****

**Green University of Bangladesh**

**Department of Computer Science and Engineering(CSE)**

**Faculty of Sciences and Engineering**

**Semester: (Spring, Year:2021), B.Sc. in CSE (Day)**

**LAB REPORT NO #1**

**Course Title: DATA STRUCTURE**

**Course Code:CSE 106 Section:DA**

**Lab Experiment Name: IMPLEMENT THE STACK FOR CHARACTER ITEM**

**Student Details**

|  |  |  |
| --- | --- | --- |
| **Name** | | **ID** |
| **1.** | Mst.shahnaj Akter Smrity | 212902027 |
| **2.** |  |  |
| **3.** |  |  |

**Lab Date : 22/2/2022\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_**

**Submission Date : 28/2/2022\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_**

**Course Teacher’s Name :Rusmita halim chaity \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_**

**[For Teachers use only: Don’t Write Anything inside this box]**

|  |
| --- |
| **Lab Report Status**  **Marks: ………………………………… Signature:.....................**  **Comments:.............................................. Date:..............................** |

1. **TITLE OF THE EXPERIMENT :**

Conversions of arithmetic infix expression into postfix expression.

1. **OBJECTIVE :**

* To learn the uses of stack for equation conversion.

1. **PROBLEM A NALYSIS :**

A+(B\*C-(D/E^F)\*G)\*H, this is an infix expression .We are going to convert it into postfix expression by using stack data structure. In stack insertion and deletion are performed only from one position which is known as ‘top’. Infix expressions are readable and solvable by humans. We can easily distinguish the order of operators, and also can use the parenthesis to solve that part first during solving mathematical expressions where computer cannot differentiate the operators and parenthesis easily, that's why postfix conversion is needed.

|  |  |  |
| --- | --- | --- |
| Symbol | Stack | Expression |
| A |  | A |
| + | + | A |
| ( | +( | A |
| B | +( | AB |
| \* | +(\* | AB |
| C | +(\* | ABC |
| - | +(- | ABC\* |
| ( | +(-( | ABC\* |
| D | +(-( | ABC\*D |
| / | +(-(/ | ABC\*D |
| E | +(-(/ | ABC\*DE |
| ^ | +(-(/^ | ABC\*DE |
| F | +(-(/^ | ABC\*DEF |
| ) | +(- | ABC\*DEF^/ |
| \* | +(-\* | ABC\*DEF^/ |
| G | +(-\* | ABC\*DEF^/G |
| ) | + | ABC\*DEF^/G\*- |
| \* | +\* | ABC\*DEF^/G\*- |
| H | +\* | ABC\*DEF^/G\*-H |
|  |  | ABC\*DEF^/G\*-\* |
|  |  | ABC\*DEF^/G\*-\*+ |

1. **ALGORITHM:**

* Start
* Declare variable’ S’ for stack and other necessary variable
* Using while and if else loop
* while(more tokens)
* x<=next token
* if(x == operand)
* print x
* else
* while(precedence(x)<=precedence(top(s)))
* print(pop(s))
* push(s,x)
* while(! empty (s))
* print(pop(s))
* Print the output
* End

1. **IMPLEMENTATION:**

/\*212902027\*/

#include <stdio.h>

#include <conio.h>

#include <ctype.h>

#define SIZE 50

char s[SIZE];

int top=-1;

push(char elem)

{

s[++top]=elem;

}

char pop()

{

return(s[top--]);

}

int pr(char elem)

{

switch(elem)

{

case '#': return 0;

case '(': return 1;

case '+':

case '-': return 2;

case '\*':

case '/': return 3;

}

}

int main()

{

char infx[50],pofx[50],ch,elem;

int i=0,k=0;

printf("\n\nRead the Infix Expression ? ");

scanf("%s",infx);

push('#');

while( (ch=infx[i++]) != '\0')

{

if( ch == '(') push(ch);

else

if(isalnum(ch)) pofx[k++]=ch;

else

if( ch == ')')

{

while( s[top] != '(')

pofx[k++]=pop();

elem=pop();

}

else

{

while( pr(s[top]) >= pr(ch) )

pofx[k++]=pop();

push(ch);

}

}

while( s[top] != '#')

pofx[k++]=pop();

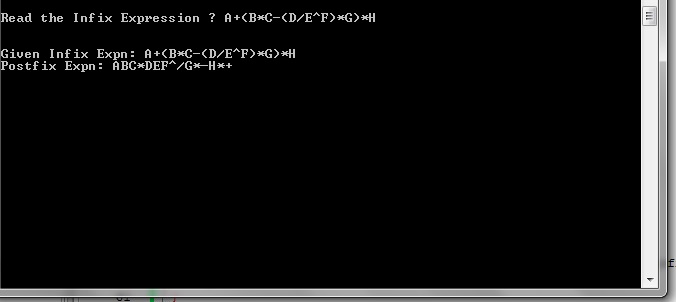
pofx[k]='\0';

printf("\n\nGiven Infix Expn: %s |Postfix Expn: %s\n",infx,pofx);

getch();

}

1. **OUTPUT:**

****

1. **DISCUSSION AND CONCLUSION :**

Initially I found the conversion procedure little bit difficult. But as long as I clear my concept over associativity and precedence I found it very interesting.