



# OLYMPIC TIN HỌC SINH VIÊN LẦN THỨ XXVIII, 2019

## Khối thi: SIÊU CÚP

*Thời gian làm bài: 180 phút*

**Ngày thi: 04-12-2019**

**Nơi thi: ĐẠI HỌC BÁCH KHOA ĐÀ NẴNG**

### TỔNG QUAN ĐỀ THI

Bài 1. Thi vấn đáp — ORAL (100 điểm) . . . . .	2
Bài 2. Ghép xâu con — STRLNK (100 điểm) . . . . .	4
Bài 3. Dây con lượn sóng — WIGGLY (100 điểm) . . . . .	5
Bài 4. Cắt bánh — GATEAU (100 điểm) . . . . .	6

# Bài 1. Thi vấn đáp — ORAL

Kì thi kết thúc môn học của một số môn học tại trường Đại học Bách Khoa UTD được tổ chức dưới hình thức kiểm tra vấn đáp. Đến kì thi, một lớp học có  $N$  ( $1 \leq N \leq 300$ ) bạn sinh viên vào phòng thi môn Xác suất. Thầy giáo gọi từng bạn lên một để kiểm tra vấn đáp. Trước khi kết thúc kiểm tra mỗi bạn, thầy giáo hỏi thêm một câu: "Em hãy dự đoán cả lớp có bao nhiêu sinh viên được đánh giá ĐẠT sau buổi thi?". Bạn sinh viên nêu ra một số nguyên trong khoảng từ 0 đến  $N$ . Sau đó thầy sẽ công bố luôn trước cả lớp là bạn này là ĐẠT hay KHÔNG ĐẠT. Câu hỏi thêm có ý nghĩa là nếu như bạn đó không trả lời tốt các câu hỏi trước mà đoán đúng được số người ĐẠT thì theo qui định thầy giáo vẫn phải cho bạn ấy ĐẠT.

Hôm sau Ban thanh tra trường kiểm tra kết quả, nếu thấy có bạn nào trả lời đúng số sinh viên ĐẠT mà thầy lại cho KHÔNG ĐẠT thì thầy sẽ bị kỉ luật và cả lớp sẽ được ĐẠT môn học này. Nếu không có bạn nào như vậy thì kết quả số lượng bạn ĐẠT giữ nguyên.

**Yêu cầu:** Biết rằng thầy giáo nghiêm khắc có xu hướng đánh trượt sinh viên, hãy giúp các bạn sinh viên tìm chiến thuật tốt nhất để cuối cùng càng nhiều bạn ĐẠT càng tốt.

Chương trình phải sử dụng một thư viện riêng. Thư viện bao gồm các file sau: orallib.h (cho C++), orallib.java (cho Java). Trong chương trình của bạn cần khai báo các thư viện này ở đầu chương trình:

- `#include "orallib.h"` đối với C++
- Đối với Java, mọi hàm đều là static được khai báo public thông qua class `orallib`, ví dụ: `n = orallib.get_N()`

Thư viện cung cấp các hàm sau:

- Hàm khởi tạo trò chơi
  - `int get_N();` // đối với C++
  - `public static int get_N();` // đối với JavaChương trình phải gọi hàm này để khởi tạo trò chơi. Hàm này trả về một giá trị  $N$  là số lượng sinh viên. Hàm này chỉ được gọi một lần duy nhất.

Sau hàm khởi tạo cần sử dụng hàm sau để trả lời  $N$  câu hỏi, lần gọi thứ  $i$  là câu trả lời của sinh viên thứ  $i$  cho câu hỏi của thầy giáo.

- Hàm trả lời câu hỏi
  - `int guess(int k);` // đối với C++
  - `public static int guess(int k);` // đối với JavaBạn dùng hàm này để đưa vào  $k$  là câu trả lời cho câu hỏi số người đạt của thầy giáo và hàm sẽ trả về kết quả của thầy giáo cho sinh viên đó, bằng 1 nếu là ĐẠT và bằng 0 nếu là KHÔNG ĐẠT.

Bạn có thể xem các file được cung cấp trên hệ thống để hiểu rõ hơn về cách tương tác với hệ thống.

