



OLYMPIC TIN HỌC SINH VIÊN LẦN THỨ XXVI, 2017

Khởi thi: Siêu cúp

Thời gian làm bài: 180 phút

Ngày thi: 06-12-2017

Nơi thi: ĐẠI HỌC SƯ PHẠM TP HỒ CHÍ MINH

TỔNG QUAN ĐỀ THI

	Tên bài	Tên file chương trình	Tên file dữ liệu	Tên file kết quả
Bài 1	Hàm tiền tố	PREFIX.???	PREFIX.INP	PREFIX.OUT
Bài 2	Con dê	GOAT.???	GOAT.INP	GOAT.OUT
Bài 3	Xoay tròn	PIROUETTE.???	PIROUETTE.INP	PIROUETTE.OUT
Bài 4	Tìm số vắng mặt	MISSING.???	Không có	Không có

Chú ý: Dấu ??? được thay thế bởi đuôi ngầm định của ngôn ngữ được sử dụng để cài đặt chương trình.

Hãy lập trình giải các bài sau đây:

Bài 1. Hàm tiền tố

Khi học môn “Mã hóa và nén dữ liệu”, Tuấn rất hứng thú với những phương pháp mã hóa đòi hỏi ít bộ nhớ hơn nhiều so với mã hóa ACSII thông dụng. Vừa rồi, khi khảo sát những xâu chứa nhiều ký tự liên tiếp giống nhau, Tuấn phát hiện ra rằng có thể thiết lập tương ứng một-một giữa một xâu S như vậy với một dãy số nguyên dương và một dãy các ký tự. Cụ thể là: xâu S bắt đầu bởi a_1 ký tự c_1 , tiếp đến là a_2 ký tự c_2 , ..., và cuối cùng là a_n ký tự c_n :

$$S = \underset{\leftarrow a_1 \rightarrow}{c_1 c_1 \dots c_1} \underset{\leftarrow a_2 \rightarrow}{c_2 c_2 \dots c_2} \dots \underset{\leftarrow a_n \rightarrow}{c_n c_n \dots c_n}$$

được đặt tương ứng với dãy số nguyên dương $A = (a_1, a_2, \dots, a_n)$ và xâu ký tự $C = c_1 c_2 \dots c_n$.

Việc mã hóa và giải mã theo cách mã hóa như vậy là hết sức đơn giản. Hơn nữa, Tuấn cũng chỉ ra những tình huống ứng dụng mà phương pháp mã hóa này có thể đòi hỏi ít bộ nhớ. Bây giờ một vấn đề nảy sinh trong ứng dụng là cần giải quyết bài toán tìm kiếm xâu mẫu trên xâu mã hóa theo phương pháp này. Tuấn có ý định áp dụng thuật toán KMP (Knuth-Morris-Pratt), mà như đã biết vấn đề đầu tiên cần giải quyết là tính giá trị của hàm tiền tố đối với xâu S được định nghĩa như sau: Với mỗi $p = 1, 2, \dots, l$ (l là độ dài của xâu S):

$$\pi(p) = \max\{i \mid 0 \leq i < p; S[1 \dots i] = S[p-i+1 \dots p]\},$$

trong đó ký hiệu $S[u \dots v]$ để chỉ xâu con gồm các ký tự liên tiếp từ vị trí u đến vị trí v của xâu S (theo định nghĩa, nếu $u > v$ thì $S[u, v]$ là xâu rỗng).

Hãy giúp Tuấn giải quyết vấn đề đặt ra, cụ thể bạn được yêu cầu giải quyết vấn đề sau đây:

Yêu cầu: Cho dãy số A và dãy ký tự C là mã hóa của xâu S và m truy vấn p_1, p_2, \dots, p_m , hãy đưa ra giá trị của hàm tiền tố $\pi(p_i)$ với mỗi $i = 1, 2, \dots, m$.

