



# OLYMPIC TIN HỌC SINH VIÊN LẦN THỨ XIV, 2005

## Khối thi: Tập thể “Lều chõng” Chuyên Tin học

Thời gian làm bài: 180 phút

Ngày thi: 24-04-2005

Nơi thi:

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH**

Tên bài	Tên file chương trình	Tên file dữ liệu	Tên file kết quả	Hạn chế thời gian
Ô CHỮ	PUZZLE.EXE	PUZZLE.INP	PUZZLE.OUT	2s
BỘ BA	BOBA.EXE	BOBA.INP	BOBA.OUT	5s
ROBOT	ROBOT.EXE	ROBOT.INP	ROBOT.OUT	3s
GIÁ TRỊ LỚN NHẤT	MAX.EXE	MAX.INP	MAX.OUT	10s

**Nộp chương trình được dịch dưới dạng file EXE.**

**Hãy lập trình giải các bài sau đây:**

### Bài 1. Ô chữ

*Tên chương trình: PUZZLE.EXE*

Một trò chơi ô chữ được mô tả như sau. Cho một bảng có  $n$  hàng và  $m$  cột. Trong mỗi ô của bảng có một chữ cái. Bạn được cho một danh sách các từ. Trước hết hãy tìm ra các từ này trong bảng đã cho. Các từ này được cấu tạo bằng các ô liên kề trong bảng theo chiều trên xuống hoặc chiều trái sang phải. Các ô liên kề này được đánh dấu. Một ô đã được đánh dấu vẫn có thể được sử dụng để tạo thành một từ khác. Sau khi đánh dấu những ô tạo thành các từ trong danh sách đã cho, các ô không được đánh dấu tạo thành một từ khóa (đọc từ trên xuống và trái sang phải).

**Yêu cầu:** Cho bảng, danh sách các từ, hãy tìm từ khóa.

**Dữ liệu:** vào từ file văn bản PUZZLE.INP, gồm nhiều Tests:

- Dòng đầu tiên chứa số nguyên  $T$  - số lượng Tests ( $1 \leq T \leq 10$ ),
- Các dòng tiếp theo đưa ra thông tin về các Tests, với mỗi test:
  - Dòng đầu tiên chứa hai số nguyên  $n$  và  $m$  ( $0 < n, m \leq 20$ ), cách nhau bởi dấu trống,
  - Dòng thứ hai chứa số  $k$  ( $0 < k \leq 20$ ), là số lượng từ trong danh sách,
  - $k$  dòng tiếp theo mỗi dòng chứa một từ, mỗi từ không quá 20 ký tự.
  - $n$  dòng cuối: mỗi dòng chứa một xâu  $m$  ký tự mô tả một dòng của bảng ô chữ theo thứ tự từ trên xuống dưới.

**Kết quả:** Đưa ra file văn bản PUZZLE.OUT, gồm T dòng, mỗi dòng chứa xâu ký tự là từ khóa cần tìm. Dòng thứ  $i$  chứa kết quả tương ứng với Test thứ  $i$  ( $i = 1 \div T$ ).

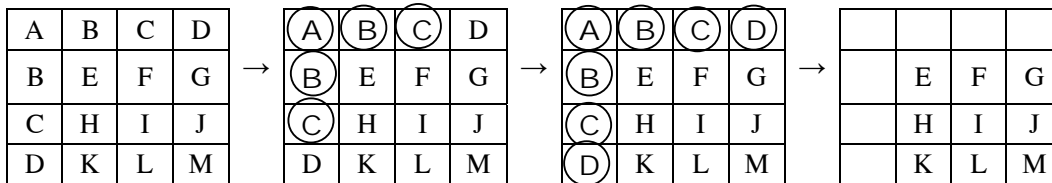
**Ví dụ:**

PUZZLE.INP
1
4 4
2
ABC
ABCD
ABCD
BEFG
CHIJ
DKLM

PUZZLE.OUT
EFGHIJKLM

Ví dụ trên tương ứng với quá trình tìm kiếm sau:

Ô chữ ban đầu                      Sau khi tìm và đánh dấu từ ABC                      Sau khi tìm và đánh dấu từ ABCD                      Các ký tự còn lại



## Bài 2. Bộ ba

Cho một xâu văn bản chỉ gồm các chữ cái latin in hoa. Ta gọi một *bộ ba* là dãy gồm ba chữ cái la tinh in hoa liên tiếp trong xâu.