## Ví dụ

Hàm tương tác	Kết quả
get_N()	3
guess(1)	1
guess(2)	1
guess(3)	0
get_N()	3
guess(1)	1
guess(2)	0
guess(3)	1

## Giải thích

Trong ví dụ thứ nhất, có 3 bạn sinh viên đi thi, bạn thứ nhất đoán có 1 bạn ĐẠT, thầy giáo đánh giá bạn thứ nhất ĐẠT; bạn thứ hai đoán có 2 bạn ĐẠT, thầy giáo đánh giá bạn thứ hai ĐẠT; bạn thứ ba đoán có 3 bạn ĐẠT, thầy giáo đánh giá bạn thứ ba KHÔNG ĐẠT. Thầy giáo không bị kỉ luật, tổng số bạn ĐẠT là 2.

Trong ví dụ thứ hai, có 3 bạn sinh viên đi thi, bạn thứ nhất đoán có 1 bạn ĐẠT, thầy giáo đánh giá bạn thứ nhất ĐẠT; bạn thứ hai đoán có 2 bạn ĐẠT, thầy giáo đánh giá bạn thứ hai KHÔNG ĐẠT; bạn thứ ba đoán có 3 bạn ĐẠT, thầy giáo đánh giá bạn thứ ba ĐẠT. Thầy giáo bị kỉ luật vì bạn thứ hai trả lời đúng số lượng bạn ĐẠT là 2 người nhưng thầy giáo lại đánh giá bạn ấy KHÔNG ĐẠT. Vì vậy cuối cùng cả 3 bạn đều ĐẠT.

### Cách tính điểm

Với mỗi test, gọi  $\delta^*$  là số lượng sinh viên ĐẠT tìm được theo chiến thuật của Ban Giám Khảo. Gọi  $\delta$  là số lượng sinh viên ĐẠT tìm được theo chiến thuật của bạn, điểm của bạn sẽ được tính như sau:

- nếu  $\delta \geq \delta^*$ , đạt 100% số điểm;
- ngược lại, nếu  $\delta \geq \delta^* - 5$ , đạt  $90\% \times \frac{\delta}{\delta^*}$  số điểm;
- ngược lại, nếu  $\delta \geq \sqrt{\delta^*}$ , đạt  $50\% \times \frac{\delta}{\delta^*}$  số điểm;
- ngược lại, không có điểm.



## Bài 2. Ghép xâu con — STRLNK

Bình đam mê lập trình từ nhỏ và được chọn vào lớp huấn luyện tham gia kì thi Olympic tin học cấp trường. Hôm nay cô giáo dạy về cách ghép các chuỗi kí tự cho cả lớp. Đầu tiên cô giáo đưa định nghĩa xâu là một chuỗi các kí tự viết liên tiếp nhau. Tiếp đến là cách ghép xâu. Xâu  $X$  được ghép vào xâu  $Y$  bằng cách viết liên tiếp các kí tự của xâu  $X$  vào sau xâu  $Y$ . Cuối buổi cô giáo đưa ra bài tập cho cả lớp cùng làm để học sinh có thể áp dụng ngay được những kiến thức vừa học. Cô giáo đưa ra một xâu  $S$  và muốn mỗi học sinh phải đưa ra một xâu con  $T$  mà có thể nhân bản một số lần xâu  $T$  lên và ghép vào nhau để tạo ra một xâu là xâu con của  $S$ . Cô giáo giảng thêm xâu con của một xâu được hiểu là một chuỗi các kí tự liên tiếp nằm trong xâu đó.

Để tạo được sự thi đua học tập của các học sinh cô giáo hứa sẽ trao một phần thưởng cho ai đưa ra được xâu mà ghép nhiều lần nhất xâu đó mà vẫn tạo ra xâu con của  $S$ . Bình muốn dành được phần quà này nhưng không dễ dàng như vậy. Xâu  $S$  mà cô giáo đưa ra rất dài, Bình không thể tự làm được nên muốn nhờ các bạn lập trình giải quyết giúp Bình bài toán này.

**Yêu cầu:** Cho xâu  $S$  tìm số lần nhân một xâu  $T$  lên nhiều nhất để ghép thành xâu con của  $S$ .

### Dữ liệu

Vào từ thiết bị vào chuẩn:

- Dòng đầu tiên chứa số nguyên  $q$  là số test ( $1 \leq q \leq 5$ ).
- $q$  dòng tiếp theo mỗi dòng chứa xâu  $S$  và chỉ chứa các kí tự chữ cái thường.

