#include <iostream>

#include <string>

#include <vector>

#include <algorithm>

#include <exception>

#include<cmath>

#define MAX\_STACK 16

using namespace std;

class MyStack

{

private:

int arr[MAX\_STACK];

int sz;

public:

MyStack() // default constructor

{

sz = 0;

}

~MyStack()

{

;

}

void clear()

{

sz = 0;

}

int pop()

{

int result;

if (sz != 0)

{

result = arr[sz - 1];

sz--;

}

else

{

exception e;

throw e;

}

return result;

}

void push(int v)

{

if (sz == MAX\_STACK)

{

return;

}

arr[sz] = v; //??

sz++;

}

int size()

{

return sz;

}

string toString()

{

string result;

result = "[";

for (int i = sz; i > 0; i--)

{

result = result + to\_string(arr[i - 1]);

if (i != 1)

{

result = result + ", ";

}

}

result = result + "]";

return result;

}

};

int main()

{

string input;

MyStack calculator;

while (cin >> input)

{

if (input == "?")

{

cout << calculator.toString() << endl;

}

else if (input == "+")

{

int a;

int b;

try

{

b = calculator.pop();

}

catch (exception e)

{

cout << "#Not enough arguments." << endl;

continue;

}

try

{

a = calculator.pop();

}

catch (exception e)

{

cout << "#Not enough arguments." << endl;

calculator.push(b);

continue;

}

calculator.push(a + b);

}

else if (input == "-")

{

int a;

int b;

try

{

b = calculator.pop();

}

catch (exception e)

{

cout << "#Not enough arguments." << endl;

continue;

}

try

{

a = calculator.pop();

}

catch (exception e)

{

cout << "#Not enough arguments." << endl;

calculator.push(b);

continue;

}

calculator.push(a - b);

}

else if (input == "\*")

{

int a;

int b;

try

{

b = calculator.pop();

}

catch (exception e)

{

cout << "#Not enough arguments." << endl;

continue;

}

try

{

a = calculator.pop();

}

catch (exception e)

{

cout << "#Not enough arguments." << endl;

calculator.push(b);

continue;

}

calculator.push(a \* b);

}

else if (input == "/")

{

int a;

int b;

try

{

b = calculator.pop();

}

catch (exception e)

{

cout << "#Not enough arguments." << endl;

continue;

}

try

{

a = calculator.pop();

}

catch (exception e)

{

cout << "#Not enough arguments." << endl;

calculator.push(b);

continue;

}

calculator.push(a / b);

}

else if (input == "%")

{

int a;

int b;

try

{

b = calculator.pop();

}

catch (exception e)

{

cout << "#Not enough arguments." << endl;

continue;

}

try

{

a = calculator.pop();

}

catch (exception e)

{

cout << "#Not enough arguments." << endl;

calculator.push(b);

continue;

}

calculator.push(a % b);

}

else if (input == "^")

{

int a;

try

{

a = calculator.pop();

}

catch (exception e)

{

cout << "#Not enough arguments." << endl;

continue;

}

cout << a << endl;

}

else if (input == "!")

{

break;

}

else // either a number or invalide.

{

try

{

int result = stoi(input);

calculator.push(result);

}

catch (exception e)

{

cout << "#Invalid input." << endl;

}

}

}

return 0;

}