage from 1% to 100% using recursion.



Task# 07

Yasir Waheed is working in an organization and he gets the different salary in every month. Following table shows his salary each month.

Month	Per Month Salary (in thousands)
January	50K
February	70K
March	55K
April	66K
May	67K
June	72K

He has hired you so that you can make a C program for him in which he can find

i. Sum of total salary of all the months.
ii. Sort the salary in ascending order w.r.t. amount.
iii. Average of the salary of six months

Task# 08

Write a program in c to count digits of inputted number by user using recursion.

```
Enter a positive integer number: 123
Total digits in number 123 is: 3
```