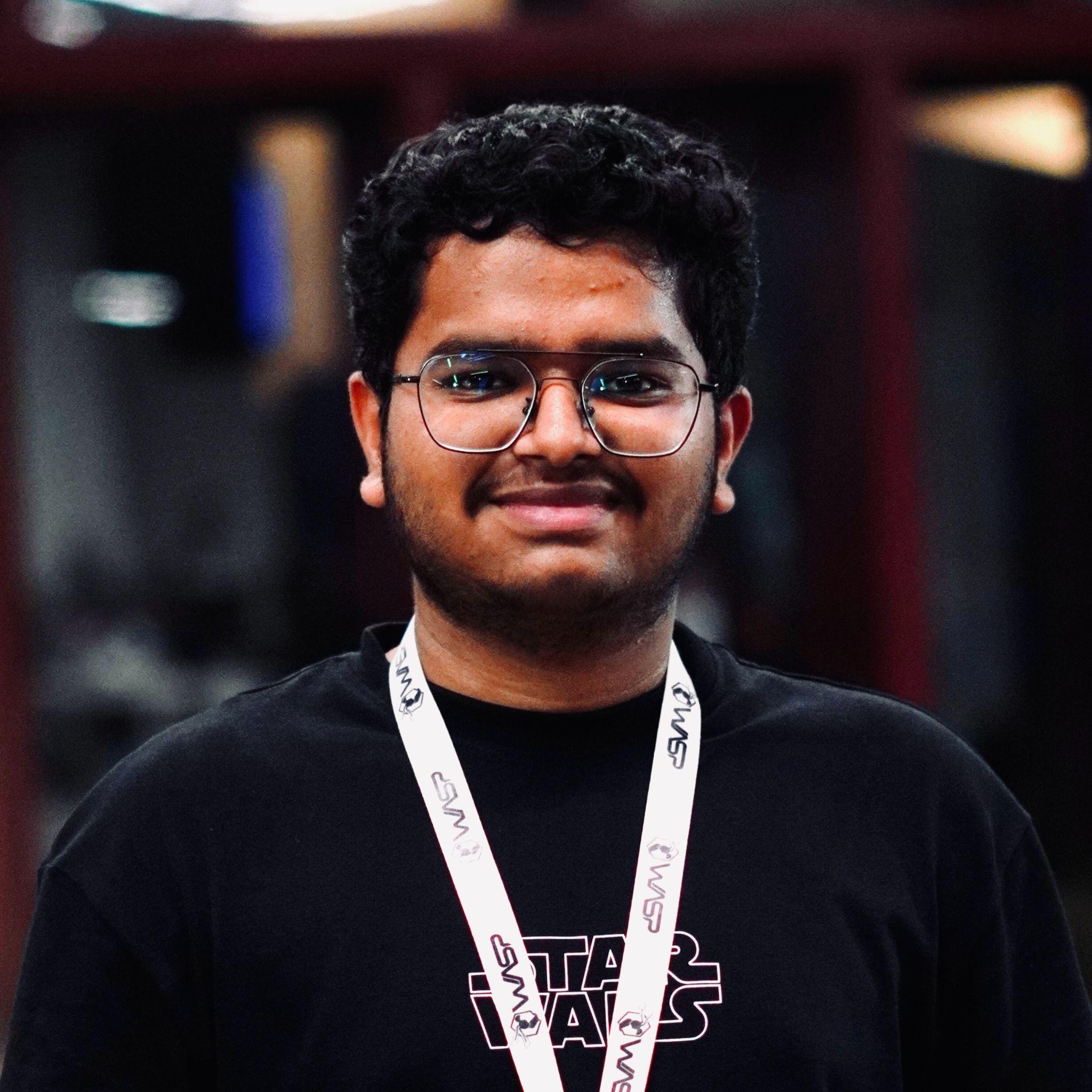
Name: Dhruv Mittal

Roll no: 1024030364

Subgroup: 2C24



**Q-1 :**

/\*

Menu driven program demonstrating Stack operations using array:

(i) push(), (ii) pop(), (iii) isEmpty(), (iv) isFull(), (v) display(), (vi) peek()

\*/

#include <iostream>

using namespace std ;

class Stack {

public:

int size ;

int \*arr ;

int top ;

Stack(int size){

this->size = size ;

arr = new int[size] ;

top = -1 ;

}

void push(int x){

if(top < size-1){

top++ ;

arr[top] = x ;

}

else{

cout << "Stack Overflow" << endl ;

}

}

void pop(){

if(top >= 0){

top -- ;

