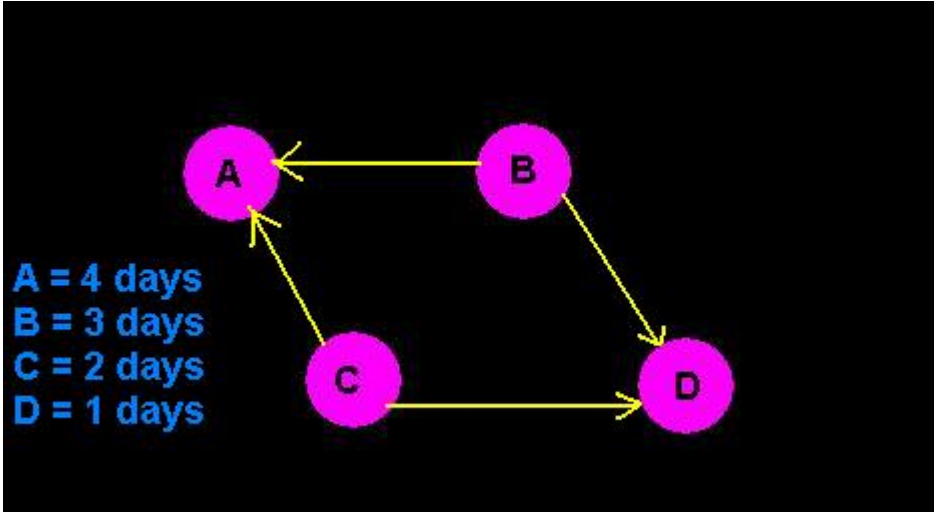


You are given a list of jobs with associated time to complete them. Also are given a list of job dependencies. Your job is to calculate the maximum possible difference among all possible completion time of a particular job. For this problem, job dependency is a bit simpler. When a job depends on several other jobs to be completed, finishing of all of them will allow it to execute. Furthermore, jobs cannot be executed in parallel and there will be no idle moment.



In the above example job-B cannot be started before A and D. So it can be started after 5 days and it must be started after 7 days. Here for job-B, the difference is 2 days.

### Input

Input starts with a pair of integers  $v$  ( $1 \leq v \leq 500$ ) and  $e$  ( $0 \leq e \leq 500$ ) where  $v$  represents the number of jobs and  $e$  represents the number of dependencies in the dependency list. Following is a line with  $v$  integers each indicating the time in days to complete the jobs where the  $i$ 'th integer denotes the completion time of the  $i$ 'th job. The jobs are numbered by integers from 1 to  $v$ . Next  $e$  lines has the form ' $x y$ ' which means that job  $x$  should be completed prior to job  $y$  ( $1 \leq x, y \leq v$ ). Next line has an integer  $q$  which denotes the number of queries to answer which is followed by  $q$  integers  $x$  ( $1 \leq x \leq v$ ). A line with  $v = e = 0$  ends the input session. Every block will be followed by a blank line.

There will be no impractical situation (each job can be completed) in input.

### Output

For each query in a block output the result according to the problem description in a separate line. A blank line is essential after every block of data. See the sample output.

### Sample Input

```

4 4
4 3 2 1
2 1
2 4
3 1
3 4
2
1 2

4 4
4 3 2 1
2 1
2 4
3 1
3 4
2
3 4

0 0
  
```

### Sample Output

```

Case #1:
1
2

Case #2:
3
4
  
```