include<bits/stdc++.h>

using namespace std;

float scanNum(char ch){

int value;

value = ch;

return float(value-'0');//return float from character

}

int isOperator(char ch){

if(ch == '+'|| ch == '-'|| ch == '\*'|| ch == '/' || ch == '^')

return 1;//character is an operator

return -1;//not an operator

}

int isOperand(char ch){

if(ch >= '0' && ch <= '9')

return 1;//character is an operand

return -1;//not an operand

}

float operation(int a, int b, char op){

//Perform operation

if(op == '+')

return b+a;

else if(op == '-')

return b-a;

else if(op == '\*')

return b\*a;

else if(op == '/')

return b/a;

else if(op == '^')

return pow(b,a); //find b^a

else

return INT\_MIN; //return negative infinity

}

float postfixEval(string postfix){

int a, b;

stack<float> stk;

string::iterator it;

for(it=postfix.begin(); it!=postfix.end(); it++){

//read elements and perform postfix evaluation

if(isOperator(\*it) != -1){

a = stk.top();

stk.pop();

b = stk.top();

stk.pop();

stk.push(operation(a, b, \*it));

}else if(isOperand(\*it) > 0){

stk.push(scanNum(\*it));

}

}

return stk.top();

}

main(){

string post = "21+3\*";

cout <<postfixEval(post);

}