#include<bits/stdc++.h>

using namespace std;

int preferences(char c){

if(c=='\*' || c=='/'){

return 2;

}

else if(c=='+' || c=='-'){

return 1;

}

else{

return 2;

}

}

int operation(int v1,char op,int v2){

if(op=='\*'){

return v1\*v2;

}

else if(op=='/'){

return v1/v2;

}

else if(op=='+'){

return v1+v2;

}

else if(op=='-'){

return v1-v2;

}

else{

return -1000;

}

}

int solve(string s){

stack<int> opnd;

stack<char> oprt;

for(int i=0;i<s.size();i++){

char ch = s[i];

if(ch=='('){

oprt.push(ch);

}

else if(isdigit(ch)){

opnd.push(ch-'0');

}

else if(ch==')'){

while(oprt.size()!=0 && oprt.top()!='('){

int v2 = opnd.top();

opnd.pop();

int v1 = opnd.top();

opnd.pop();

char op = oprt.top();

oprt.pop();

int lres = operation(v1,op,v2);

opnd.push(lres);

}

oprt.pop();

}

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

while(oprt.size()!=0 && oprt.top()!='(' && preferences(ch)<=preferences(oprt.top())){

int v2 = opnd.top();

opnd.pop();

int v1 = opnd.top();

opnd.pop();

char op = oprt.top();

oprt.pop();

int lres = operation(v1,op,v2);

opnd.push(lres);

}

oprt.push(ch);

}

}

while(oprt.size()!=0){

int v2 = opnd.top();

opnd.pop();

int v1 = opnd.top();

opnd.pop();

char op = oprt.top();

oprt.pop();

int lres = operation(v1,op,v2);

opnd.push(lres);

}

return opnd.top();

}

int main(){

string s;

getline(cin,s);

cout<<solve(s)<<endl;

}