Date:

LAB 2

Page N

```
#include <stdio.h>
#define MAX 100
char stack[MAX]
int top = -1;

Void push(char ch)
{
if (top ==MAX-1)
printf("stack is full\n");
 else
 {
  top++;
  stack[top]=ch;
 }
}
char pop()
{
char item;
if (top == -1)
printf("\n Stack is empty!");
else
{ item = stack[top];
top--;
return item;
} }
```

te:

Page No.

```
int stackempty()
{
if (top==-1) return 1;
else return 0;
}
char stacktop()
{
if (top==-1)
printf ("\n stack is empty!");
else
return stack[top];
}
int priority (char ch)
{ switch (ch)
{
 case '+':
 case '-': return(1);
 case '*':
 Case '/': return(2);
 default: return (0);
 } }
 int main (int argc, char**argV)
 {
  char infix[100];
  Char postfix[100];
```

Date:

Page No.

```
int i, item;
printf("Enter the Infix Expression:");
Scanf("%S", infix);
printf("Expression: %S", infix);
printf("\n Postfix: ");
i=0;
int K=0;
int a,b;
int invalid =0;
while(infix[i]!='\0')
{
if(infix[i+1] =='\0')
{
if((infix[i]=='*'|| infix[i]=='+'|| infix[i]=='/'||
  infix[i]=='-'))
invalid =1;
}
if(infix[i]=='(')
{ push(infix[i]);
 a=i;
}
else if(infix[i]==')')
{ while((item=pop())!='(')
{
postfix[K] = item;
```

Date:

Page No.

```
 k++;
}
b=i;
int q, ot-count, op-count;
for(q=a+1; q<b; q++)
{
if (infix[q]=='*'|| infix[q]=='+'|| infix[q]=='/'||
 infix[q]=='-')
   ot-count++;
 else
  op-count++;
}
if(ot-count >=op-count)
 invalid=1;
 }
else if (infix[i]=='*'|| infix[i]=='+'|| infix[i]=='/'||
infix[i]=='-')
{
while(!stackempty()&&priority(infix[i])<=priority
(stacktop()))
{
 item=pop();
 postfix[k]=item;
 k++;
}
```

Date:

Page No.

```
push(infix[i]);
 }
else
{
 postfix[k] = infix[i];
 K++;
}
i++;
}
while(!stackempty())
{
char item;
item = pop();
postfix[k] = item;
K++;
}
int j=0;
if(invalid==1)
printf("Invalid expression\n");
else{
for( j=0; j<k; j++)
printf("%c", postfix[j]);
}
printf("\n");
return 0;
}
```