Dữ liệu: Vào từ file PREFIX.INP:

- Dòng đầu tiên chứa số nguyên dương n, m ($n, m \leq 10^5$);
- Dòng thứ hai chứa n số nguyên dương a_1, a_2, \dots, a_n ($a_i \leq 10^9, 1 \leq i \leq n$) mô tả dãy A ;
- Dòng thứ ba là xâu C gồm n ký tự $c_1c_2\dots c_n$, các ký tự được lấy từ bảng chữ cái tiếng Anh chữ in thường được viết liên tiếp nhau, và $c_i \neq c_{i+1}, i = 1, 2, \dots, n-1$;
- Dòng cuối cùng chứa m số nguyên dương p_1, p_2, \dots, p_m ($1 \leq p_i \leq l, 1 \leq i \leq m$) là m truy vấn cần được trả lời.

Các số trên cùng dòng được ghi cách nhau bởi dấu cách.

Kết quả: Ghi ra file PREFIX.OUT trên một dòng các giá trị $\pi(p_1), \pi(p_2), \dots, \pi(p_m)$ là các câu trả lời cho các truy vấn. Các số được ghi cách nhau bởi một dấu cách.

Ví dụ:

PREFIX.INP	PREFIX.OUT
4 4	1 0 2 4
2 4 3 5	
abab	
2 5 9 11	

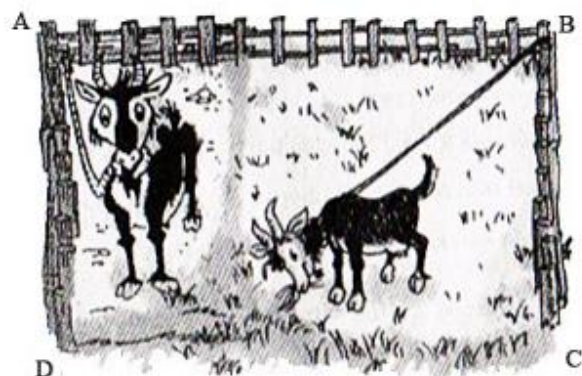
Bài 2. Con dê

Khi học chương trình Toán lớp 9 Bờm rất thích bài toán số 80 về hai con dê trong vườn cỏ. Biết vậy, để thách đố Bờm, Cuội đã cải biên bài toán đó thành bài toán chỉ với một con dê (mà thoạt nhìn, có vẻ là dễ hơn bài toán với hai con dê) sau đây:

Trên một cánh đồng cỏ rộng mênh mông (đối với bài toán ở đây hãy coi là rộng vô hạn) có một khu biệt thự được bao bọc bởi một hệ thống tường bê tông cao ngất có dạng một hình đa giác lồi gồm n đỉnh. Tại một đỉnh của đa giác có đóng một cọc mà tại đó có sợi dây với độ dài L để buộc một con dê. Con dê có thể ăn cỏ ở bất cứ điểm nào nằm ngoài đa giác mà nó có thể đến được, nhưng không thể đi vào điểm nằm trong hình đa giác (khu biệt thự). Cần tính diện tích vùng đồng cỏ mà con dê có thể ăn được.

Yêu cầu: Giúp Bờm tính diện tích vùng đồng cỏ mà con dê có thể ăn được.

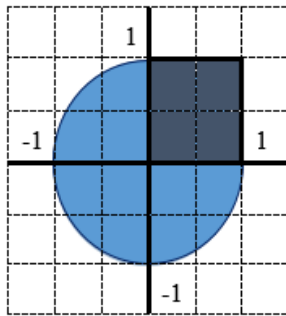
Dữ liệu: Vào từ file GOAT.INP:



- Dòng đầu tiên chứa 3 số nguyên n t L được ghi cách nhau bởi dấu cách, trong đó n là số đỉnh của đa giác, t là chỉ số đỉnh của đa giác mà dê được buộc vào cọc đóng tại đó, còn L là độ dài của sợi dây buộc dê ($1 \leq n \leq 10000$, $1 \leq L \leq 100000$);
- Dòng thứ i trong số n dòng tiếp theo chứa hai số nguyên x_i, y_i là tọa độ của đỉnh thứ i của đa giác, $|x_i| \leq 10000$, $|y_i| \leq 10000$, $i = 1, 2, \dots, n$. Các đỉnh được liệt kê theo một chiều đi vòng quanh đa giác.

Kết quả: Ghi ra file GOAT.OUT diện tích của vùng đồng cỏ mà con dê có thể ăn được với 8 chữ số sau dấu phẩy.

