#include <iostream>

#include<unordered\_map>

using namespace std;

class Node{

public:

char data;

unordered\_map<char, Node\*> children;

bool terminal;

Node(char d){

data=d;

terminal=false;

}

};

class Trie{

Node \*root;

int cnt;

public:

Trie(){

root=new Node('\0');

cnt=0;

}

void insert(char \*w){

Node \*temp=root;

for(int i=0;w[i]!='\0';i++){

char ch=w[i];

if(temp->children.count(ch)){

temp=temp->children[ch];

}else{

Node \*n=new Node(ch);

temp->children[ch]=n;

temp=n;

}

}

temp->terminal=true;

}

bool find(char \*w){

Node \*temp=root;

for(int i=0;w[i]!='\0';i++){

char ch=w[i];

if(temp->children.count(ch)==0){

return false;

}

else{

temp=temp->children[ch];

}

}

return temp->terminal;

}

};

int main() {

Trie t;

char words[][10]={"a","not","news","apple","hello"};

char w[10];

cin>>w;

for(int i=0;i<5;i++){

t.insert(words[i]);

}

if(t.find(w)){

cout<<"Present";

}

else{

cout<<"Absent";

}

}

Input-

apple

no

Output-

Present

Absent