**Yêu cầu:** Hãy tìm số lần xuất hiện của bộ ba xuất hiện nhiều nhất trong xâu văn bản đã cho.

**Ví dụ:**

- a) Trong xâu văn bản ‘DAIHOCKHOAHOCTUNHIENTHANHPHOHOCHIMINH’ xâu con ‘HOC’ xuất hiện nhiều nhất (3 lần).
- b) Trong xâu văn bản ‘AAAAAD’ xâu ‘AAA’ xuất hiện nhiều nhất (3 lần).

**Dữ liệu:** Vào từ file văn bản BOBA.INP gồm nhiều Tests:

- Dòng đầu tiên chứa số nguyên T - số lượng Tests ( $1 \leq T \leq 10$ ),
- T dòng tiếp theo đưa ra thông tin về các Tests, mỗi dòng chứa thông tin về một test, gồm chứa xâu văn bản gồm không quá  $10^6$  chữ cái la tinh in hoa.

**Kết quả:** Ghi ra file văn bản BOBA.OUT gồm T dòng, mỗi dòng chứa một số nguyên. Dòng thứ  $i$  chứa kết quả tương ứng với Test thứ  $i$  ( $i = 1 \div T$ ) là số lần xuất hiện nhiều nhất tìm được.

Ví dụ:

BOBA . INP
2
AAAAAD
BCDEGHDJL

BOBA . OUT
3
1

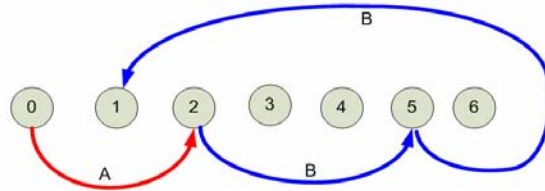
### Bài 3. ROBOT

**Tên chương trình:ROBOT. EXE**

Để chuẩn bị cho cuộc thi *Robocon* các đội thi phải tự thiết kế và lắp ráp robot riêng cho mình. Nội dung cuộc thi là xuất phát từ vị trí I robot phải đi tới vị trí J, làm một số công việc theo quy định của cuộc thi. Các vị trí được đánh dấu từ 0 đến N-1. Những công việc cần làm này khá đơn giản và các mô đun tương ứng đã được lắp đặt hoàn hảo. Chỉ còn lại hai khó khăn “nhỏ” phải vượt qua:

- Robot phải rất gọn nhẹ, có độ tin cậy cao để đảm bảo chuyển động an toàn từ vị trí này tới vị trí khác vì số vị trí trong danh sách cho trước là rất lớn,
- Lấy kinh phí và thời gian ở đâu để thiết kế, mua linh kiện, lắp ráp và thử nghiệm.

Nói một cách chính xác, chỉ có phần thứ nhất trong khó khăn thứ hai mới thực sự làm mọi thành viên trong đội phải đau đầu. Một bạn nữ đến cổ động càng làm cho mọi người sôi lên khi phát biểu một cách mát mẻ: “*Có thể cảm thông được với các bạn đó vì đâu có phải ai cũng là Abramovic hay Bill Gate!*”. Nhưng chính câu nói đó đã làm loé sáng một giải pháp tuyệt vời: Ta không giàu như Bill Gate, nhưng cũng có cái đầu như Bill Gate và không cần phải chế tạo một robot quá phức tạp, mà thay vào đó, trang bị một phần mềm hiệu quả để khai thác tối đa khả năng kỹ thuật hiện có như các hệ điều hành WINDOWS làm với máy IBM PC.



Các bạn đang có trong tay một bộ thiết bị rất đáng tin cậy thiết kế từ các hợp đồng NCKH trong năm với 2 lệnh điều khiển: lệnh A dẫn dắt robot từ một vị trí I bất kỳ đến vị trí  $(I+K) \bmod N$  và lệnh B dẫn dắt robot từ vị trí I đến vị trí  $(I+L) \bmod N$ . Như vậy, có thể chỉ ra một số lượng thích hợp các lệnh A và B, có thể dẫn dắt robot từ vị trí I tới vị trí J. Vấn đề còn lại chỉ là có thể phải thiết kế thêm bộ phận chỉnh lý khi không thể bằng các lệnh A và B đưa robot từ I tới J, nhưng đó là vấn đề đơn giản vì ta có thể đưa robot tới vị trí gần J nhất và giải quyết tìm kiếm trong phạm vi nhỏ.