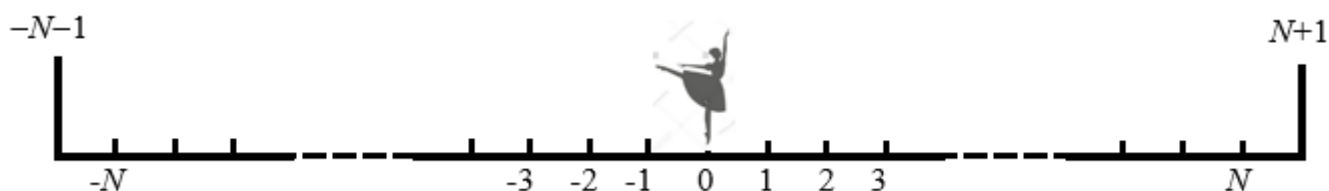
Ví dụ:

GOAT . INP	GOAT . OUT	Hình vẽ minh họa cho ví dụ
4 1 1 0 0 1 0 1 1 0 1	2.35619449	

Giải thích: Trong hình vẽ minh họa cho ví dụ: Vùng tô màu đen là mô tả khu biệt thự, vùng tô màu xanh (tô màu đen nhạt – trong bản in đen-trắng) là vùng cánh đồng cỏ mà con dê có thể ăn được.

Bài 3. Xoay tròn

Cho một số nguyên dương N và một căn phòng có chiều dài $2*N + 2$ được xét như một đoạn thẳng đóng $[-N-1, N + 1]$. Ban đầu vũ nữ ba lê Mathilde Froustey đứng ở điểm trung tâm C của căn phòng ($C = 0$). Cô ấy phải thực hiện T bước nhảy có độ dài 1, bước đầu tiên hướng về phía bên phải. Trong số $2*N$ điểm khác nhau có tọa độ nằm bên trong căn phòng có đánh dấu K điểm xoay khác nhau. Khi vũ nữ ba lê đạt đến một điểm xoay, cô sẽ dừng lại và thực hiện động tác xoay tròn. Động tác này sẽ làm thay đổi hướng di chuyển của vũ nữ, và điểm xoay ở điểm đó sẽ bị xóa bỏ.



Hình vẽ minh họa

Không được đặt điểm xoay nào tại các điểm với tọa độ $-N-1$, 0 và $N + 1$. Hơn nữa, do các điểm với tọa độ $-N-1$ và $N + 1$ là ở sát với tường của căn phòng nên chúng là các điểm xoay vĩnh cửu, nghĩa là chúng sẽ không bị xóa khi vũ nữ di chuyển qua, còn điểm với tọa độ $C = 0$ là vị trí xuất phát của Mathilde (xem hình vẽ minh họa ở trên).

Yêu cầu: Cho các giá trị của T , N và K , hãy tính xem có bao nhiêu cách đánh dấu K điểm xoay khác nhau để sau khi thực hiện đúng T bước di chuyển, Mathilde quay trở về điểm bắt đầu C .

Dữ liệu: Vào từ file PIROUETTE.INP gồm một dòng chứa ba số nguyên T , N và K được ghi cách nhau bởi dấu cách có ý nghĩa như đã nêu ở trên ($0 \leq T \leq 200$; $1 \leq N \leq 100$; $0 \leq K \leq 2 \cdot N$).

Kết quả: Ghi ra file PIROUETTE.OUT một số nguyên là phần dư trong phép chia của giá trị tìm được cho $1000000007 (10^9+7)$.

Ví dụ:

PIROUETTE.INP	PIROUETTE.OUT
6 3 4	7

Giải thích: Vũ nữ phải thực hiện $T = 6$ bước nhảy trong căn phòng dạng đoạn thẳng độ dài $2 \cdot N + 2 = 2 \cdot 3 + 2 = 8$, và phải di chuyển qua $K = 4$ điểm xoay. Có 7 cách đặt điểm xoay thỏa mãn yêu cầu đầu bài:

1. [. x . C x x x] 2. [. x x C . x x] 3. [x . x C . x x]
4. [x x . C x . x] 5. [x x . C x x .] 6. [x x x C . x .] 7. [x x x C . . x]

Ký hiệu '.' là vị trí tự do, 'x' là vị trí có điểm xoay, còn 'C' là vị trí xuất phát.



