Lab → 1/02/2024

Q.1) Write a program to simulate the working of stack using an array with:

a.) Push
b.) Pop
c.) Display

The program should print message for overflow & underflow.

Ans 2)

```
~~char~~ int stack [20]
~~#include <stdio.h>~~
void push (int);
void pop ();
void display ();
int top = -1;
void main ()
{
        push (50);
         push (22);
          display ();
           push (21);
            pop ();
            pop ();
            display ();
}           pop ();
void push (int x)
{
     top ++;
     ~~stack [top] = x~~
     if ( top == 2)
     printf ("Overflow");
     stack [top] = x;
```

```
void pop()
{        if (top! = -1)
       { int p = stack[top]
         printf("%d ", p);
          top -- ;
       }
       else
       {    printf("underflow");
       }
}
void display ()
{
       int i ;
       for (i = top ;  i >= 0 ; i - -)
        {       printf("%d \n", stack[i]);
        }
}
```

Output -:
~~Output~~
22
50
Overflow
Underflow

Date _______ Page _______

Q.2.) Write a program to convert a given valid paranthesized infix arithmetic expression to postfix expression. The expression consists of single character operands and the binary operators + (plus), - (minus), * (multiply) and / (divide).

Ans →

```
char infix[20], post[20], stack[20];
int top = -1;    char result;    int p=0;
void push(char);
char pop();
int order(char);
void main()
{          i=0
    print(Enter infix)
    int l= strlen(infix)
    while (i<l)
    {
        symb = infix[i]
        switch(symb)
        {
            case '(':
                push(symb)
                break;
            case ')':
                while(Stack[top]!='(')
                {
                    result = pop()
                    post[p++] = result
                }
                top--;
                break;
```

```
case '+' :
case '-' :
case '^' :
case '*' :
case '/' :
        while(order(symb) < order(stack[top]))
        {
            [illegible]
            result = pop()
            ~~post[p++] = result~~
            ~~push(symb);~~
        }
        if( order(symb) > order(stack[top])
        { push(symb)
        }
      else if (order(symb) <= order(stack[top])
              && (order(symb) == 1 )
              || (order(symb) == 2 ) )
      {
            result = pop()
            post[p++] = result
             push(symb)
      }
      else if (order(symb) == order(stack[top]
              && order(symb) == 3 )
      {     push(symb)
      }
      break;
}
i++
}
while ( top >= 0 )
{      post[p++] =  stack[top];
```

```
void push(char x)
{
        top++;
        stack[top] = x
}

~~void pop()~~
~~{~~
char pop()
{
        char re = stack[top];
        top--;
        return re;
}

int order(char symb)
{                                   int j;
        switch(symb)
        {
                case '+':
                case '-':
                          j = 1

                case '^':
                          j = 3

                case '*':
                case '/':
                          j = 2
        }
}
```

[illegible] 1/1/24

Output:-

```
Enter infix :    A*B+C*D-E
Postfix is       AB*CD*+E-
```