#include <assert.h>

#include <ctype.h>

#include <limits.h>

#include <math.h>

#include <stdbool.h>

#include <stddef.h>

#include <stdint.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

char\* readline();

char\* ltrim(char\*);

char\* rtrim(char\*);

int parse\_int(char\*);

char\* isBalanced(char\* s) {

int len = strlen(s);

char stack[len];

int top = -1;

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

char ch = s[i];

if (ch == '(' || ch == '{' || ch == '[') {

stack[++top] = ch;

} else {

if (top == -1) {

return "NO";

}

char topChar = stack[top--];

if ((ch == ')' && topChar != '(') ||

(ch == '}' && topChar != '{') ||

(ch == ']' && topChar != '[')) {

return "NO";

}

}

}

return (top == -1) ? "YES" : "NO";

}

int main() {

FILE\* fptr = fopen(getenv("OUTPUT\_PATH"), "w");

int t = parse\_int(ltrim(rtrim(readline())));

for (int t\_itr = 0; t\_itr < t; t\_itr++) {

char\* s = readline();

char\* result = isBalanced(s);

fprintf(fptr, "%s\n", result);

}

fclose(fptr);

return 0;

}

char\* readline() {

size\_t alloc\_length = 1024;

size\_t data\_length = 0;

char\* data = malloc(alloc\_length);

while (true) {

char\* cursor = data + data\_length;

char\* line = fgets(cursor, alloc\_length - data\_length, stdin);

if (!line) {

break;

}

data\_length += strlen(cursor);

if (data\_length < alloc\_length - 1 || data[data\_length - 1] == '\n') {

break;

}

alloc\_length <<= 1;

data = realloc(data, alloc\_length);

}

if (data[data\_length - 1] == '\n') {

data[data\_length - 1] = '\0';

}

return data;

}

char\* ltrim(char\* str) {

while (\*str != '\0' && isspace(\*str)) {

str++;

}

return str;

}

char\* rtrim(char\* str) {

char\* end = str + strlen(str) - 1;

while (end >= str && isspace(\*end)) {

end--;

}

\*(end + 1) = '\0';

return str;

}

int parse\_int(char\* str) {

char\* endptr;

int value = strtol(str, &endptr, 10);

if (endptr == str || \*endptr != '\0') {

exit(EXIT\_FAILURE);

}

return value;

}