Name: Pratik Rameshwar Shinde

Roll No.: 2274

Class: SE-IT

Div: B Batch : B-3

Subject: DSA LAB

Practical 2: Stack

Aim: Write a program to implement stack as an abstract data type using linked list and use this ADT for conversion of infix expression to postfix, prefix and evaluation of postfix/prefix expression.

****************************PROGRAM****************************

```
#include<iostream>

#include<string>            using

namespace std;class stack


typed ef struct node


char  data;  struct

node *next;

) node; node

*top=NULL;

public:
```

```
int     is_empty();
void     display();
int  count();  void
push(char       x);
char  pop();  char
get_top();

//get       top
char stack::

return top->data;

//display           void
stack::display()
for(node *P=top;P!=NULL;P=P->next)
{
cout<<"\n --------";cout<<"\n[ "<<P->data<<" ]";
}
cout<<"\n --------";
}
//push
void stack::push(charx)

node  node; if(St)
```

```
{
St->data=x;
St-snext=top;
top=St;

else "\nStackis full\n";

Vis empty
intstack::i

return(top=NULL) ;

//pop          char
stack::pop()
node *St=top;
top=top->next;
char       a—St-
>data;     delete
St; return(a);

int result(int sl ,int s2,char optr)
```

```
switch(optr)

case :
return(sl+s2)
; case -:
return(sl-s2);
case :return(s
I *s2); case
'T: return(s
l/s2); ease 'N:
return(s I
RS2);
default:
return O;



int priority(char a)

switch(a)

case :
case ' ':
return(
l); case:
 :



case
```

```
return(2)
; case 'A
': case :
ease !:
return(
l);
default:
return(O); void


inftopre(char ins [] )


stack s,k;
int i,x; //to
reverse
for(i=();ins[i] ;
i--;
cout<<"\n\n";for(x=i;x>=0;x--)switch(ins[x])
{
case 'Y:
s.push(ins [x]);break; case 'C: while(!s.is_empty() &&
s.get_top()!=')' )k.push(s.pop());
s.pop()
; break;
case :
case - :
case '
*':ease :
case :
while(!
s.is_em
pty()
priority
(s.get_t
op())>p
riority(
ins[x]))
k.push(s.pop());
s.push(ins
[x]);break;
default:
k.push(ins[x]);
}
while(!s.is_empty())cout<<s.pop();
while(!k.is_empty())cout<<k.pop();cout<<"
}                                is      the      prefix
expression\n\n"; void inftopost(char ins[])

stack s;

case
```

```
cout<<"\n\n";for(int i=0;ins[i]!='\0';i++)switch(ins[i])
{
case 'C:
s.push(ins        [i]);
break;
        'y:
while(!s.is_empty() && s.get_top()!='(' )cout<<s.pop();
s.pop();  break;  case  : ease  ' ': case  case  case  while(!s.is_empty()  &&
priority(s.get_top())>=priority(ins[i]) )cout<<s.pop();
s.push(ins
[i]); break;
       '*':
       '/':
       '^':
default:
cout<<ins[i];
}
while(!s.is_empty())cout<<s.pop();cout<<"
}                                                  is the postfix expressi
                                                void posteva(char ins[])

stack    s;
int sl ,s2;
cout<<"\n\n";
for(int i=0;ins[i]!='\0';i++)
switch(ins[i])

case:
case ' *':case :
case 'A ':
if(!s.is_empty()
)
       -':

case
```

```
s2=s.pop()-48;

if(!s.is_empty() )
s1=s.pop()-48;

s.push(result(s1,s2,ins[i]) +

48); break; default:

s.push(ins[i]);
}

if(!s.is_empty())

cout<<"\n\nRESULT="<<s.pop()-48;cout<<"\n\n";

}

void preeva(char ins [l)


stack     s;
int sls2,i;
//to reverse

for(i=();ins[i] ='\0';i++);i--;cout<<"\n\n" ;for(intx=i;x>=0;x--)switch(ins[x])


ease :case

- :case :case

'A ':

if(!s.is_em

pty() )





            :
```

```
s1=s.pop()-48;
if(!s.is_empty() )
s2=s.pop()-48;
s.push(result(s1,s2,ins[x])+ 48);
break; default:
s.push(ins [x]);
}
if(!s.is_empty())
cout<<"RESULT="<<s.pop()-48;
cout<<"\n\n";
}
int main()


stack   s;   int
eh,        num;
char          a,
ins[2]; do


        "\nThe following operations are available:\n1-Infix to Prefix\n2.Infix to
Postfik\n3.Postfix evaluation\n4.Prefix evaluation\n5.Exit\n\nEnter your choice: '
, ein»ch;


case
```

```
switch(ch)

case 1:
          "\nEnterthe Infix expression:";


inftopre(ins)
; break; case
2:
          "\nEnterthe Infix expression: • ;
ein»ins; inftopost(ins); break; case 3:
          "\nEnterthe Postfix expression:


posteva(ins)
; break; case                              ";
4:
          "\nEnterthe Prefix expression:";
cin>>ins;
preeva(ins)
; break;
ease 5:
          "\n\nYou   chose   to   exit",
return(());
break;
default:
          "\nInvalidchoice \nTry again!";
```

```
) while(ch!=5);
```

```
***************************OUTPUT***************************
[admin@fedora ~]$ g++ hfb2n.cpp
[admin@fedora ~]$ ./a.out

The following operations are available:
I .1nfix to Prefix
2.1nfix to Postfix 3
.Postfix evaluation
4.Prefix evaluation
5.Exit

Enter your choice: I

Enter the Infix expression: (A-B/C)*(A/K-L)

* -A/BC-/AKL is the prefix expression

The following operations are available:
I .1nfix to Prefix
2.1nfix to Postfix 3
.Postfix evaluation
4.Prefix evaluation
```

```
5.Exit

Enter your choice: 2
Enter the Infix expression: A+B
*C+D

ABC*+D4 is the postfix expression

The following operations are
available:
I .1nfix to Prefix
2.1nfix to Postfix 3
.Postfix evaluation
4.Prefix evaluation
5.Exit

Enter your choice: 3

Enter the Postfix expression: 231*+9-

RESULT—4

The following operations are
available:
I .1nfix to Prefix
```

```
2.1nfix to Postfix 3
.Postfix evaluation
4.Prefix evaluation
5.Exit
Enter your choice: 4

Enter the Prefix expression: -+8/632

RESULT=8

The following operations are available:
I .1nfix to Prefix
2.1nfix to Postfix 3
.Postfix evaluation
4.Prefix evaluation
5.Exit

Enter your choice: 5

You chose to exit
```