Name :- Anurag Halder Roll no :- 2215055
Subject :- Data Structures ~~and~~ Lab
Enrolment No :- MITU21BTIT0010

Lab - Writ- up (Que - Ans)
Practical No :- 8

Q.1) What is an expression ?
→ An expression is a collection of operators and operands that represents a specific value. In above definition, operator is a symbol which performs a particular task like arithematic operation or logical operation or conditional operation, etc. Operands are the values on which the operators can perform the task.
An expression is a ~~collection of operators~~ number, a variable, or a combi-nation of numbers and vakiables and operation symbols.

Q. 2) Explain different types of expressions?
→ Different types of expressions are :-
i) Infix expression
→ In this operator is used in between operands.
Example → a+b

b) Postfix expression –
In this operator is used after operand.
Example – ab+

c) Prefix expression
In this expression, operator is used before operands :-
example :- +ab

Q 3) Explain the role of stack for Infix to postfix conversion?

→ We know that the stack data structure works on the last in First out (LIFO) principle. Therefore previously encountered operands can easily be accessed using the stack. In Infix to postfix conversion, we use a stack. The stack helps us to store the operands. Whenever an operator is found, we pop two operands from the stack and push a new operand, which is the result of the current operator on the popped operands, into the stack with parenthesis. Add them in the stack maintaining the precedence of them.

4) Explain postfix evaluation.

→ Create a stack to store operands (or values). Scan the given expression from left to right and do the following for every scanned element. If the element is a number, push it into the stack. If the element is an operator, pop operands for the operator from stack.

Evaluate the operator and push the result back to the stack.

When the expression is ended, the number in the stack is the final answer.