


[Watch](#)

[Ask a Question](#)
[All Questions](#) [All Unanswered](#) [Mine](#) [FAQ](#)

C++: how to solve this problem? Almost got it, just stuck a bit

See more: [★ C++](#) [★ C++11](#) [★ C++14](#)

★☆☆☆☆ 1.00/5 (1 vote)

I was grinding C++ problems until I met this one, which I just couldn't AC after a long time (couldn't even pass sample). Please help!

Farmer John has an important task - figuring out what type of hay to buy for his cows.

Farmer John's N cows ($2 \leq N \leq 10^5$) are numbered 1 through N and each cow likes exactly one type of hay $h(i)$:

($1 \leq h(i) \leq N$). He wants all his cows to like the same type of hay.

To make this happen, Farmer John can host focus groups. A focus group consists of getting all cows in a contiguous range numbered i to j , inclusive, together for a meeting. If there is a type of hay that more than half the cows in the group like, then after the focus group finishes meeting, all cows end up liking that type of hay. If no such type of hay exists, then no cows change the type of hay they like. For example, in focus group consisting of a range of 16 cows, 9 or more of them would need to have the same hay preference to cause the remaining cows to switch their preference to match.

Farmer John wants to know which types of hay can become liked by all cows simultaneously. He can only host one focus group at a time, but he can run as many focus groups as necessary to get all cows to like the same type of hay.

Top Experts

Last 24hrs

[Member 1619...](#) 116

[OriginalGriff](#) 65

[Richard MacC...](#) 60

[Dave Kreskowi...](#) 60

This month

[OriginalGriff](#) 1,861

[Richard MacC...](#) 1,544

[Graeme_Grant](#) 935

[Dave Kreskowi...](#) 460

INPUT FORMAT:

The first will consist of an integer T , denoting the number of independent test cases ($1 \leq T \leq 10$)

The first line of each test case consists of N .

The second line consists of N integers, the favorite types of hay $h(i)$ for the cows in order.

It is guaranteed that the sum of N over all test cases does not exceed $2 \cdot 10^5$.

OUTPUT FORMAT:

Output T lines, one line per test case.

If it possible to make all cows like the same type of hay simultaneously, output all possible such types of hay in increasing order. Otherwise, output -1 . When printing a list of numbers on the same line, separate adjacent numbers with a space, and be sure the line does not end with any extraneous spaces.

SAMPLE INPUT:

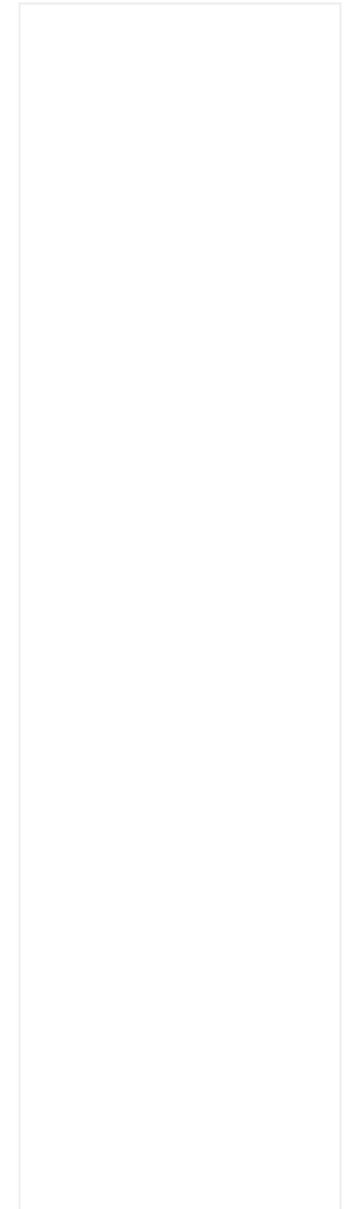
```
5
5
1 2 2 2 3
6
1 2 3 1 2 3
6
1 1 1 2 2 2
3
3 2 3
2
2 1
```

SAMPLE OUTPUT:

```
2
-1
1 2
3
-1
```

In the sample input, there are 5 test cases.

In the first test case, it is only possible to make all cows like type 2. FJ can do this by running a single focus group with all cows.



Related Questions

[How do I solve this problem please](#)

In the second test case, we can show that no cows will change which type of hay they like.

In the third test case, it is possible to make all cows like type 1 by running three focus groups - first by having cows 1 through 4 in a focus group, then by having cows 1 through 5 in a focus group, then by having cows 1 through 6 in a focus group. By similar logic, using cows 3 through 6, cows 2 through 6, then cows 1 through 6, we can make all cows like type 2.

In the fourth test case, it is possible to make all cows like type 3 by running a single focus group with all cows.

In the fifth test case, we can show that no cows will change which type of hay they like.

SCORING:

- * Input 2: $N=2$.
- * Inputs 3-4: $N \leq 50$.
- * Inputs 5-6: $h(i) \leq h(i+1)$ for all $1 \leq i \leq N-1$.
- * Inputs 7-15: No additional constraints.

What I have tried:

I've worked out the following code (which even gets the sample i/o's output last line wrong):

C++

Expand ▼ 

```
#include<bits/stdc++.h>
using namespace std;

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);

    int T;
    cin >> T;
    while(T--) {
        int N;
        cin >> N;
        vector<int> h(N);
        for(int i=0; i<N; i++) {
            cin >> h[i];
        }
    }
}
```

[How do I solve this I have tried little bit but I am stuck in between so help me I am beginner](#)

[How to solve this problem?](#)

[Please! How to solve the MLE in code!](#)

[How do I solve this I have tried little bit but I am stuck in between so help me I am beginner](#)

[How to solve this problem by Python](#)

[How to solve this problem?](#)

[How do I solve this problem](#)

[How to solve this problem?](#)

[Help me to solve this problem](#)

```
map<int, int> count;
for(int i=0; i<N; i++) {
    count[h[i]]++;
}

vector<int> possible;
```

If anyone have any solutions or changes to the code that can pass the question, please tell me! Thank you all so much!

