#include <iostream>

using namespace std;

class stack

{

char s[25];

int top;

public :

void push(char val);

char pop();

bool isempty();

bool isfull();

void display();

stack()

{

top=-1;

}

};

bool stack::isempty()

{

if(top==-1)

{

return -1;

}

else

return 0;

}

void stack::push(char val)

{

if(top<=24)

{

top++;

s[top]=val;

}

else

cout<<"\n stack is full";

}

char stack::pop()

{

char val;

if(!isempty())

{

val=s[top];

top--;

return val;

}

else

{

cout<<"\n stack is empty";

return '\*';

}

}

class paranthesis

{

char expn[25];

stack obj;

public:

void read();

void checkexpn();

};

void paranthesis::read()

{

cout<<"\n enter the expression";

cin>>expn;

cout<<"\nentered expression is"<<expn;

}

void paranthesis::checkexpn()

{

int i,flag;

char ch;

for(i=0;expn[i]!='\0';i++)

{

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

{

obj.push(expn[i]);

}

}

flag=0;

for(i=0;expn[i]!='\0';i++)

{

if(expn[i]=='}'||expn[i]==']'||expn[i]==')')

{

if(!obj.isempty())

{

ch=obj.pop();

if(expn[i]=='}'&&ch!='{')

{

cout<<"\n expression is not well paranthesis";

flag=1;

break;

}

if(expn[i]==']'&&ch!='[')

{

cout<<"\n expression is not well paranthesis";

flag=1;

break;

}

if(expn[i]==')'&&ch!='(')

{

cout<<"\n expression is not well paranthesis";

flag=1;

break;

}

}

else

{

cout<<"\n expression is not well paranthesis";

flag=1;

break;

}

}

}

if(flag==0&&obj.isempty())

cout<<"\n expression is in well paranthesis";

else

cout<<"\n expression is not well paranthesis";

}

int main()

{

paranthesis obj1;

obj1.read();

obj1.checkexpn();

return 0;

}