### Kết quả

Ghi ra thiết bị ra chuẩn  $q$  dòng mỗi dòng đưa ra số  $k$  lớn nhất mà có thể tìm được một xâu  $T$  khi ghép  $k$  lần xâu  $T$  này tạo thành một xâu con của  $S$ .

### Hạn chế

- Subtask 1 (28 điểm):  $|S| \leq 1000$ .
- Subtask 2 (72 điểm):  $1000 < |S| \leq 100000$ .

### Ví dụ

Dữ liệu	Kết quả	Giải thích
3	3	Testcase 1: Xâu $T = 'a'$ nhân bản 3 lần
aaabaaaba	4	Testcase 2: Xâu $T = 'ab'$ nhân bản 4 lần
abbaaabababab	1	Testcase 3: Xâu $T$ có thể là bất kì xâu con nào và không nhân bản
abcdabd		

## Bài 3. Dãy con lượn sóng — WIGGLY

Một dãy  $X$  được gọi là dãy con của  $Y$  nếu như tồn tại cách xóa một số phần tử trong  $Y$  và giữ nguyên thứ tự ta được dãy  $X$ . Cho 2 dãy số nguyên dương  $A$  và  $B$ . Một dãy số nguyên  $C = (c_1, c_2, c_3, \dots, c_k)$  được gọi là dãy đẹp khi thỏa mãn tất cả các điều kiện sau:

- $k \geq 3$ ;
- là dãy lượn sóng, có nghĩa là  $(c_i - c_{i-1}) * (c_i - c_{i+1}) > 0 \forall i, 1 < i < k$ ;
- $C$  là dãy con của cả  $A$  và  $B$ .

**Yêu cầu:** Hãy xác định độ dài lớn nhất của dãy đẹp và số lượng dãy đẹp với độ dài lớn nhất tìm được.

### Dữ liệu

Vào từ thiết bị vào chuẩn:

- Dòng đầu tiên chứa 2 số nguyên dương  $m, n$  lần lượt là độ dài 2 dãy  $A$  và  $B$
- Dòng thứ 2 chứa  $m$  số nguyên dương  $A_1, A_2, \dots, A_m$
- Dòng thứ 3 chứa  $n$  số nguyên dương  $B_1, B_2, \dots, B_n$ .

Các số nguyên dương trong dữ liệu vào đều không vượt quá  $10^4$ .

### Kết quả

Kết quả cần ghi ra thiết bị ra chuẩn. Nếu tồn tại dãy đẹp:

- Dòng thứ nhất ghi ra một số nguyên là độ dài dãy đẹp nhất tìm được.
- Dòng thứ hai ghi ra phần dư của số lượng dãy đẹp tìm được trong phép chia cho 2019.

Nếu không tồn tại dãy đẹp thì ghi ra duy nhất một số 0.

### Hạn chế

- Subtask 1 (12 điểm):  $m, n \leq 50$ .
- Subtask 2 (16 điểm):  $50 < m, n \leq 500$ .
- Subtask 3 (20 điểm):  $500 < m, n \leq 10^4$  và các số trong từng dãy đôi một khác nhau.
- Subtask 4 (52 điểm):  $500 < m, n \leq 10^4$ .

Chương trình của bạn sẽ được 50% số điểm của test nếu ghi ra đúng độ dài dãy đẹp.

### Ví dụ

Dữ liệu	Kết quả	Giải thích
7 6 1 3 4 5 2 2 3 1 2 4 5 3 2	3 7	Có 7 dãy thỏa mãn (1,3,2); (1,4,2); (1,5,2); (1,5,3); (1,4,3); (4,5,3); (4,5,2)

## Bài 4. Cắt bánh — GATEAU

Tuấn mang tới một chiếc bánh ga-tô để chia cho các thí sinh cuộc thi SuperCoders. Chiếc bánh hình chữ nhật kích thước  $m \times n$ . Mặt bánh có thể coi là mặt phẳng với hệ tọa độ Đề các vuông góc  $Oxy$ , trong đó góc trái dưới của bánh ở tọa độ  $(0,0)$  còn góc phải trên của bánh ở tọa độ  $(m,n)$ .

