Define a *T\_Prime* number as a number that has exactly 3 distinct divisors. For example, 4 is a*T\_Prime* number, since its only divisors are 1,2 and 4.

My friends Amr and Farag left the classes and started to play hide-and-seek in the college. Amr can search in 5 places, identified by some numbers. Since Farag is an intelligent person, he hides only in places identified by the *T\_Prime*numbers. If Amr searches a place identified by the *T\_Prime* number, he finds Farag and wins.

Given the identifier of the places Amr searched in, determine if he found Farag, i.e. if there is a place identified by the *T\_Prime* number he searched in.

**Example:**

* HideGame([1 , 3 , 4 , 5 , 6]) = "Amr", since 4 is a *T\_Prime* number.
* HidGame([15 , 100 , 20 , 3 , 2]) = "Farag", since Amr didn't search in the room marked by a *T\_Prime* number, so he lost.
* **[input] array.integer identifiers**
  + Array of length 5, identifiers of the places to search in.
* **[output] string**
  + "Amr" if Amr finds Farag, "Farag"otherwise.

<https://codefights.com/challenge/JWRGjuhJKx4hCQu4f>

--ACEPTADO--

#include <iostream>

#include <stdio.h>

#include <conio.h>

#include <vector>

std::string HideGame(std::vector<int> identifiers) {

struct Helper {

bool esT\_Prime(int n){

int cont = 2;

for(int i = 2; i <= n/2; i++)

{

if(n % i == 0) cont ++;

}

if(cont == 3) return true;

return false;

}

};

Helper h;

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

{

if(h.esT\_Prime(identifiers[i])){

return "Amr";

}

}

return "Farag";

}

int main(){

std::vector<int> v;

HideGame(v);

getch();

return 0;

}