}

else{

cout << "Stack Underflow" << endl;

}

}

bool isEmpty(){

if(top == -1){

return true ;

}

else{

return false ;

}

}

bool isFull(){

if(top == size-1){

return true ;

}

else{

return false ;

}

}

void display(){

if(top == -1){

cout << "Stack is empty" << endl;

return;

}

else{

cout << "Stack elements : " ;

for(int i=0 ; i<=top ; i++){

cout << arr[i] << " " ;

}

cout << endl ;

}

}

int Top(){ // top = peek

if(top == -1){

return -1 ;

}

else{

return arr[top] ;

}

}

};

int main(){

Stack S1(5);

S1.push(33) ;

S1.push(42) ;

S1.push(21) ;

cout << S1.Top() << endl ; // 21

S1.pop() ;

cout << S1.Top() << endl ; // 42

cout << S1.isFull() << endl ; // 0 -> false

cout << S1.isEmpty() << endl ; // 0 -> false

S1.display() ;

return 0 ;

}

**Q-2 :**

/\*

2. Given a string, reverse it using STACK. For example “DataStructure” should be output as

“erutcurtSataD.”

\*/

#include <iostream>

#include <stack>

using namespace std;

int main() {

string str ="DataStructure";

stack<char> s;

for(int i=0 ;i< str.length(); i++){

char ch = str[i];

s.push(ch);

}

cout << "Reversed: ";

while (!s.empty()) {

cout << s.top();

s.pop();

}

cout << endl;

return 0;

}

**Q-3 :**

#include <iostream>

#include <stack>

using namespace std ;

bool isbalanced(string str){

stack<int> s ;

for(int i=0 ; i<str.length() ; i++){

if(str[i] == '(' || str[i] == '[' || str[i] == '{'){

s.push(str[i]) ;

}

else if(str[i] == ')' || str[i] == ']' || str[i] == '}'){

if(s.empty()){

return false ;

}

char Top = s.top() ;

s.pop() ;

if(str[i]==')' && Top != '(' || str[i]==']' && Top != '[' || str[i]=='}' && Top != '{'){

return false ;

}

}

}

return s.empty();

}

int main(){

string test1 = "{[()]}";

string test2 = "{[(])}";

cout << isbalanced(test1) << endl; // 1 (true)

cout << isbalanced(test2) << endl; // 0 (false)

return 0 ;

}

**Q-4 :**

/\*

Write a program to convert an Infix expression into a Postfix expression.

\*/

#include <iostream>

#include <stack>

using namespace std;

int priority(char c)

{

if (c == '^')

return 3;

if (c == '\*' || c == '/')

return 2;

if (c == '+' || c == '-')

return 1;

return -1;

}

string infixtopostfix(string s)

{

int n = s.length();

stack<char> st;

string ans = "";

for (int i = 0; i < n; i++)

{

if ((s[i] >= 'A' && s[i] <= 'Z') || (s[i] >= 'a' && s[i] <= 'z') || (s[i] >= '0' && s[i] <= '9'))

{

ans += s[i];

}

else if (s[i] == '(')

{

st.push(s[i]);

}

else if (s[i] == ')')

{

while (!st.empty() && st.top() != '(')

{

ans += st.top();

st.pop();

}

if (!st.empty())

st.pop();

}

else

{

while (!st.empty() && priority(s[i]) <= priority(st.top()))

{

ans += st.top();

st.pop();

}

st.push(s[i]);

}

}

while (!st.empty())

{

ans += st.top();

st.pop();

}

return ans;

}

int main()

{

string str;

cout << "Enter Infix: ";

cin >> str;

cout << "Postfix: " << infixtopostfix(str) << endl;

return 0;

}

**Q-5 :**

/\*

Write a program for the evaluation of a Postfix expression.

\*/

#include <iostream>

#include <stack>

#include <math.h>

using namespace std;

int evaluatePostfix(string str) {

stack<int> s;

for (int i=0; i< str.size();i++) {

char ch = str[i];

if (isdigit(ch)) s.push(ch - '0');

else {

int val2 = s.top(); s.pop();

int val1 = s.top(); s.pop();

switch (ch) {

case '+': s.push(val1 + val2); break;

case '-': s.push(val1 - val2); break;

case '\*': s.push(val1 \* val2); break;

case '/': s.push(val1 / val2); break;

}

}

}

return s.top();

}

int main() {

string str;

cout << "Enter Postfix: ";

cin >> str;

cout << "Evaluation: " << evaluatePostfix(str) << endl;

return 0;

}