PROBLEM STATEMENT 1

/* Write a program using a stack for push, pop, peek, and isEmpty operations. Write isBalanced() Function that Iterates through the input expression, Pushes opening brackets onto the stack. For closing brackets, it checks the top of the stack for a matching opening bracket. Ensures that all opening brackets are matched by the end of the traversal. Main Function: Accepts a string expression from the user. Uses isBalanced() to determine if the parentheses in the expression are balanced.
*/

```
#include <iostream>
#include <string>
using namespace std;
#define MAX 50

class Stack {
    char arr[MAX];
    int top;

public:
    Stack() {
        top = -1;
    }

    // Push function
    void push(char ch) {
        if(top>=MAX - 1)
            cout<<"Stack is full!\n";
        else {
            top = top + 1;
            arr[top] = ch;
        }
    }

    // Pop function
    char pop() {
        if(top == -1) {
            cout<<"Stack is empty!\n";
            return '\0';
        } else {
            char ch = arr[top];
            top = top - 1;
```

```
            return ch;
        }
    }

    // Peek function
    char peek() {
        if(top == -1)
            return '\0';
        return arr[top];
    }

    bool isEmpty() {
        return top == -1;
    }
};

// Function to check if expression is balanced
bool isBalanced(string exp) {
    Stack s;
    char x;

    for(int i=0;i<exp.length();i++) {
        char ch = exp[i];

        if(ch == '{' || ch == '[' || ch == '(') {
            s.push(ch);
        }else if(ch == '}' || ch == ']' || ch == ')') {
            if(s.isEmpty())
                return false;

            x=s.peek();

            if((ch == '}' && x == '{') ||
                (ch == ')' && x == '(') ||
                (ch == ']' && x == '[')) {
                s.pop();
            } else {
                return false;
            }
        }
```

```
    }

    return s.isEmpty();
}


int main() {
    string expr;
    cout<<”Name: Manasvi Lunawat   PRN:B24CE1136”<<endl;
    cout<<"\nEnter an expression: ";
    cin>>expr;

    if(isBalanced(expr))
        cout<<"The expression is Balanced.\n";
    else
        cout<<"The expression is NOT Balanced.\n";

    return 0;
}
```

OUTPUT: