In any language program mostly syntax error occurs due to unbalancing delimiter such as

(),{},[]. Write C++ program using stack to check whether a given expression is well parenthesized or not.

\*/

#include<iostream>

#include<string.h>

#include<ctype.h>

using namespace std;

#define MAX 100

class Stack {

private:

char data[MAX],str[MAX];

int top,length,count;

void pushData(char);

char popData();

public:

Stack() {

top=-1;

length=0;

count=0;

}

void getString();

void checkExpression();

void extractString();

};

int main() {

Stack obj;

obj.getString();

obj.extractString();

obj.checkExpression();

return 0;

}

void Stack::getString() {

cout<<"\n Enter a code snippet: ";

cin.getline(str,MAX);

length=strlen(str);

}

void Stack::extractString() {

char temp[MAX];

int i,j;

for(i=0; i<length; i++) {

temp[i]=str[i];

}

j=0;

for(i=0; i<length; i++ ) {

if(temp[i]=='('||temp[i]==')'||temp[i]=='{'||temp[i]=='}'||temp[i]=='['||temp[i]==']') {

str[j]=temp[i];

j++;

}

}

length=j; //update length with new str length

}

void Stack::checkExpression() {

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

pushData(str[i]);

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

if(str[i]==popData())

count++;

}

if(count==length) {

cout<<"\n Entered code is correctly parenthesized. \n";

}

else cout<<"\n Entered code is not correctly parenthesized. \n";

}

void Stack::pushData(char temp) {

if(top==MAX-1) {

cout<<"\n Stack Overflow!!!";

return;

}

top++;

data[top]=temp;

}

char Stack::popData() {

if(top==-1) {

cout<<"\n Stack Underflow!!!";

char ch='\n';

return ch;

}

char temp=data[top];

top--;

switch(temp){

case '{': return '}'; break;

case '}': return '{'; break;

case '(': return ')'; break;

case ')': return '('; break;

case '[': return ']'; break;

case ']': return '['; break;

}

}