

ĐỀ THI THỬ

Đề thi gồm 4 bài toán, 10 trang

Bảng C1 (Trung học phổ thông chuyên)

Ngày thi: 20/06/2026

Thời gian làm bài: 150 phút

TỔNG QUAN ĐỀ THI

Bài	Tên bài tập	Thời gian	Bộ nhớ	Điểm	Trang
1	Dãy con tăng dài nhất	2 giây	256 MB	100	2
2	Hàng cây đẹp	1 giây	256 MB	100	3
3	Nhà ảo thuật	2 giây	256 MB	100	5
4	Trại hè Tin học	1 giây	1024 MB	100	8

LƯU Ý DÀNH CHO THÍ SINH

- Tất cả các bài toán đều sử dụng luồng nhập / xuất chuẩn (STDIN / STDOUT).
- Mỗi bài tập gồm nhiều subtask, mỗi subtask gồm nhiều test đơn. Trừ khi có chỉ dẫn khác, điểm của thí sinh được tính theo từng test đơn.
- Thí sinh **không** được phép sử dụng các định hướng biên dịch chương trình có các từ khóa sau: `pragma`, `optimize`, `target`, `O3`, `Ofast`, `unroll-loops`, `avx`, `avx2`, `fma`, `omit-frame-pointer`, ...
- Với các bài toán **tương tác với máy chấm (interactive)**, sau khi bạn in ra đầu ra cho máy chấm, bạn cần in ra kí tự xuống dòng và thực hiện thao tác **flush** luồng ra chuẩn bằng cách gọi các lệnh sau:
 - `fflush(stdout)` hoặc `cout.flush()` trong C++;
 - `stdout.flush()` trong Python;
 - `System.out.flush()` trong Java;
 - `flush(output)` trong Pascal;
- Bảng xếp hạng được ẩn trong suốt kỳ thi. Thí sinh chỉ biết số điểm của mình trong thời gian làm bài.
- Tất cả các bài toán được chấm với **bộ thử chính thức (main testcases)**.

Dãy con tăng dài nhất (100 điểm)

Thời gian: 2.0 giây
Bộ nhớ: 256 megabytes

Huy có một dãy số nguyên dương a_1, a_2, \dots, a_n . Huy muốn bạn tìm một dãy con dài nhất $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ của dãy a sao cho $a_{i_j} < a_{i_{j+1}}$ và $\gcd(a_{i_j}, a_{i_{j+1}}) > d$ với mọi $1 \leq j < k$. Bạn hãy giải quyết bài toán giúp Huy nhé.

Dữ liệu

- Dòng đầu tiên chứa hai số nguyên n và d ($1 \leq n \leq 2 \times 10^5, 0 \leq d \leq 1$).
- Dòng thứ hai chứa n số nguyên dương a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^7$).

Kết quả

- Dòng đầu tiên in ra một số nguyên k là độ dài lớn nhất của dãy con thỏa mãn yêu cầu.
- Dòng thứ hai in ra k chỉ số i_1, i_2, \dots, i_k ($1 \leq i_1 < i_2 < \dots < i_k \leq n$) tương ứng với các phần tử được chọn.
- Nếu có nhiều đáp án thỏa mãn, bạn có thể in ra bất kỳ đáp án nào.

Subtask

- Subtask 1: 15% số điểm có $n \leq 20$.
- Subtask 2: 15% số điểm khác có $n \leq 2000$.
- Subtask 3: 20% số điểm khác có $d = 0$.
- Subtask 4: 50% số điểm còn lại có $d = 1$.

Ví dụ

Dữ liệu	Kết quả
6 1 2 3 4 8 9 12	4 1 3 4 6
5 0 5 1 4 2 3	3 2 4 5

Giải thích

Ở ví dụ thứ nhất, ta có thể chọn các phần tử có chỉ số 1, 3, 4, 6, tương ứng với dãy 2, 4, 8, 12. Dãy này tăng nghiêm ngặt và các giá trị $\gcd(2, 4), \gcd(4, 8), \gcd(8, 12)$ đều lớn hơn 1. Không thể chọn được dãy con hợp lệ nào dài hơn, nên đáp án có độ dài là 4. Ngoài ra, một đáp án hợp lệ khác là 1, 2, 5, 6, tương ứng với dãy 2, 3, 9, 12.

