BÀI TẬP THỰC HÀNH CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

LAB MANUAL

Academic Year : **2022 - 2023**

Semester : **I**

Prepared by

MSc. KhangVQH

S. No.	Experiment
1	**SEARCHING TECHNIQUES**
2	**SORTING TECHNIQUES**
3	**SORTING TECHNIQUES**
4	**IMPLEMENTATION OF STACK AND QUEUE**
5	**APPLICATIONS OF STACK**
6	**IMPLEMENTATION OF SINGLE LINKED LIST**
7	**IMPLEMENTATION OF DOUBLE LINKED LIST**
8	**IMPLEMENTATION OF STACK USING LINKED LIST**
9	**IMPLEMENTATION OF QUEUE USING LINKED LIST**
10	**IMPLEMENTATION OF BINARY SEARCH TREE**

WEEK-5

APPLICATIONS OF STACK

5.1 OBJECTIVE:

a. Write a C program to convert infix expression into postfix expression using stack.
b. Write a C program to evaluate the postfix expression using stack.

5.2 PROGRAM LOGIC:

Procedure to convert Infix Expression into Postfix Expression

1. Read an infix expression and scan the symbols from left to right.
2. If the symbol is an operand, then write down in the postfix string.
3. If the symbol is a left parenthesis, then push it onto stack.
4. If the symbol is a right parenthesis, then pop the operators from until it find a left parenthesis or the stack is empty.
5. If the symbol is an operator, then check it's priority with the top most operator in the stack.
6. If the incoming operator is having high priority then the top most operator in the stack, then push the new operator onto stack, otherwise pop the existing operator and push the new operator.
7. Display the content of the postfix string.

Example: Convert the following expression A + B * C - D / E * H into its equivalent postfix expression.

Symbol	Postfix String	Stack	Remarks
A	A	+	Place A in the postfix string
+	A	+	Push + onto stack
B	A B	+	Place B in the postfix string
*	A B	+ *	Push * onto stack
C	A B C	+ *	Place C in the postfix string
-	A B C * +	-	Pop * and + from stack and push -.
D	A B C * + D	-	Place D in the postfix string
/	A B C * + D	- /	Push / onto stack
E	A B C * + D E	- /	Place E in the postfix string
*	A B C * + D E /	- *	Push * onto stack
H	A B C * + D E / H	- *	Place H in the postfix string
End of string	A B C * + D E / H * -		The input is now empty, pop the output symbols from the stack until it is empty.

Procedure to evaluate a Postfix Expression

1. Read a postfix expression and scan the symbols from left to right.
2. If the symbol is an operand, then push it onto the stack.
3. If the symbol is an operator, the pop the top most two symbols and apply the operator.
4. Then push the result again in to stack.
5. Display the final result which is in stack.

Evaluate the postfix expression: 6 5 2 3 + 8 * + 3 + *

Symbol	Operand 1	Operand 2	Value	Stack	Remarks
6				6	
5				6, 5	
2				6, 5, 2	
3				6, 5, 2, 3	The first four symbols are placed on the stack.
+	2	3	5	6, 5, 5	Next a '+' is read, so 3 and 2 are popped from the stack and their sum 5, is pushed
8	2	3	5	6, 5, 5, 8	Next 8 is pushed
*	5	8	40	6, 5, 40	Now a '*' is seen, so 8 and 5 are popped as 8 * 5 = 40 is pushed
+	5	40	45	6, 45	Next, a '+' is seen, so 40 and 5 are popped and 40 + 5 = 45 is pushed
3	5	40	45	6, 45, 3	Now, 3 is pushed
+	45	3	48	6, 48	Next, '+' pops 3 and 45 and pushes 45 + 3 = 48 is pushed
*	6	48	288	**288**	Finally, a '*' is seen and 48 and 6 are popped, the result 6 * 48 = 288 is pushed

5.3 IMPLEMENTATION:

Program to convert Infix Expression Into Postfix Expression

OUTPUT:

```
The postfix expression of infix expression A + B × C - D / E × H is
 A B C × + D E / H × -
```

5.4 LAB ASSIGNMENT:

1. Formulate a program to convert infix expression into postfix expression.
2. Write a program to evaluate any postfix expression.
3. Compose a program to convert infix expression into prefix expression.
4. Write a program to convert prefix expression into postfix expression.
5. Write a program to evaluate any prefix expression.

5.5 POST-LAB VIVA QUESTIONS:

1. What is the output of the following expression: 2 3 4 5 + * -
2. What is the advantage of postfix expression?
3. What is the maximum difference between number of operators and operands?
4. Which expression doesn't require parenthesis?
5. What is the output of the following expression: + * - 2 3 4 5