LAB PROGRAM 2:-

```
#include<stdio.h>

#include<string.h>

#include<stdlib.h>


int F(char symbol)

{

switch(symbol)

{

 case'+':

 case'-':return 2;

 case'*':

 case'/':return 4;

 case'^':

 case'$':return 5;

 case'(':return 0;

 case'#':return -1;

 default:return 8;

 }

}


int G(char symbol)

{

switch(symbol)

  {

 case'+':

 case'-':return 1;
```

```
 case'*':
 case'/':return 3;
 case'^':
 case'$':return 6;
 case'(':return 9;
 case')':return 0;
 default:return 7;
  }
}

void infix_postfix(char infix[],char postfix[])
{
int top,i,j;
char s[30],symbol;
top=-1;
s[++top]='#';
j=0;

for(i=0;i<strlen(infix);i++)
 {
symbol=infix[i];
while(F(s[top])>G(symbol))
{
postfix[j]=s[top--];
j++;
}
```

```
if(F(s[top])!=G(symbol))
s[++top]=symbol;
else
top--;
}

while(s[top]!='#')
{
postfix[j++]=s[top--];
}
postfix[j]='\0';
}

void main()
{
char infix[20];
char postfix[20];
printf("Enter the valid infix expression ");
scanf("%s",infix);
infix_postfix(infix , postfix );
printf("The postfix expression is \n");
printf("%s\n",postfix);
}
```

SCREENSHOT OF PROGRAM AND OUTPUT:-

RepLit - PastelCluelessAlphatest

repl.it/repls/PastelCluelessAlphatest#main.c

@anonymous / PastelCluelessAlphat...

Run

Talk

Sign up

Files

main.c

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

int F(char symbol)
{
switch(symbol)
{
 case'+':
 case'-':return 2;
 case'*':
 case'/':return 4;
 case'^':
 case'$':return 5;
 case'(':return 0;
 case'#':return -1;
 default:return 8;
 }
}

int G(char symbol)
{
```

```
> clang-7 -pthread -lm -o main main.c
> ./main
Enter the valid infixa^b*c-d+e/f/(g+h)
The postfix expression is
ab^c*d-ef/gh+/+
>
```

Type here to search

ENG 21:19 31-10-2020

RepLit - PastelCluelessAlphatest

repl.it/repls/PastelCluelessAlphatest#main.c

@anonymous / PastelCluelessAlphat...

Run

Talk

Sign up

Files

main.c

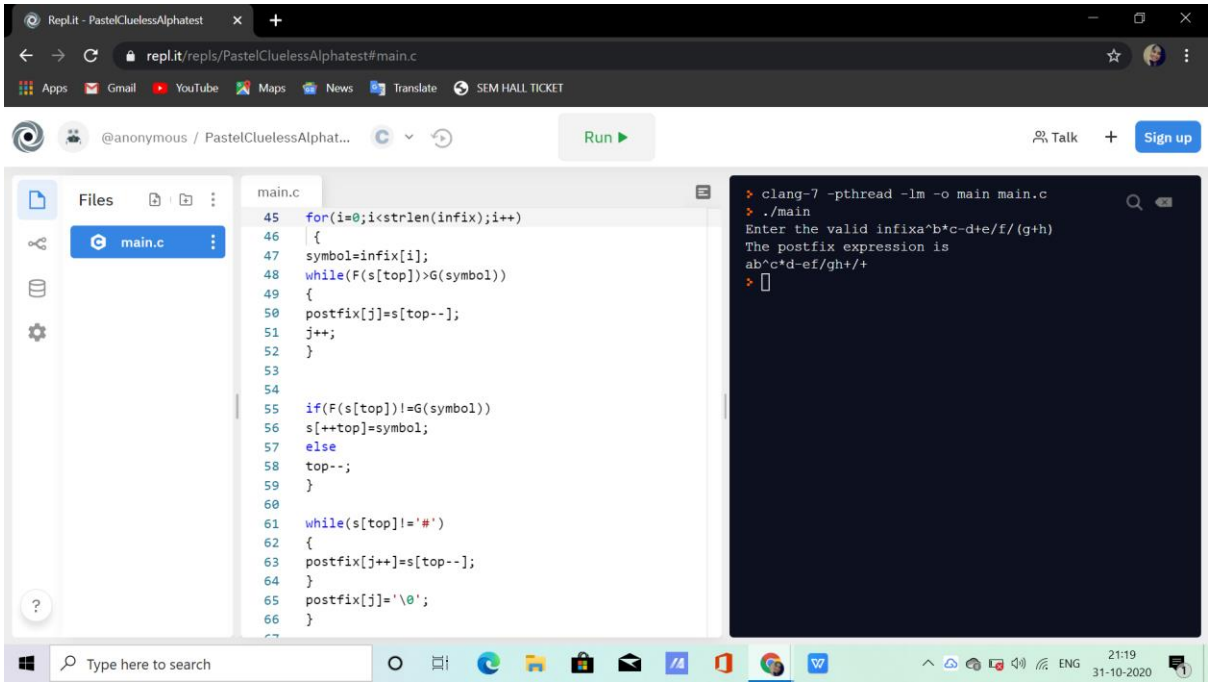
```c
switch(symbol)
  {
 case'+':
 case'-':return 1;
 case'*':
 case'/':return 3;
 case'^':
 case'$':return 6;
 case'(':return 9;
 case')':return 0;
 default:return 7;
  }
}