Ở ví dụ thứ hai, một đáp án tối ưu là chọn các chỉ số 2, 4, 5, tương ứng với dãy 1, 2, 3.

Hàng cây đẹp (100 điểm)

Thời gian: 1.0 giây
Bộ nhớ: 256 megabytes

Trong khu vườn ở nhà Hoàng có n cây xếp thành một hàng ngang và được đánh số từ 1 đến n , trong đó cái cây thứ i có chiều cao h_i . Hoàng thấy một dãy các cây là đẹp khi và chỉ khi trong dãy các cây đó không có hai cây nào có chiều cao bằng nhau. Một dãy không có cây nào cũng được tính là đẹp.

Hôm nay, Hoàng quyết định chặt bỏ bớt một số cây để khu vườn trở nên đẹp hơn. Hoàng có q câu hỏi, với mỗi câu hỏi Hoàng muốn biết có bao nhiêu cách chặt bỏ một số cây (có thể không chặt cây nào hoặc chặt tất cả các cây), sao cho những cây còn lại trong các cây được đánh số từ l đến r tạo thành một dãy đẹp.

Các bạn hãy trả lời câu hỏi này giúp Hoàng nhé.

Dữ liệu

- Dòng đầu tiên gồm hai số nguyên dương n, q ($1 \leq n, q \leq 10^5$) lần lượt là số cây trong khu vườn và số câu hỏi của Hoàng.
- Dòng thứ hai gồm n số nguyên dương h_1, h_2, \dots, h_n ($1 \leq h_i \leq 10^9$), trong đó h_i là chiều cao của cây thứ i .
- Trong q dòng tiếp theo, mỗi dòng gồm hai số nguyên dương l, r ($1 \leq l \leq r \leq n$) mô tả một câu hỏi.

Subtask

- Subtask 1: 20% số điểm có $n \leq 20, q = 1$.
- Subtask 2: 20% số điểm khác có $n, q \leq 2000$.
- Subtask 3: 20% số điểm khác có $h_i = i$ với mọi i .
- Subtask 4: 20% số điểm khác có $h_i \leq 100$ với mọi i .
- Subtask 5: 20% số điểm còn lại không có giới hạn gì thêm.

Kết quả

- Với mỗi câu hỏi, in ra trên một dòng số cách chặt bỏ một số cây trong đoạn từ l đến r sao cho các cây còn lại tạo thành một dãy đẹp. Vì kết quả có thể rất lớn nên chỉ cần in phần dư của nó khi chia cho $10^9 + 7$.

Ví dụ

Dữ liệu	Kết quả
5 3	6
1 2 1 3 2	8
1 3	2
2 4	
5 5	

Giải thích

Với câu hỏi đầu tiên, Hoàng có 6 cách chọn các cây để chặt thỏa mãn được liệt kê dưới đây:

- Không chặt cây nào.
- Chặt cây thứ nhất.
- Chặt cây thứ nhất và thứ hai.
- Chặt cây thứ nhất và thứ ba.
- Chặt cây thứ hai và thứ ba.
- Chặt cả ba cây.

Với câu hỏi thứ ba, Hoàng có thể chặt hoặc không chặt cây thứ năm, tổng cộng có 2 cách.



Nhà ảo thuật (100 điểm)

Thời gian: 2.0 giây
Bộ nhớ: 256 megabytes

Đây là bài toán tương tác với máy chấm (interactive).

Hôm nay trường của Đức và Quỳnh mời một nhà ảo thuật đến để biểu diễn cho các bạn học sinh. Trong các tiết mục được biểu diễn, có một tiết mục đoán số rất thú vị.

