AIM: Write a program to check imbalance of parenthesis using stack.

To execute the program as required, we need to consider the following main points:

- Number of opening parenthesis('{') must be same as number of closing parenthesis('}').
- For every '}', there must be a corresponding '{' **before**.
- At any moment of time number of '{' must be >= number of '}'.

PROGRAM:

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#define MAX 50
/* Declare few global variables which can be used by more than 1 function */
char stack[MAX];
int top=-1;
/*-------PUSH Function------*/
void push(char sym)
{
    ++top;
    stack[top]=sym;
}
/*-------POP Function------*/
int pop()
{
  if(top==-1)
  {
   return 0;
  }
  else
  {
    stack[top];
    --top;
    return 1;
  }
}
/*-------Check Function------*/
void check(char exp[])
{
  int length;
  length = strlen(exp);
  for(int i=0; i<length; i++)
  {
```

```
   if(exp[i]=='{')
   {
     push(exp[i]);
   }
   if(exp[i]=='}')
   {
     int x;
     x=pop();
     if(x==1)
     {
       printf("A closing bracket(at
position %d) has a balancing
opening bracket.\n", i+1);
     }
     if(x==0)
     {
       printf("There is no opening
bracket before a closing bracket
(of Postion-%d).\n",i+1);
     }
   }
  }/*For loop Termination*/
  if (pop()==1)
  {
    printf("There is/are extra
opening bracket(s).\n")   ;
  }
}
/*-------MAIN Function------*/
void main()
{
  char exp[MAX];
  clrscr();
  printf("Enter the expression: ");
  gets(exp);
  check(exp);
  getch(); }
```

OUTPUT

Enter the expression:
{a+b{c/d}}
A closing bracket(at position 9) has a balancing opening bracket.
A closing bracket(at position 10) has a balancing opening bracket.

Enter the expression:
{a+b{c/d}}*d}
A closing bracket(at position 9) has a balancing opening bracket.
A closing bracket(at position 10) has a balancing opening bracket.
There is no opening bracket for a closing bracket(of Postion-13).

Enter the expression:
{d*{a+b{c/d}}
A closing bracket(at position 12) has a balancing opening bracket.
A closing bracket(at position 13) has a balancing opening bracket.
There is/are extra opening bracket(s).

Enter the expression: **}}a+b{{**
There is no opening bracket before a closing bracket(of Postion-1).
There is no opening bracket before a closing bracket(of Postion-2).
There is/are extra opening bracket(s).