**Assignment Number: 9**

**Subject: Data Structure and Algorithms**

**Name: Shrirang Mhalgi**

**Roll No.:222006**

**Class: S.E**

**Division: B**

**Batch: B1**

**Title/Problem Statement:**

In any language program mostly syntax error occurs due to unbalancing delimiter such as (),{},[].

Write C++ program using stack to check whether given expression is well parenthesized or not.

**CODE**

#include<iostream>

#include<string>

using namespace std;

#define max 30

class exp

{

char stack[max];

int top;

public:

void push(char);

int pop(char ,char);

char preference();

void check(char[],char[]);

}ob;

void exp::push(char token)

{

if(top==max)

{

cout<<"\nSTACK OVERFLOW !!!";

}

else

{

stack[++top]=token;

}

}

int exp::pop(char token,char ans)

{

if(top==-1 || token!=ans)

{

// cout<<"\nINVALID USE OF PARENTHESIS!!!";

return -1;

}

else

{

top--;

return 0;

}

}

char exp::preference()

{

if(top!=-1)

{

switch(stack[top])

{

case '(' : return ')';

case '{' : return '}';

case '[' : return ']';

}

}

}

void exp::check(char infix[],char op[])

{

int i,j=0,k=0,res;

char label,ans;

top=-1;

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

{

label=infix[i];

if(isalnum(label))

op[j++]=label;

else if(label=='(' || label =='{' || label =='[')

push(label);

else if(label==')' || label =='}' || label ==']')

{

ans=preference();

res=pop(label,ans);

if(res==-1)

break;

}

}

if(top!=-1 || res==-1)

{

cout<<"\nINVALID USE OF PARENTHESIS!!!";

}

else

{

cout<<"\nEXPRESSION HAS BEEN FULLY PARENTHESISED!!!";

}

}

int main()

{

char infix[max],op[max];

cout<<"\nENTER AN EXPRESSION IN INFIX : ";

ws(cin);

cin.getline(infix,30);

ob.check(infix,op);

return 0;

}

/\*OUTPUT

ENTER AN EXPRESSION IN INFIX : (a\*b)+c

EXPRESSION HAS BEEN FULLY PARENTHESISED!!!\*/