

## Assignment 10

Class-SETT Roll NO-21430

Batch-F4

Dos-4/12/2020

Problem statement.

Implement the program for expression conversion as infix to post fix and its evaluation using stack based on givencondition 1. operands and operator both must be single that acter.

2. Input postfix expression must be in desired format.

3. only +, -, \* and '1' operands are expected

Objective :

1. Understand the concept of how to conxent infix to postfix. 2 Understand now to evaluate the expression

using stack

Outcomes.

1. Will be able to understand concept of now to convert infix to postfix expression

2. Will be able to understand how to evaluate the expression using stack

Theory -

stack is abstract datatype and I mear data
structure. A. stack, is Ds. in which add not new element

of exiting element always takes place at a same end. This end is known as the the top of the stack. That means that it is possible to remove elements from a stack in a reverse order from the inseration of elements into the stack.

One other way to describing the stack is as last in first (IIFO) abstract data type and linear data structure.

Operations in Stack -:

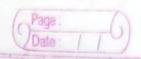
Push operation refers to inserting an element in the Stack. Since there is only one position at which the new element can be inserted.

Top of the Stack. The new element is inserted at top of the Stack.

2 Pop.

Pop operation refers to the removal of an element Again Since we only have acress to the element at the top of the stack, their is only one element that we can remove, we just remove the top of the Stack.

popped element back but its the chaice of programmer.



iii. Empty .:

If returns whether stack is empty or not If top points to - 1 it returns The value otherwise it returns fulse value.

ix Full -:

If returns whether stack is full or not If top points to max-1 it returns true value otherwise it returns false value.

\* Pseudocode

\* ADT for a class Stack

classstack

int top 11 point to top dement.

push() / Push element in Stack

pop() // permove element from Stack

Full () 11 check stack is full or not

Empty() 11 check stack is empty or not

Psuedocode fo initialize

vold initialize()

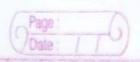
top = -1;

	Page: Date: / /
*	Pseudocode for push
	(h = element to be inserted.
	if (Stack is not full)
	top = top+1;
	data[top]=cb;
	END ic
	dse
	print stack is full "
	END.
	>
	The Party Daniel
*	Pseudocode for pap
	Annual State of the State of th
	if (5tack is not empty)
	top = top-1
	neturn ch
	END IF
	else
	print "stack is empty?
	END SALES OF THE S



*	Pseudocade to convert mfix to postfix()
	mput infix expression
	While (i is not infix length)
	ch = mfix[i]
	if (ch<= ig' and ch>= 'o')
	appond thin postfix.
	else
	if (stact is empty())
	Spush(ch)
	else
	while (stack is not empty)
	Sch & S. pop
	K & priority (sch.ch);
	if(KiSI)
	append sch in postfix.
	dsc
	S.push (sch)
	break.
	END IF
	END While
	sipush (ch)
	iciti
	END While.
	While ( stack is not empty)
	append s.pop in postfix.
	END While
	END.

\* Pseudocode for Evaluate j=0 for i from 1 to postfix length () che postfix[i] if (ch>='0' and ch<='g') s.push (ch-48) else if (ch is 't') nies.pop n2 € 5. pop spush (n2 tri) else if c ch is \*1 mI (S. pop () DIES POPC) Spush (n1 \* n2) clse if (ch is '-') nies.popc) nz (s. popc) Spush (n2-ne) else if (th is "/") ni + si. popi) nz & s.pcp() 5. push ( 12/11 ENDIE END HABITE. return s.pop END.



the same			
Tes-	1	-	00
100	0	5	63
	_	_	_

Description	Input	Experted	Actual	Result
	,	0/P	018.	
anfix	2 * 3-1	2,3,*,1,-	2,3,*1-	
Expression		Ans-5	Ans-5	Pass
Dofix	5+2+6-10	5,2,6.* +10	5,2,6,4,+,	
Expression		Ams-1	10,-	Pass
	Expression	Expression  Defix 5+2+6-10	7nfix 2*3-1 2,3*,1,- Expression Ans-5  Defix 5+2*6-10 5,2,6.*,+,10 Expression -	Theix 2*3-1 2.3*11- 2,3*1-  Expression Ans-5  Theix 5+2*6-10 5.2.6.*+10 5.2.6.*+  Expression - 10, -

## \* Complexity.

function	Time Complexity	Space Complexity
Drittalize	0(1)	0(1)
push()	0(1)	0(1)
pop ()	0(1)	(1)0
empty()	0(1)	0(1)
Ryio	0(1)	(1)0
mainci	0(1)	0(1)
convert()	0(n2)	ocn)
evaluate()	o(n2)	0(h)
priority	0(1)	0(1)
Conclusion.		