Do vừa cắt bánh vừa nói chuyện nên cách thức cắt bánh của Tuấn khá lộn xộn: Tuấn đặt dao vào điểm  $P_0$  và thực hiện  $k$  bước di chuyển dao (đánh số từ 1 tới  $k$ ). Tại bước thứ  $i$ , dao di chuyển theo đoạn thẳng từ điểm  $P_{i-1}$  tới điểm  $P_i$ , di chuyển tới đâu cắt xuống chiếc bánh tới đó. Cách di chuyển dao tuân theo những ràng buộc sau:

- Tại mỗi bước, dao chỉ di chuyển theo *một trong tám hướng* tạo với tia  $Ox$  một góc có số đo độ là bội số của  $45^\circ$ .
- Trong quá trình di chuyển, *dao không đi ra khỏi chiếc bánh*. Tuy nhiên dao có thể chạm rìa bánh, di chuyển dọc rìa bánh hoặc di chuyển dọc theo một đoạn thẳng đã cắt ở những bước trước.
- Các điểm  $P_0, P_1, \dots, P_k$  có *hoành độ và tung độ đều là số nguyên*, hai điểm liên tiếp bất kỳ trong dãy điểm này không trùng nhau.

**Yêu cầu:** Cho biết chiếc bánh ga-tô bị cắt rời làm mấy mảnh và cho biết diện tích các mảnh đó.

### Dữ liệu

Dữ liệu cần được nhập từ thiết bị vào chuẩn. Dòng 1 chứa số nguyên  $T \leq 10$  là số test,  $T$  nhóm dòng tiếp theo, mỗi nhóm dòng mô tả một test theo khuôn dạng:

- Dòng 1 chứa 3 số nguyên  $m, n, k$  cách nhau bởi dấu cách.
- $k+1$  dòng tiếp theo, dòng thứ  $i$  chứa hai số nguyên cách nhau bởi dấu cách lần lượt là hoành độ và tung độ của điểm  $P_{i-1}$ ,  $\forall i : 1 \leq i \leq k+1$ .

### Kết quả

Ghi ra thiết bị ra chuẩn  $T$  nhóm dòng, mỗi nhóm dòng bao gồm:

- Dòng 1 ghi số mảnh được cắt ra  $q$ .
- Dòng 2 ghi  $q$  số thực cách nhau bởi dấu cách là diện tích các mảnh *liệt kê theo thứ tự tăng dần*. Các số thực phải được ghi chính xác với 2 chữ số sau dấu chấm thập phân.

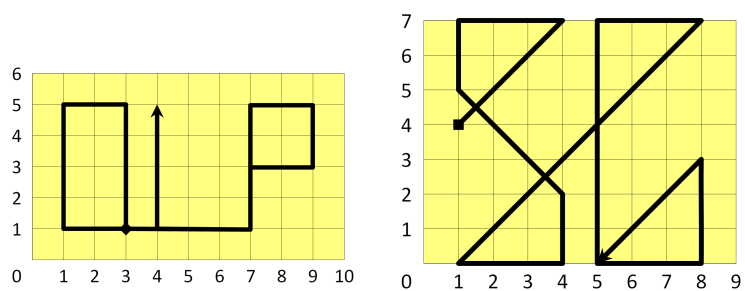
### Hạn chế

Trong tất cả các test,  $k \leq 1000$ .

- Subtask 1 (15 điểm):  $m, n \leq 1000$  và tại mỗi bước dao di chuyển theo phương song song với cạnh chiếc bánh.
- Subtask 2 (15 điểm):  $1 \leq m, n \leq 10^9$  và tại mỗi bước dao di chuyển theo phương song song với cạnh chiếc bánh.
- Subtask 3 (25 điểm):  $m, n \leq 1000$ .
- Subtask 4 (45 điểm):  $1 \leq m, n \leq 10^9$ .

## Ví dụ

Dữ liệu	Kết quả
2	3
10 6 11	4.00 8.00 48.00
3 1	8
3 5	3.75 4.25 4.25 4.50 4.50 9.50 13.25 19.00
1 5	
1 1	
7 1	
7 5	
9 5	
9 3	
7 3	
7 1	
4 1	
4 5	
9 7 12	
1 4	
4 7	
1 7	
1 5	
4 2	
4 0	
1 0	
8 7	
5 7	
5 0	
8 0	
8 3	
5 0	



Hình minh họa testcase 1 và testcase 2 trong ví dụ trên