#include <iostream>

using namespace std;

string removeComments(string prgm)

{

int n = prgm.length();

string res;

bool s\_cmt = false;

bool m\_cmt = false;

for (int i=0; i<n; i++)

{

if (s\_cmt == true && prgm[i] == '\n')

s\_cmt = false;

else if (m\_cmt == true && prgm[i] == '\*' && prgm[i+1] == '/')

m\_cmt = false, i++;

else if (s\_cmt || m\_cmt)

continue;

else if (prgm[i] == '/' && prgm[i+1] == '/')

s\_cmt = true, i++;

else if (prgm[i] == '/' && prgm[i+1] == '\*')

m\_cmt = true, i++;

else res += prgm[i];

}

return res;

}

void removeSpaces(string &prgm)

{

int n = prgm.length();

int i = 0, j = -1;

bool spaceFound = false;

while (++j < n && prgm[j] == ' ');

while (j < n)

{

if (prgm[j] != ' ')

{

if ((prgm[j] == '.' || prgm[j] == ',' ||

prgm[j] == '?') && i - 1 >= 0 &&

prgm[i - 1] == ' ')

prgm[i - 1] = prgm[j++];

else

prgm[i++] = prgm[j++];

spaceFound = false;

}

else if (prgm[j++] == ' ')

{

if (!spaceFound)

{

prgm[i++] = ' ';

spaceFound = true;

}

}

}

if (i <= 1)

prgm.erase(prgm.begin() + i, prgm.end());

else

prgm.erase(prgm.begin() + i - 1, prgm.end());

}

string removemacros(string prgm)

{

int n = prgm.length();

string res;

bool s\_cmt = false;

bool m\_cmt = false;

for (int i=0; i<n; i++)

{

if (s\_cmt == true && prgm[i] == '\n')

s\_cmt = false;

else if (m\_cmt == true && prgm[i] == '#' && prgm[i+1] == 'd')

m\_cmt = false, i++;

else if (s\_cmt || m\_cmt)

continue;

else if (prgm[i] == '#' && prgm[i+1] == 'd')

s\_cmt = true, i++;

else if (prgm[i] == '#' && prgm[i+1] == 'd')

m\_cmt = true, i++;

else res += prgm[i];

}

return res;

}

int main()

{

string prgm;

string new\_prgm;

string rem\_macros;

getline(std::cin,prgm,'^');

cout << " Modified Program \n";

new\_prgm = removeComments(prgm);;

removeSpaces(new\_prgm);

rem\_macros = removemacros(new\_prgm);

cout << rem\_macros;

return 0;

}