



# Printer Queue

**Link submit:**

[https://uva.onlinejudge.org/index.php?option=com\\_onlinejudge&Itemid=8&page=show\\_problem&problem=3252](https://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=3252)

**Solution:**

C++	<a href="http://ideone.com/UkYHku">http://ideone.com/UkYHku</a>
Java	<a href="https://ideone.com/aiRI7U">https://ideone.com/aiRI7U</a>
Python	<a href="https://ideone.com/ugbXfB">https://ideone.com/ugbXfB</a>

**Tóm tắt đề:**

Chỉ có duy nhất một cái máy in dành cho sinh viên ở trường, thỉnh thoảng có đến hàng trăm tác vụ in ấn dành cho máy nên mọi người phải chờ hàng giờ. Bởi vì có những tác vụ đặc biệt quan trọng nên Hacker General đã thiết kế một hệ thống hàng đợi ưu tiên để giải quyết các công việc in ấn, theo đó mỗi tác vụ có một con số từ 1 đến 9. Số càng lớn thì công việc càng được ưu tiên xử lý trước.

Bạn có  $n$  tác vụ in ấn cho trước và vị trí công việc của bạn, hãy xác định xem mất bao nhiêu lâu để chờ tới tác vụ của bạn được thực hiện. Quy tắc thực hiện các tác vụ như sau:

- Trong  $n$  tác vụ chọn tác vụ lấy 1 tác vụ ra trên đầu hàng đợi ra, nếu tác vụ đó không phải là tác vụ ưu tiên lớn nhất thì đem tác vụ đó vào cuối hàng đợi. Nếu lấy ra đó là tác vụ ưu tiên cao nhất thì thực hiện tác vụ đó và loại tác vụ đó khỏi hàng đợi.
- Cứ tiếp tục làm như vậy cho đến khi gặp được tác vụ của bạn. Giả sử mỗi tác vụ thực hiện trong 1 phút bạn hãy tính xem tác vụ của bạn được thực hiện ở phút thứ bao nhiêu.

**Input:**

Dòng đầu tiên chứa số lượng bộ test (không lớn hơn 100). Mỗi bộ test có định dạng như sau:

- Dòng đầu chứa hai số  $n$  và  $m$  ( $1 \leq n \leq 100$ ,  $0 \leq m \leq n - 1$ ), với  $n$  là số lượng tác vụ,  $m$  là vị trí tác vụ của bạn trong hàng đợi (hàng đợi vị trí đầu tiên là 0).
- Dòng tiếp theo chứa  $n$  số, mỗi số mang giá trị từ 1 đến 9 là mức độ ưu tiên của tác vụ trong hàng đợi.

**Output:**

Mỗi bộ test in ra một số duy nhất là số phút mà tác vụ của bạn được hoàn thành.

**Ví dụ:**

3	1
1 0	2
5	5
4 2	
1 2 3 4	
6 0	
1 1 9 1 1 1	

**Giải thích ví dụ:**

Ví dụ gồm 3 bộ test.

*Bộ 1:* có 1 tác vụ, đó cũng chính là tác vụ của bạn nên kết quả là 1.

*Bộ 2:* có 4 tác vụ, tác vụ của bạn nằm ở vị trí số 2 mang giá trị 3.

Đầu tiên lấy số 1 ra, đây không phải là tác vụ ưu tiên nhất nên bỏ lại vào hàng đợi 2 3 4 1.

Lấy tác vụ 2 ra, đây không phải là tác vụ ưu tiên nhất nên bỏ lại hàng đợi 3 4 1 2.

Lấy tác vụ 3 ra, đây cũng không phải là tác vụ ưu tiên nhất 4 1 2 3.

Lấy tác vụ 4 ra, đây là tác vụ ưu tiên nhất nên xử lý nó → mất 1 phút.

Lấy 1 ra lại thì hàng đợi lúc này 2 3 1

Lấy 2 ra lại thì hàng đợi lúc này 3 1 2.

Lấy 3 ra và xử lý tác vụ này → mất 1 phút, đây cũng là tác vụ của bạn, vậy nên để hoàn thành tác vụ của bạn tốn 2 phút.

**Hướng dẫn giải:**

Đầu tiên để toàn bộ các tác vụ vào trong hàng đợi và trong mảng một chiều.

Sắp xếp mảng một chiều lại tăng dần, việc sắp xếp này để luôn biết được phần tử có độ ưu tiên lớn nhất để xử lý hiện tại có giá trị là bao nhiêu.

Bạn đem từng giá trị trong queue ra xét:

- Nếu nó trùng với giá trị max, nghĩa là tác vụ này sẽ được xử lý. Xóa giá trị này khỏi queue. Đồng thời tăng biến đếm lên vì đã hoàn thành một tác vụ.
- Nếu giá trị không trùng với giá trị max. Thì bạn cho nó vào lại trong queue, lúc này nó sẽ nằm ở cuối hàng đợi.

Trong lúc đem giá trị trong queue ra xét thì bạn cũng xét xem giá trị đó có phải là tác vụ của bạn cần xử lý không, nếu đúng là tác vụ của bạn thì bạn sẽ dừng lại và in ra biến đếm vì đây là thời gian bạn cần chờ in tác vụ của mình.

**Độ phức tạp:**  $O(T * N \log N)$  với T là số lượng bộ test, N là số lượng tác vụ của từng bộ test.