Để thực hiện tiết mục này, nhà ảo thuật mời Đức và Quỳnh lên sân khấu. Đầu tiên, ông yêu cầu Đức nghĩ về một số tự nhiên từ 0 đến 32, sau đó ông đưa cho Quỳnh một bộ bài với 32 lá được ghi các số nguyên dương phân biệt từ 1 đến 32. Quỳnh được phép bí mật tráo các lá bài, tuy nhiên bộ bài sau khi tráo phải thỏa mãn điều kiện lá bài thứ i từ trên xuống phải ghi một số không nhỏ hơn $i - k$ và không lớn hơn $i + k$, với k là một số tự nhiên không lớn hơn 1, **đồng thời lá bài trên cùng phải ghi số 1**.

Nhà ảo thuật cần đoán ra số mà Đức đang nghĩ trong đầu. Ông có thể thực hiện một số lượt hỏi. Trong mỗi lượt hỏi, nhà ảo thuật nói với Quỳnh một số nguyên dương i , sau đó yêu cầu Quỳnh bí mật cho Đức xem số ghi trên lá bài thứ i từ trên xuống trong bộ bài. Đức so sánh số đó với số mà mình đang nghĩ và cho biết số trên lá bài đó có lớn hơn số mà Đức đang nghĩ hay không.

Nhà ảo thuật tuyên bố rằng ông có thể đoán được số mà Đức nghĩ trong không quá 6 lượt hỏi, bất kể Đức có nghĩ đến số nào và Quỳnh có tráo các lá bài như thế nào. Thật vậy, cho dù các học sinh có yêu cầu ông thực hiện trò biểu diễn đó rất nhiều lần với các số và cách tráo bài khác nhau, ông đều có thể đoán được số mà Đức nghĩ đến.

Đức và Quỳnh rất ấn tượng với tiết mục ảo thuật này và muốn tái hiện nó. Đức nhờ bạn viết một chương trình đóng vai nhà ảo thuật đoán số mà Đức nghĩ đến trong càng ít lượt hỏi càng tốt. Các bạn hãy giải quyết bài toán này giúp Đức nhé.

Tương tác

- Đầu tiên, chương trình của bạn cần nhập vào hai số nguyên T, k , trong đó T là số bộ dữ liệu ($1 \leq T \leq 1000$) và k là tham số giới hạn cách tráo bài ($0 \leq k \leq 1$).
- Với mỗi bộ dữ liệu, máy chấm đã cố định trước một số s mà Đức nghĩ đến và một hoán vị p_1, p_2, \dots, p_{32} của các số từ 1 đến 32 do Quỳnh chọn, thỏa mãn $i - k \leq p_i \leq i + k$ với mọi $1 \leq i \leq 32$. Quy trình tương tác cho bộ dữ liệu đó như sau:

- Để đặt câu hỏi, bạn cần in một dòng theo định dạng:

? i

với $1 \leq i \leq 32$. Khi đó, Quỳnh sẽ bí mật cho Đức xem số p_i ghi trên lá bài thứ i , và máy chấm sẽ trả về:

- * 1 nếu $p_i > s$;
- * 0 nếu $p_i \leq s$.

Nếu câu hỏi sai định dạng, không hợp lệ, hoặc bạn đã vượt quá 32 câu hỏi trong bộ dữ liệu hiện tại, máy chấm sẽ trả về -1 .

- Để đưa ra câu trả lời, bạn cần in một dòng theo định dạng:

! s

với s là số bạn đoán Đức đang nghĩ. Máy chấm sẽ trả về 1 nếu bạn đoán đúng, và trả về -1 nếu bạn đoán sai.

- Nếu đọc được input -1 của máy chấm sau khi đặt câu hỏi hoặc đưa ra câu trả lời, chương trình của bạn cần kết thúc ngay lập tức. Nếu bạn đã đoán đúng trong một bộ dữ liệu, hãy chuyển sang bộ dữ liệu tiếp theo hoặc kết thúc chương trình nếu đã xử lý xong tất cả bộ dữ liệu.

Trong bài này, máy chấm **không thích ứng (non-adaptive)**. Hay nói cách khác, giá trị s và dãy số p ở tất cả các ván chơi đã được cố định trước khi chạy các chương trình.