void infix_postfix(char infix[],char postfix[])
{
int top,i,j;
char s[30],symbol;
top=-1;
s[++top]='#';
j=0;
```

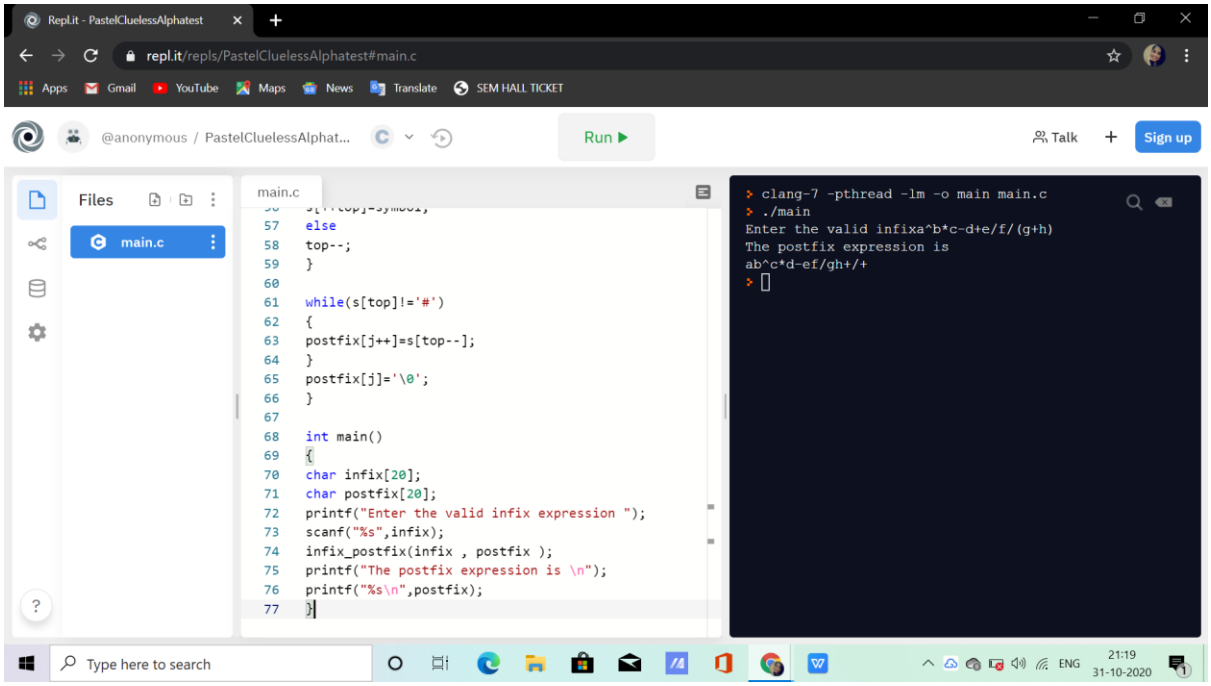
```
> clang-7 -pthread -lm -o main main.c
> ./main
Enter the valid infixa^b*c-d+e/f/(g+h)
The postfix expression is
ab^c*d-ef/gh+/+
>
```

Task View

Type here to search

ENG 21:19 31-10-2020


RepLit - PastelCluelessAlphatest
repl.it/repls/PastelCluelessAlphatest#main.c
@anonymous / PastelCluelessAlphat...
Run
Talk
Sign up
Files
main.c
for(i=0;i<strlen(infix);i++)
{
symbol=infix[i];
while(F(s[top])>G(symbol))
{
postfix[j]=s[top--];
j++;
}
if(F(s[top])!=G(symbol))
s[++top]=symbol;
else
top--;
}
while(s[top]!='#')
{
postfix[j++]=s[top--];
}
postfix[j]='\0';
}
clang-7 -pthread -lm -o main main.c
./main
Enter the valid infixa^b*c-d+e/f/(g+h)
The postfix expression is
ab^c*d-ef/gh+/+
Type here to search
ENG
21:19
31-10-2020


RepLit - PastelCluelessAlphatest
repl.it/repls/PastelCluelessAlphatest#main.c
@anonymous / PastelCluelessAlphat...
Run
Talk
Sign up
Files
main.c
else
top--;
}
while(s[top]!='#')
{
postfix[j++]=s[top--];
}
postfix[j]='\0';
}
int main()
{
char infix[20];
char postfix[20];
printf("Enter the valid infix expression ");
scanf("%s",infix);
infix_postfix(infix , postfix);
printf("The postfix expression is \n");
printf("%s\n",postfix);
}
clang-7 -pthread -lm -o main main.c
./main
Enter the valid infixa^b*c-d+e/f/(g+h)
The postfix expression is
ab^c*d-ef/gh+/+
Type here to search
ENG
21:19
31-10-2020

WRITTEN:-

PAGE NO.
