class Solution {

public:

int calculate(string s) {

stack<pair<int,int>> st; // pair(prev\_calc\_value , sign before next bracket () )

long long int sum = 0;

int sign = +1;

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

{

char ch = s[i];

if(isdigit(ch))

{

long long int num = 0;

while(i < s.size() and isdigit(s[i]))

{

num = (num \* 10) + s[i] - '0';

i++;

}

i--; // as for loop also increase i , so if we don't decrease i here a sign will be skipped

sum += (num \* sign);

sign = +1; // reseting sign

}

else if(ch == '(')

{

// Saving current state of (sum , sign) in stack

st.push(make\_pair(sum , sign));

// Reseting sum and sign for inner bracket calculation

sum = 0;

sign = +1;

}

else if(ch == ')')

{

sum = st.top().first + (st.top().second \* sum);

st.pop();

}

else if(ch == '-')

{

// toggle sign

sign = (-1 \* sign);

}

}

return sum;

}

};