Posted 16hrs ago
Marvin Wang 2023

Add a Solution

Comments

jeron1 16hrs ago [Reply](#)

```
while(T--) {
int N;
cin >> N;
vector<int> h(N);
for(int i=0; i<n; i++)="" {
="" cin="">> h[i];
}
```

I believe N and h will go out of scope once you exit this while block, I am surprised this compiles.

Have a Question or Comment?

2 solutions

Top Rated

Most Recent



Solution 1

Compiling does not mean your code is right! :laugh:

Think of the development process as writing an email: compiling successfully means that you wrote the email in the right language - English, rather than German for example - not that the email contained the message you wanted to send.

So now you enter the second stage of development (in reality it's the fourth or fifth, but you'll come to the earlier stages later): Testing and Debugging.

Start by looking at what it does do, and how that differs from what you wanted. This is important, because it give you information as to why it's doing it. For example, if a program is intended to let the user enter a number and it doubles it and prints the answer, then if the input / output was like this:



Input	Expected output	Actual output
1	2	1
2	4	4
3	6	9
4	8	16

Then it's fairly obvious that the problem is with the bit which doubles it - it's not adding itself to itself, or multiplying it by 2, it's multiplying it by itself and returning the square of the input.

So with that, you can look at the code and it's obvious that it's somewhere here:

C++



```
int Double(int value)
{
    return value * value;
}
```

Once you have an idea what might be going wrong, start using the debugger to find out why. Put a breakpoint on the first line of the method, and run your app. When it reaches the breakpoint, the debugger will stop, and hand control over to you. You can now run your code line-by-line (called "single stepping") and look at (or even change) variable contents as necessary (heck, you can even change the code and try again if you need to).

Think about what each line in the code should do before you execute it, and compare that to what it actually did when you use the "Step over" button to execute each line in turn. Did it do what you expect? If so, move on to the next line.

If not, why not? How does it differ?

Hopefully, that should help you locate which part of that code has a problem, and what the problem is.

This is a skill, and it's one which is well worth developing as it helps you in the real world as well as in development. And like all skills, it only improves by use!

Posted 16hrs ago

[OriginalGriff](#)

Comments

[Marvin Wang 2023](#) 16hrs ago

[Reply](#)



hmm. thank you for your advise, but I couldn't find the bug. It would be more helpful if you could hopefully help me on the code? thanks a lot.

[OriginalGriff](#) 16hrs ago

[Reply](#)



So after I gave you the advice, you spent a maximum of 12 minutes (and probably considerably less than that given that your email system probably only refreshes every 15 minutes) "trying to find the bug" and then gave up?

Debugging is part of the tool set: if you don't learn how to do it with trivial examples of code you are going to be completely stuck when you get up into the "multiple megabytes of source" apps that are the real world!

So tell me what the debugger showed you: where did your expectations differ from reality?

Have a Question or Comment?

Solution 2



This is from the 2024 January USACO Bronze Contest, which runs from January 26 to 29. As stated in the rules for **Contest Conduct and Academic Integrity**,

Quote:

Consultation about the contest problems with people other than the contest director is prohibited.

Quote:

Do not share any technical information or code pertaining to a contest while it is actively running.

Quote:

PARTICIPANTS WHO VIOLATE OF ANY OF THE POLICIES ABOVE WILL BE BANNED FOR LIFE FROM ALL USACO ACTIVITIES. DON'T CHEAT -- THERE ARE NO SECOND CHANCES! (And on a practical note, there really is no benefit to you in cheating on a USACO contest; you can learn much more by making an honest attempt at solving the problems!). We often follow up on cheating offences by contacting a student's teachers or school principal; expulsions have resulted from this in the past, so know that consequences of cheating may extend beyond just participation in USACO.

Posted 9hrs 15mins ago **Updated** 5hrs 15mins ago v2

Member [16190445](#)

Comments

[OriginalGriff](#) 1hr 15mins ago

Good spot! Have my upvote. :thumbsup:



[Nelek](#) 38 mins ago

Now the question is... is the Name of this OP the real name and to be found in the other site?



Have a Question or Comment?

Add your solution here

B / U ~~S~~ small BIG code var < > & link [^] encode untab case indent outdent

Preview

...

Submit your solution!

When answering a question please:

1. Read the question carefully.
2. Understand that English isn't everyone's first language so be lenient of bad spelling and grammar.
3. If a question is poorly phrased then either ask for clarification, ignore it, or **edit the question** and fix the problem. Insults are not welcome.
4. Don't tell someone to read the manual. Chances are they have and don't get it. Provide an answer or move on to the next question.

Let's work to help developers, not make them feel stupid.

This content, along with any associated source code and files, is licensed under [The Code Project Open License \(CPOLE\)](#)

[Advertise](#)

[Privacy](#)

[Cookies](#)

[Terms of Use](#)

Last Updated 26 Jan 2024

Layout: [fixed](#) | [fluid](#)

Copyright © [CodeProject](#), 1999-2024

All Rights Reserved.

Web03 2.8:2024-01-23:1

CodeProject, 20 Bay Street, 11th Floor Toronto, Ontario, Canada M5J 2N8 +1 (416) 849-8900