**SUBJECTIVE ASSIGNMENT OF ARRAY & STACK**

1. Write a c program to find and display Largest Number in an array A[] of size n ,by using function named 'largest' having two argument first array and second size of an array and this function will return largest number of array A[] to main function, and main will display largest number of array.

( Hint: int largest(int[],int); //prototype of largest function.)

SAMPLE INPUT:

n=5;

A[]={10,30,20,60,50};

SAMPLE OUTPUT:

Largest number in given array is: 60

1. Write a program to print the Next Greater Element (NGE) for every element of an array. The Next greater Element for an element x is the first greater element on the right side of x in the array. Elements for which no greater element exist, consider the next greater element as -1.

Remember:

For an array, the rightmost element always has the next greater element as -1.

For an array that is sorted in decreasing order, all elements have the next greater element as -1.

For the input array [4, 5, 2, 25], the next greater elements for each element are as follows.

Element NGE

4 --> 5

5 --> 25

2 --> 25

25 --> -1

SAMPLE INPUT :

arr[] = {11, 13, 21, 3};

SAMPLE OUTPUT:

11 --> 13

1. --> 21

3 --> -1

1. --> -1

1. Modify the above program to find and print the Next Greater Element (NGE) for all array element using Stack
2. Write a program to calculate prefix sum of array where Given an array ‘A’ of size n, its prefix sum array is an array of the same size n such that the i’th element of the prefix sum array ‘Prefix’ is the sum of all elements of the given array till i’th index from the beginning, i.e Prefix[i] = A[0] + A[1] + A[2] + … +A[i].

For Example: Given A[] = [3, 4, -1, 2, 5],

The prefix sum array P[] is given as –

P[0] = 3, P[1] = 7, P[2] = 6, P[3] = 8, P[4] = 13

i.e. P[] = [3, 7, 6, 8, 13]

1. Write a C program to Reverse a string using Stack For example “SCABHOPAL” should be converted to “LAPOHBACS”.

Following is simple algorithm to reverse a string using stack.

* Create an empty stack.
* One by one push all characters of string to stack.
* One by one pop all characters from stack and put them back to string.

1. Convert given Infix notation to postfix notation:

a. A+B\*C/D$E

b. A\*B$C/D\*E+F

c. 3\*5/3-1+8

d. 2$3 /2+1

1. Convert above infix notation to prefix notation

a. A+B\*C/D$E

b. A\*B$C/D\*E+F

c. 2\*5/5 -2+8

d. 3$ 2/3 +1

1. Write a c program to evaluate postfix expression which is

entered through keyboard

SAMPLE INPUT: 34\*2/5+2-8+

SAMPLE OUTPUT: 17.000000

1. Write a c program to evaluate prefix expression which is entered through keyboard.

SAMPLE INPUT: -+\*/$323266

SAMPLE OUTPUT: 6.000000

1. Write a c program to convert infix expression to postfix

expression using sack.

SAMPLE INPUT: A\*B$C–D/E+F\*G

SAMPLE OUTPUT: ABC&\*DE/-FG\*+

1. Write a c program to convert infix expression to prefix

expression using sack.

SAMPLE INPUT: A\*B/C$D+E-F\*G

SAMPLE OUTPUT: -+/\*AB$CDE\*FG