// C++ program to find value of derivative of

// a polynomial.

#include <bits/stdc++.h>

using namespace std;

long long derivativeTerm(string pTerm, long long val)

{

// Get coefficient

string coeffStr = "";

int i;

for (i = 0; pTerm[i] != 'x'; i++)

coeffStr.push\_back(pTerm[i]);

long long coeff = atol(coeffStr.c\_str());

// Get Power (Skip 2 characters for x and ^)

string powStr = "";

for (i = i + 2; i != pTerm.size(); i++)

powStr.push\_back(pTerm[i]);

long long power = atol(powStr.c\_str());

// For ax^n, we return anx^(n-1)

return coeff \* power \* pow(val, power - 1);

}

long long derivativeVal(string& poly, int val)

{

long long ans = 0;

// We use istringstream to get input in tokens

istringstream is(poly);

string pTerm;

while (is >> pTerm) {

// If the token is equal to '+' then

// continue with the string

if (pTerm == "+")

continue;

// Otherwise find the derivative of that

// particular term

else

ans = (ans + derivativeTerm(pTerm, val));

}

return ans;

}

// Driver code

int main()

{

string str = "4x^3 + 3x^1 + 2x^2";

int val = 2;

cout << derivativeVal(str, val);

return 0;

}