Lưu ý, sau khi bạn in ra output cho máy chấm, bạn cần in ra ký tự xuống dòng và thực hiện thao tác flush luồng ra chuẩn bằng cách gọi các lệnh sau:

- `fflush(stdout)` hoặc `cout.flush()` trong C++;
- `stdout.flush()` trong Python;
- `System.out.flush()` trong Java;
- `flush(output)` trong Pascal;

Subtask

- Subtask 1: 20% số điểm có $k = 0$.
- Subtask 2: 80% số điểm còn lại có $k = 1$.

Ví dụ

Thí sinh	Máy chấm	Giải thích
	2 1	Máy chấm cho biết có 2 bộ dữ liệu và $k = 1$. Trong bộ dữ liệu đầu tiên, giả sử Đức nghĩ đến số $s = 7$ và Quỳnh chọn một hoán vị p thỏa mãn đề bài.
? 8		Chương trình của bạn hỏi tại vị trí 8. Máy chấm sẽ trả lời xem p_8 có lớn hơn 7 hay không.
	1	Giả sử trong dãy số đang dùng, $p_8 > 7$, nên máy chấm trả về 1.
? 7		Chương trình của bạn hỏi tại vị trí 7.
	0	Giả sử $p_7 \leq 7$, nên máy chấm trả về 0. Từ hai câu trả lời này, có thể suy ra $s = 7$.
! 7		Chương trình của bạn đoán số Đức đã nghĩ là 7.
	1	Bạn đoán đúng nên máy chấm trả về 1 và chuyển sang bộ dữ liệu tiếp theo. Trong bộ dữ liệu này, giả sử Đức nghĩ đến số $s = 13$.
? 14		Chương trình của bạn hỏi tại vị trí 14.
	1	Giả sử $p_{14} > 13$, nên máy chấm trả về 1.
? 13		Chương trình của bạn hỏi tại vị trí 13.
	0	Giả sử $p_{13} \leq 13$, nên máy chấm trả về 0. Từ đó, chương trình có thể kết luận $s = 13$.
! 13		Chương trình của bạn đoán số Đức đã nghĩ là 13.
	1	Bạn đoán đúng nên máy chấm trả về 1. Đây là bộ dữ liệu cuối cùng nên chương trình kết thúc.

Ví dụ trên chỉ nhằm minh họa một quy trình tương tác có thể xảy ra. Lưu ý rằng hoán vị p là bí mật, nên trong thực tế bạn chỉ nhìn thấy các phản hồi 0/1 từ máy chấm.

Trong một lần chạy thực tế, chương trình của bạn cần đọc dữ liệu phản hồi từ máy chấm sau mỗi câu hỏi hoặc câu trả lời được in ra.

Thí sinh có thể tham khảo các lời giải mẫu ở phần **Phụ lục 1** để hiểu quy cách tương tác giữa chương trình và máy chấm.

Chấm điểm

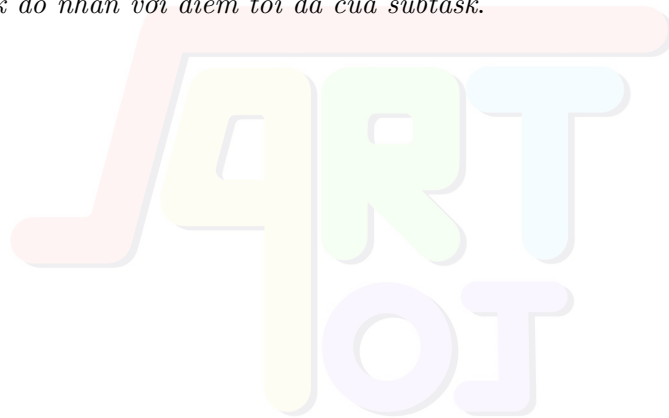
Trong mỗi testcase, coi điểm tối đa là 1, gọi Q là số câu hỏi lớn nhất mà chương trình của bạn sử dụng trong một bộ dữ liệu của testcase đó.