the learn to implement Stack Pota structure

```
#include<iostream>
    #include<stdlib.h>
    using namespace std;
    #define MAX 30
                                                                                                 Report ...
    class Stack
                                                                                                    Compiler
 6...
                                                                                                    Resources
 7
        int top:
                                                                                                    ✓ Debug
 8
        int data [MAX];
                                                                                                   Find Results
 9
        public:
                                                                                                     Close
                                                                                                   Compile Log
            Stack()
10
11
12
                 top=-1;
13
14
            void push(int x)
15
                 if(!full())
16
17
18
                     top++;
                     data[top] = x;
19
20
21
                 else
22
                     cout<<"\nStack is Full\n";
23
24
25
26
            int pop()
27
28
                 int x
29
                 if(!empty())
30
31
                     x=data[top];
32
                     top--;
```

```
34
35
                 else
36
37
                     cout<<"\nStack is empty\n";</pre>
                                                                                                 Report ...
38
                     return 0;
                                                                                                    Compiler
39
                                                                                                   Resources
40
                                                                                                    ✓ Debug
             int full()
41
                                                                                                   Find Results
42
                                                                                                     Close
43
                 if(top==MAX-1)
                                                                                                   Compile Log
44
45
                     return 1;
46
47
                 return 0;
48
49
             int empty()
50
51
                 if(top==-1)
52
53
                     return 1;
54
55
                 return 0;
56
57
             int priority(int op)
58
                 if(op=='*'||op=='/')
59
60
61
                     return 2;
62
63
                 else if(op=='+'||op=='-')
64
65
                     return 1;
```

```
66
67
68
69
                                                                                                 Report ...
    void convert(char infix[30],char postfix[30])
                                                                                                    Compiler Compiler
71
                                                                                                   Resources
72
        Stack st;
                                                                                                    Debug
         char expr st op;
                                                                                                   Find Results
                                                                                                     Close
74
         int i,j;
                                                                                                   Compile Log
75
         i=0:
76
        while(infix[i]!='\0')
77
78
79
             expr=infix[i];
             if(expr=='*'||expr=='+'||expr=='-')
80
81
82
                 if(st.empty())
83
84
                     st.push(expr);
85
                 else
86
87
                     while(!st.empty())
88
89
90
                         st_op=st.pop();
                         if(st.priority(st_op)>=st.priority(expr))
91
92
93
                             postfix[j]=st_op;
94
95
                         else
96
97
```

```
break;
99
100
101
102
                      st.push(expr);
                                                                                                    Report ...
103
                                                                                                        Compiler
104
                                                                                                       Resources
              else
105
                                                                                                        Debug
106
                                                                                                       Find Results
107
                  postfix[j]=expr;
                                                                                                        Close
                                                                                                       Compile Log
108
                  j++;
109
110
111
112
         while(!st.empty())
113
114
              postfix[j]=st.pop();
115
              j++;
116
117
          postfix[j]='\0';
118
119
     int evaluate(char postfix[30])
120
         Stack st;
121
122
          char num[2];
123
         num[1]='\0';
          char st op;
124
125
          int value, i;
126
          i=0;
127
         while(postfix[i]!='\0')
128
129
              st op=postfix[i];
              if(st_op=='*'||st_op=='/'||st_op=='+'||st_op=='-')
130
```

```
133
                  int ans;
134
                  x1=st.pop();
135
                  x2=st.pop();
                  if(st_op=='*')
136
                                                                                                    Report ...
137
                                                                                                       Compiler
138
                      ans = x2*x1;
                                                                                                      Resources
139
                      st.push(ans);
                                                                                                       Debug
140
                                                                                                      Find Results
                  else if(st_op='/')
                                                                                                        Close
141
                                                                                                      Compile Log
142
143
                      ans = x2/x1;
144
                      st.push(ans);
145
                  else if(st_op='+')
146
147
148
                      ans = x2+x1;
149
                      st.push(ans);
150
                  else if(st_op=-'-')
151
152
153
                      ans = x2-x1;
154
                      st.push(ans);
155
156
157
              else
158
159
                  value=st op-48;
                  st.push(value);
160
161
162
163
         return st.pop();
164
```

Enter expression : 4\*5+6/3

Postfix = 45\*63/+ Evaluation = 22

Process exited after 9.914 seconds with return value 0
Press any key to continue . . . \_