Brackets are tall punctuation marks used in matched pairs within text, to set apart or interject other text. Brackets refer to different types of brackets in different parts of the world and in different contexts.

Write a program which reads a String, which consists of alphabets [a-z, A-Z] and 3 types of brackets listed below:

1. Parentheses - ()
2. Square brackets - []
3. Braces or Curly brackets - {}

And determine whether every open bracket has a matching close bracket. If any open/close bracket doesn’t have a matching close/open bracket or any extra open/close bracket then it is to be treated as invalid string.

Following are 3 examples of valid string:

* (the[is]{valid})
* {the(is[valid])}
* (this)(is)(valid)

Following are 4 examples of invalid string:

* (the[is]{invalid))
* (the[is]{invalid}}
* (this](is}{invalid)
* [this]{is}(invalid))

input

First line of the input consists of an integer N, followed by N number of strings with each string is on a separate line.

Output

For each input string, print “TRUE” if it is a valid string else print “FALSE”, terminated by newline charater.

**Sample Input:**

4

(the[is]{valid})

(the[is]{valid))

{the(is[valid])}

(this](is}{valid)

**Sample Output:**

TRUE

FALSE

TRUE

FALSE

Test case

9

(test)

[test]

{test}

(test}

{test]

[test)

{test)

(test]

[test}

Testcase 2

6

([one])

{(one)}

[{one}]

([one]}

{(one)]

[{one})

#include<stdio.h>

#include<string.h>

char stack[100];

int ptr;

void push(char v)

{

stack[ptr]=v;

ptr++;

}

char pop()

{

if(ptr==0)

return '0';

ptr--;

return stack[ptr];

}

void fun(char tem[])

{

int len=strlen(tem),i=0,flag=0;

char te;

ptr=0;

//printf("exp = %s %d\n",tem,strlen(tem));

for(i=0;i<len;i++)

{

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

{

push(tem[i]);

}

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

{

te=pop();

if( (te=='{' && tem[i]=='}') || (te=='[' && tem[i]==']') || (te=='(' && tem[i]==')') )

{

}

else

{

flag=1;

}

if(te=='0')

{

flag=1;

}

}

}

if(ptr>0)

{

flag=1;

}

if(flag==1)

{

printf("FALSE\n");

}

else

{

printf("TRUE\n");

}

}

int main()

{

int count=0,i=0;

char exp[100];

scanf("%d",&count);

for(i=0;i<count;i++)

{

scanf("%s",exp);

fun(exp);

}

return 0;

}