**Yêu cầu:** Cho biết N, K, L, I, J ( $0 < N < 32\ 767$ ,  $0 < K, L < N$ ,  $0 \leq I, J < N$ ). Hãy xác định số lần ít nhất thực các lệnh A, B để đưa được robot từ I tới J hoặc cho biết là không thể thực hiện được yêu cầu đề ra.

**Dữ liệu:** Vào từ file văn bản ROBOT.INP gồm nhiều Tests:

- Dòng đầu tiên chứa số nguyên T - số lượng Tests ( $1 \leq T \leq 10$ ),
- T dòng tiếp theo đưa ra thông tin về các Tests, mỗi test trên một dòng gồm 5 số nguyên N K L I J.

**Kết quả:** Đưa ra file văn bản ROBOT.OUT, gồm T dòng, mỗi dòng chứa một số nguyên. Dòng thứ i chứa kết quả tương ứng với Test thứ i ( $i = 1 \div T$ ) là số lần ít nhất thực hiện lệnh. Nếu không thể đưa robot tới J được thì đưa ra số -1.

Ví dụ:

ROBOT.INP
2
7 2 3 0 1
8 2 4 5 6

ROBOT.OUT
3
-1

#### Bài 4. Giá trị lớn nhất

*Tên chương trình: MAX. EXE*

Cho hàm **F** với giá trị ban đầu bằng 1 và xâu *S* độ dài *L* không quá 40 gồm các ký tự chữ cái la tinh in hoa. Mỗi ký tự là một biến, các ký tự khác nhau tương ứng với những biến khác nhau. Mỗi biến *C* nhận một giá trị nguyên  $G(C)$  trong phạm vi từ 1 tới  $N$  ( $1 \leq N \leq 10\,000$ ), trong đó *C* là một ký tự có mặt trong *S*.

Giá trị của **F** được tính dựa vào *S* theo quy tắc sau: Với mọi *I* thay đổi từ 1 đến *N* giá trị của **F** có thể biến đổi theo một trong hai cách:

- Không thay đổi,
- Nhận giá trị mới là  $(F * G(S[I])) \bmod N$ , trong đó  $S[I]$  – ký tự thứ *I* trong *S*, *mod* – phép lấy mô đun.

Ví dụ, với  $S = \text{'ABAB'}$  và  $G(A) = 2$ ,  $G(B) = 3$ ,  $N = 5$ , ta có thể tính **F** bằng nhiều cách, 2 trong số các cách tính đó là như sau:

I	Giá trị của <b>F</b>	
	Cách 1	Cách 2
1	$1 * G(A) \bmod N =$ $1 * 2 \bmod 5 = 2$	$1 * G(A) \bmod N =$ $1 * 2 \bmod 5 = 2$
2	$2 * G(B) \bmod N =$ $2 * 3 \bmod 5 = 1$	Giữ nguyên giá trị (=2)
3	$1 * G(A) \bmod N =$ $1 * 2 \bmod 5 = 2$	$2 * G(A) \bmod N =$ $2 * 2 \bmod 5 = 4$
4	$2 * G(B) \bmod N =$ $2 * 3 \bmod 5 = 1$	Giữ nguyên giá trị (=4)
Giá trị cuối cùng	1	4

**Yêu cầu:** Hãy tìm giá trị lớn nhất mà **F** có thể nhận được.

**Dữ liệu:** Vào từ file văn bản MAX.INP: Gồm nhiều Tests:

- Dòng đầu tiên chứa số nguyên *T* - số lượng Tests ( $1 \leq T \leq 10$ ),
- $3 * T$  dòng tiếp theo đưa ra thông tin về các Test, mỗi test trên 3 dòng:
  - Dòng đầu tiên chứa số nguyên dương *N*,
  - Dòng thứ 2 chứa xâu *S*,
  - Dòng thứ 3 chứa 26 số nguyên dương là giá trị  $G(A), G(B), \dots, G(Z)$ , các số cách nhau ít nhất một dấu cách.

**Kết quả:** Đưa ra file văn bản MAX.OUT gồm *T* dòng, mỗi dòng chứa một số nguyên. Dòng thứ *i* chứa kết quả tương ứng với Test thứ *i* ( $i = 1 \div T$ ).

Ví dụ:

MAX . INP																	
1																	
5																	
ABAB																	
2	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1															

MAX . OUT
4