Điểm của bạn cho testcase đó được tính như sau:

- Nếu chương trình của bạn vi phạm quy cách tương tác, đoán sai, hoặc sử dụng quá 32 câu hỏi trong một bộ dữ liệu, bạn được 0 điểm cho testcase đó.
- Nếu $Q \leq 6$, bạn được 1 điểm cho testcase đó.
- Nếu $6 < Q \leq 32$, số điểm của bạn cho testcase đó, ký hiệu là S , được tính theo công thức sau:

$$S = \frac{3.6}{Q - 2} + 0.1$$

Trong bài tập này, điểm của bạn cho mỗi subtask được tính bằng điểm **thấp nhất** mà bạn đạt được cho một testcase trong subtask đó nhân với điểm tối đa của subtask.



Trại hè Tin học (100 điểm)

Thời gian: 1.0 giây

Bộ nhớ: 1024 megabytes

Một trại hè Tin học thu hút rất nhiều bạn học sinh đến từ n trường khác nhau. Các trường được đánh số từ 1 đến n , và trường thứ i có a_i bạn học sinh tham gia trại hè. Trong số n trường, có ít nhất một trường có nhiều hơn một bạn học sinh tham gia.

Ban tổ chức muốn sắp xếp chỗ ngồi cho các trại sinh khi tham gia các hoạt động của trại hè. Các chỗ ngồi được bố trí thành một hàng ngang, với các ghế được đánh số từ 1 đến $a_1 + a_2 + \dots + a_n$. Để khuyến khích các trại sinh từ các trường khác nhau giao lưu và kết bạn với nhau, Ban tổ chức muốn hai trại sinh ngồi liên tiếp nhau bất kỳ đều phải đến từ hai trường khác nhau. Đồng thời, để các trại sinh cùng trường có thể dễ dàng trao đổi thông tin với nhau khi cần thiết, ban tổ chức muốn tìm giá trị d nhỏ nhất sao cho giữa hai trại sinh cùng trường bất kỳ phải có ít nhất một trại sinh khác cùng trường, hoặc không quá d trại sinh khác trường với hai trại sinh đó (hay nói cách khác, chênh lệch số ghế giữa hai trại sinh liên tiếp cùng trường không vượt quá $d + 1$).

Bạn hãy giúp ban tổ chức trại hè tìm một cách sắp xếp thỏa mãn với giá trị d nhỏ nhất, hoặc chỉ ra rằng không tồn tại cách sắp xếp nào như vậy.

Dữ liệu

- Dòng đầu tiên gồm một số nguyên dương n ($1 \leq n \leq 10^5$) là số trường có học sinh tham gia trại hè.
- Dòng thứ hai gồm n số nguyên dương a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^6, n + 1 \leq a_1 + a_2 + \dots + a_n \leq 10^6$), trong đó a_i là số học sinh đến từ trường thứ i .

Kết quả

- Nếu không tồn tại cách sắp xếp thỏa mãn, in ra -1 .
- Ngược lại, in ra một dòng gồm $a_1 + a_2 + \dots + a_n$ số nguyên. Số thứ j là chỉ số trường của trại sinh ngồi ở ghế thứ j .
- Mỗi số nguyên i ($1 \leq i \leq n$) phải xuất hiện đúng a_i lần. Nếu có nhiều cách sắp xếp cùng tối ưu, bạn có thể in ra bất kỳ cách nào. Kết quả của bạn sẽ được chấm đúng nếu có giá trị d nhỏ hơn hoặc bằng kết quả của giám khảo.

Subtask

- Subtask 1: 20% số điểm có $a_1 = a_2 = \dots = a_n$.
- Subtask 2: 20% số điểm khác có $n \leq 3$.
- Subtask 3: 30% số điểm khác có $n \leq 100$ và $a_1 + a_2 + \dots + a_n \leq 1000$.
- Subtask 4: 30% số điểm còn lại không có giới hạn gì thêm.