Xoay tròn (Pirouette): Là một động tác trong múa ba lê đòi hỏi vũ công thực hiện xoay vòng toàn thân trên ngón chân hoặc bàn chân.

Mathilde Froustey với biệt danh "cô bé Pháp" - vũ công ballet nổi tiếng thế giới có bà ngoại là người gốc Việt. Cô đã từng biểu diễn ở Việt nam năm 2016, trước đó năm 2008 cô đã có 3 tháng du lịch ở Việt nam.

Bài 4. Tìm số vắng mặt

Alice và Bob rủ nhau chơi trò chơi tìm các số vắng mặt sau đây. Từ các phần tử của dãy số $1, 2, \dots, n$ ($2 \leq n \leq 1000$), Alice xây dựng tập X gồm m số a_1, a_2, \dots, a_m . Nhiệm vụ của Bob là tìm ra $n-m$ số trong dãy $1, 2, \dots, n$ không có mặt trong tập các số X mà Alice đang giữ, bằng cách yêu cầu Alice trả lời câu hỏi có dạng sau đây:

“Bit thứ j trong biểu diễn nhị phân của số a_i là bao nhiêu?”

Câu trả lời của Alice cho câu hỏi sẽ là 0 hoặc 1. Lưu ý là các bit của một số nguyên k được đánh số từ 0 đến $\lfloor \log_2(k) \rfloor$, từ phải qua trái.

Yêu cầu: Hãy giúp Bob tìm ra $n-m$ số không có mặt trong tập gồm m số mà Alice đang giữ với số lượng câu hỏi cần yêu cầu Alice trả lời càng ít càng tốt.

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

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

Thư viện cung cấp các hàm sau:

- **Hàm khởi tạo trò chơi**

➤ Đối với Java:

```
public static int get_n();
public static int get_m();
```

➤ Đối với C++:

```
int get_n();
int get_m();
```

Chương trình của bạn phải gọi các hàm này để khởi tạo trò chơi và lấy giá trị cho 2 biến n và m .

- **Hàm thực hiện truy vấn**

```
public static int ask(int i, int j); // đối với Java,
int ask(int i, int j);             // đối với C++.
```

Hàm này trả về 1 nếu bit thứ j của số a_i bằng 1 và trả về 0 nếu ngược lại.

- **Hàm trả lời câu hỏi**

```
public static void guess(ArrayList<Integer> res); // đối với Java,
void guess(vector<int> res);                     // đối với C++.
```

Để kết thúc, chương trình của bạn cần gọi hàm này với `res` là kiểu vector trở đến $n-m$ số tìm được đã được sắp xếp theo thứ tự tăng dần của giá trị. Sau khi gọi hàm này chương trình sẽ tự động kết thúc. Số lượng câu hỏi của chương trình của bạn sẽ bằng tổng số lần gọi hàm `ask(i, j)`.

Lưu ý:

- Mỗi hàm `get_n`, `get_m` và `guess` chỉ được gọi một lần duy nhất.
- Đối với Java bạn phải cài đặt hàm `run()` để gọi các hàm được cung cấp sẵn, xử lý và khi đã xác định được đáp án cần gọi hàm `guess` để đoán kết quả. Cần khai báo class public tên `missing` và khai báo hàm `run()`. Cần chắc chắn hàm `run()` được để public:

```
public class missing{
    public void run(){
        // your code here
    }
}
```

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

Subtask 1: có 30% số test ứng với $m = n-1$; $a_1 < a_2 < \dots < a_m$.

Subtask 2: có 30% số test ứng với $m = n-2$;

Subtask 3: 40% số test còn lại ứng với $0 \leq m \leq n$ và $m \neq n-1$.

Ví dụ:

Gọi hàm	Giá trị trả về
<code>get_n()</code>	2
<code>get_m()</code>	1
<code>ask(1,1)</code>	1
<code>guess(res) //</code> với $res = \langle 2 \rangle$	Kết thúc chương trình. Bạn đã trả lời đúng với số lượng câu hỏi là 1 và chương trình đạt điểm của ví dụ này.

----- HẾT -----