DATE: / /

LAB-2

Qn:- WAP to convert a given valid parinthesized infix arithematic expression to postfix expression. The expression consists of single character operands and the binary operators + (plus), - (minus), * (multiply) and / (divide).

① (A + (B - C) * D)
② A+B.
③ A ^ b * c - d + e / f / (g+h).
④ x $ y $ z - m + n + p / q
⑤ a $ b * c - d + e / f / (g+h)

Program:-

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

int F(char symbol)
{
 switch (symbol)
 {
 case '+':
 case '-': return 2;
 case '*':
 case '/': return 4;
```

```
case '^':
case '$' : return 5;
case '(' : return 0;
case '#' : return -1;
default : return 8;
  }
}

int G(char symbol)
{
switch(symbol)
  {
case '+' :
case '-' : return 1;
case '*' :
case '/' : return 3;
case '^' :
case '$' : return 6;
case '(' : return 9;
case ')' : return 0;
default : return 7;
   }
}
```

PAGE NO.
DATE: / /

```
void infix_postfix (char infix[], char postfix[])
{
int top, i, j;
char s[30], symbol;
top = -1;
 s[++top] = '#';
 j=0;

for (i=0; i<strlen (infix); i++)
{
 symbol = infix[i];
 while (F(s[top]) > G(symbol))
  {
 postfix[j] = s[top--];
  j++;
  }

 if (F(s[top]) != G(symbol))
  s[++top] = symbol;
  else
 top--;
}
 while (s[top] != '#')
 {
```

PAGE NO.
DATE: / /

```
postfix[j++] = s[top--];
}
postfix[j] = '\0';
}
void main()
{
char infix[20];
char postfix[20];
printf("Enter the valid infix expression");
scanf("%s", infix);
infix_postfix(infix, postfix);
printf("The postfix expression is \n");
printf("%s \n", postfix);
}
```