Ví dụ

Dữ liệu	Kết quả
3 2 2 1	1 2 1 2 3
2 3 1	-1

Giải thích

Ở ví dụ thứ nhất, dãy 1, 2, 1, 2, 3 có đúng 2 học sinh trường 1, 2 học sinh trường 2 và 1 học sinh trường 3. Không có hai học sinh ngồi cạnh nhau đến từ cùng một trường. Hai học sinh trường 1 có một học sinh khác trường ngồi giữa, còn hai học sinh trường 2 có hai học sinh khác trường ngồi giữa, nên các điều kiện đều được thỏa mãn. Cách xếp này có $d = 1$ và là cách xếp tối ưu nhất có thể.

Ở ví dụ thứ hai, có 3 học sinh trường 1 và chỉ 1 học sinh trường 2. Do hai học sinh ngồi cạnh nhau không được đến từ cùng một trường, không thể xếp cả 3 học sinh trường 1 vào hàng ghế mà không có hai học sinh trường 1 nào ngồi liên tiếp nhau. Vì vậy không tồn tại cách sắp xếp thỏa mãn.



———— - Hết! ————

*Thí sinh không được sử dụng tài liệu,
Các bộ kiểm thử đã được kiểm tra chính xác,
Giám thị coi thi không giải thích gì thêm!*

Họ và tên thí sinh: Số báo danh:

Phụ lục 1

Lời giải mẫu cho bài toán Nhà ảo thuật

Dưới đây là lời giải mẫu (chỉ mang tính chất minh họa) bằng các ngôn ngữ lập trình khác nhau để bạn có thể hiểu quy cách tương tác giữa chương trình với máy chấm.

Lời giải mẫu bằng C++:

```
#include <bits/stdc++.h>
using namespace std;
int ask (int i) {
    cout << "? " << i << endl;
    int judge_ans; cin >> judge_ans;
    if (judge_ans == -1) exit (0);
    return judge_ans;
}
void answer (int s) {
    cout << "! " << s << endl;
    int judge_ans; cin >> judge_ans;
    if (judge_ans == -1) exit (0);
    return;
}
int main () {
    int T, k; cin >> T >> k;
    while (T --) {
        int judge_ans = ask (16);
        answer (16);
    }
}
```

Lời giải mẫu bằng Java:

```
import java.util.Scanner;
public class Main {
    static Scanner scanner =
        new Scanner(System.in);
    static int ask(int i) {
        System.out.println("? " + i);
        System.out.flush();
        int judgeAns = scanner.nextInt();
        if (judgeAns == -1) {
            System.exit(0);
        }
        return judgeAns;
    }
    static void answer(int s) {
        System.out.println("! " + s);
        System.out.flush();
        int judgeAns = scanner.nextInt();
        if (judgeAns == -1) {
            System.exit(0);
        }
    }
    public static void main(String[] args) {
        int T = scanner.nextInt();
        int k = scanner.nextInt();
        while (T-- > 0) {
            int judgeAns = ask(16);
            answer(16);
        }
    }
}
```

Lời giải mẫu bằng Python:

```
import sys
def ask(i):
    print(f"? {i}", flush=True)
    judge_ans = int(input())
    if judge_ans == -1:
        sys.exit(0)
    return judge_ans
def answer(s):
    print(f"! {s}", flush=True)
    judge_ans = int(input())
    if judge_ans == -1:
        sys.exit(0)
T, k = map(int, input().split())
for _ in range(T):
    judge_ans = ask(16)
    answer(16)
```

Lời giải mẫu bằng Pascal:

```
program SampleInteraction;
uses sysutils;
function ask(i: longint): longint;
var
    judge_ans: longint;
begin
    writeln('? ', i);
    flush(output);
    readln(judge_ans);
    if judge_ans = -1 then halt(0);
    ask := judge_ans;
end;
procedure answer(s: longint);
var
    judge_ans: longint;
begin
    writeln('! ', s);
    flush(output);
    readln(judge_ans);
    if judge_ans = -1 then halt(0);
end;
var
    T, k: longint;
    judge_ans: longint;
begin
    readln(T, k);
    while T > 0 do
        begin
            judge_ans := ask(16);
            answer(16);
            dec(T);
        end;
end.
```