Lab-> 2 Infix to Postfix. 2024/01/01 Monday

* write a program to convert an infix exp into its equivalent postfix notation.

```
#include <stdio.h>
#include<conio.h>
#include<ctype.h>
#include<string.h
#define MAX 100
 char st[MAX];
 int top = -1;
void push (char st[];char),
 char pop(char st[]);
 void infix topostfix (char source[], char target[]);
 int (ea priority(char);
 int main()
{
    char infix[100], postfix[100];
    Cursor ();
    printf("\n Enter any infix expression:");
    gets(infix);
    strcpy(postfix," ");
    infix to prefix (infix, postfix);
    printf("\n The corresponding postfix
          expression is:");
    puts (postfix);
    getch();
    return 0;
}
  void infix to postfix (char source[]; char target[])
{
    int i=0, j=0;
    char temp;
    strcpy (target," ");
    while (source[i]!='\0')
```

```
{
    if(source[i]=='(')
    {
        push(st, source[i]);
        top++;
    }
    else if(source[i]==')')
    {
        while(top!=-1 && (st[top]!='('))
        {
            target[j]=pop(st);
            j++;
        }
        if(top==-1)
        {
            printf("\n Incorrect Exp");
            exit(1);
        }
        temp = pop(st); // remove left parenthesis
        i++;
    }
    else if(isdigit(source[i]) || isalpha(
            source[i]))
    {
        target[j]=source[i];
        j++;
        i++;
    }
    else if(source[i]=='+' || source[i]=='-'
    || source=='*' || source[i]=='/' ||
        source[i]=='%')
        {
    while(top!=-1 && st[top]!='(' &&
    (getpriority(st[top]) >= getpriority(source
                                    [i])))
```

```
{
    target[j]=pop(st);
    j++;
  }
  push(st, source[i]);
  i++;
 }
 else
 {
  printf("\n INCORRECT ELEMENT IN EXPRESSION");
  exit(1);
 }
}
while((top!=-1)&&(st[top]!='('))
{
   target[j]=pop(st);
   j++;
}
target[j]='\0';
}
int getpriority(char op)
{
 if(op=='/'||op=='*'||op=='%')
    return 1;
 else if(op=='+'||op=='-')
    return 0;
}
void push(char st[], char val)
{
 if(top==MAX-1)
    printf("\n STACK OVERFLOW");
 else
 {
    top++;
    st[top]=val;
 }
}
char pop(char st[])
{
   char val=' ';
   if(top==-1)
```

```
        printf("\n STACK UNDERFLOW");
    else
    {
        val=st[top];
        top--;
    }
    return val;
}
```

Out-put:

~~Enter any infix expression : (a+b/c*(d+e)~~

~~The postfix expression is~~

1) Enter any infix exp - (a+b/c*(d+e)-F)
   The postfix expression is = abc/de+*+F-

2) Enter any infix exp -> A*(B+C)
   The postfix exp is -> ABC*+

*write a program to simulate the working of a queue of integer using an array provide insert, delet + display operations

```
#include<stdio.h>
#define N 5
int q[N];
int front=-1, rear=-1;
void insert(int);
int delet();
void display();
void main()
{
   int n, choice;
   do
   {
    printf("\n1. Insert\n2. Delet\n3. Display\n4.
            Exit\n");
    printf("Enter your option:\n");
    scanf("%d", &choice);
    switch(choice)
    {
      case 1:
         printf("Enter the number to be inserted
           into queue.\n")
         scanf("%d", &n);
         insert(n);
         break;
      case 2:
         n=delet();
         if(n!=-1)
           printf("\n The number deleted is
                  : %d\n", n);
         break;
```

```
case 3:
    display();
    break;
case 4:
    exit(0);
    break;
default
    printf("Invalid option\n");
    exit(0);
    break;
    }
}
while (choice != 4);
}
void insert(int num)
{
    if (rear == N-1)
    printf("Overflow");
else if (front == -1 && rear == -1)
    front = rear = 0;
else
    rear++;
    q[rear] = num;
}
int delet()
{
    int val;
    if (front == -1 || front > rear)
    {
    printf("Underflow");
    return -1;
    }
else
    { val = q[front];
    front++;
```

```
    if(Front>Rear)
      Front=Rear=-1;
      return val;
    }
}
void display()
{
    int i;
    printf("\n");
    if(Front==-1 || Front>rear)
      printf("\n Queue is empty");
    else
    {
      for(i=Front;i<=rear;i++)
      printf("\t %d",q[i]);
    }
}
```

output:

1) Insert 2) Delet 3) Display 4) Exit.

Enter your option : 1

Enter the number to be inserted in the queue: 7

Queue

1 | 1 | 24