### 

# **NATIONAL INSTITUTE OF TECHNOLOGY RAIPUR**

### 

## **B.Tech. (5th Semester)**

## **Assignment No :- 5**

## **Department of Computer Science & Engineering**

## 

## **Subject: Advance data Structure**

## **Lab Code- CS105201CS**

## 

## **Date:- 25/08/2025**

### **Submitted By :- Jayesh Sharma**

### **Roll No :- 23115045**

### **Lab Batch No :- 1**

### 

### **Q.1) Write a code that parses an expression and checks for valid parenthesis:-**

| #include <bits/stdc++.h> using namespace std;  bool isBalanced(string expr) {  stack<char> st;  for (char ch : expr) {  if (ch == '(') st.push(ch);  else if (ch == ')') {  if (st.empty()) return false;  st.pop();  }  }  return st.empty(); }  int main() {  string expression;  cout << "Enter an expression: ";  cin>>expression;  if (isBalanced(expression)) cout << "Balanced." << endl;  else cout << "NOT Balanced." << endl;   return 0; } |
| --- |

### **Q.2)Write an algorithm and program to convert the expression from Infix notation to Postfix notation:-**

| #include <bits/stdc++.h> using namespace std;  int precedence(char op) {  if (op == '^') return 3;  if (op == '\*' || op == '/') return 2;  if (op == '+' || op == '-') return 1;  return 0; }  bool isOperator(char c) {  return (c == '+' || c == '-' || c == '\*' || c == '/' || c == '^'); }  string infixToPostfix(const string& infix) {  stack<char> st;  string postfix = "";   for (char ch : infix) {  if (isalnum(ch)) postfix += ch;  else if (ch == '(') st.push(ch);  else if (ch == ')') {  while (!st.empty() && st.top() != '(') {  postfix += st.top();  st.pop();  }  st.pop();  }  else if (isOperator(ch)) {  while (!st.empty() && precedence(st.top()) >= precedence(ch)) {  postfix += st.top();  st.pop();  }  st.push(ch);  }  }   while (!st.empty()) {  postfix += st.top();  st.pop();  }   return postfix; }  int main() {  string infix;  cout << "Enter an infix expression: ";  cin >> infix;   string postfix = infixToPostfix(infix);  cout << "Postfix expression: " << postfix << endl;   return 0; } |
| --- |

### **Q.3) Write an algorithm and program to evaluate the given Postfix notation:-**

| #include <bits/stdc++.h> using namespace std;  int evaluatePostfix(const string& postfix) {  stack<int> st;   for (char ch : postfix) {  if (isdigit(ch)) {  st.push(ch - '0');  }  else {  int oprn2 = st.top(); st.pop();  int oprn1 = st.top(); st.pop();   switch (ch) {  case '+': st.push(oprn1 + oprn2); break;  case '-': st.push(oprn1 - oprn2); break;  case '\*': st.push(oprn1 \* oprn2); break;  case '/': st.push(oprn1 / oprn2); break;  case '^': st.push(pow(oprn1, oprn2)); break;  default: throw runtime\_error("Invalid");  }  }  }  return st.top(); }  int main() {  string postfix;  cout << "Enter a postfix expression: ";  cin >> postfix;   cout << "Evaluated result: " << evaluatePostfix(postfix) << endl;  return 0; } |
| --- |