#include <iostream>

#include <stack>

const unsigned long int XX\_BIT\_USIGNED\_INT = 1048575;

using namespace std;

int postfixAddSubtr(string &S) {

stack<string> stk;

const string delim = " ";

string single\_operation;

for(size\_t pos = S.find(delim); (pos != string::npos) || (!S.empty()); pos = S.find(delim)) {

if((pos != string::npos)) {

single\_operation = S.substr(0, pos);

S.erase(0, pos + delim.length());

}

else {

single\_operation = S;

S.clear();

}

if(!isdigit(single\_operation[0])) {

if(single\_operation == "DUP") {

stk.push(stk.top());

}

else if(single\_operation == "POP") {

stk.pop();

}

else if(stk.size() > 1) {

string operand\_b = stk.top();

stk.pop();

string operand\_a = stk.top();

stk.pop();

if(!(isdigit(operand\_a[0])) || !(isdigit(operand\_b[0]))) {

//cerr << "This program expects more numbers on the stack "

//"than the stack actually contains." << endl;

return -1;

}

else if (single\_operation == "-") {

if(stoi(operand\_b) < stoi(operand\_a)) {

//cerr << operand\_b + " minus " + operand\_a +

//" yields a negative result." << endl;

return -1;

}

else {

int result = stoi(operand\_b) - stoi(operand\_a);

stk.push(std::to\_string(result));

}

}

else {

int result = stoi(operand\_b) + stoi(operand\_a);

stk.push(std::to\_string(result));

}

}

else {

//cerr << "DEFAULT. May be a negative: " + single\_operation << endl;

return -1;

}

}

else {

if((stoul(single\_operation)) > XX\_BIT\_USIGNED\_INT) {

//cerr << "This integer is out of bounds, "

//"greater than 2^20 - 1" << endl;

return -1;

}

else {

stk.push(single\_operation);

}

}

}

if(stk.empty()) {

return -1;

}

else {

return stoi(stk.top());

}

}

int main()

{

string tmp = "13 DUP 4 POP 5 DUP + DUP + -";

//string tmp = "5 6 + -";

//string tmp = "3 DUP 5 - -";

//string tmp = "10485750";

std::cout << postfixAddSubtr(tmp) << std::endl;

return 0;

}