3. Write a program that checks if an expression has balanced parentheses.

#include<iostream>

#include<cstring>

using namespace std;

#define MAX 100

char stackArr[MAX];

int top = -1;

bool isEmpty() {

return (top == -1);

}

bool isFull() {

return (top == MAX-1);

}

void push(char x) {

if(isFull()) {

cout<<"Stack Overflow! Cannot push"<<endl;

}

else {

stackArr[++top] = x;

}

}

char pop() {

if(isEmpty()) {

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

return '\0';

}

else {

return (stackArr[top--]);

}

}

bool isbalanced(char str[],int n) {

top = -1;

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

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

push(str[i]);

}

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

if(isEmpty()) {

return false;

}

char topChar = pop();

if((str[i] == ')' && topChar != '(') ||

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

(str[i] == ']' && topChar != '[')) {

return false;

}

}

}

return isEmpty();

}

int main() {

char str[MAX];

cout << "Enter string : ";

cin >> str;

int n = strlen(str);

if(isbalanced(str, n)) {

cout << "Balanced" << endl;

} else {

cout << "Not Balanced" << endl;

}

return 0;

}

OUTPUT :

