

You are a waiter at a party. There is a pile of numbered plates. Create an empty *answers* array. At each iteration, *i*, remove each plate from the top of the stack in order. Determine if the number on the plate is evenly divisible by the *i<sup>th</sup>* prime number. If it is, stack it in pile *B<sub>i</sub>*. Otherwise, stack it in stack *A<sub>i</sub>*. Store the values in *B<sub>i</sub>* from top to bottom in *answers*. In the next iteration, do the same with the values in stack *A<sub>i</sub>*. Once the required number of iterations is complete, store the remaining values in *A<sub>i</sub>* in *answers*, again from top to bottom. Return the *answers* array.

Example

*A* = [2, 3, 4, 5, 6, 7]

*q* = 3

An abbreviated list of primes is [2, 3, 5, 7, 11, 13]. Stack the plates in reverse order.

*A*<sub>0</sub> = [2, 3, 4, 5, 6, 7]

*answers* = []

Begin iterations. On the first iteration, check if items are divisible by 2.

*A*<sub>1</sub> = [7, 5, 3]

*B*<sub>1</sub> = [6, 4, 2]

Move *B*<sub>1</sub> elements to *answers*.

*answers* = [2, 4, 6]

On the second iteration, test if *A*<sub>1</sub> elements are divisible by 3.

*A*<sub>2</sub> = [7, 5]

*B*<sub>2</sub> = [3]

Move *B*<sub>2</sub> elmements to *answers*.

*answers* = [2, 4, 6, 3]

And on the third iteration, test if *A*<sub>2</sub> elements are divisible by 5.

*A*<sub>3</sub> = [7]

*B*<sub>3</sub> = [5]

Move *B*<sub>2</sub> elmements to *answers*.

*answers* = [2, 4, 6, 3, 5]

All iterations are complete, so move the remaining elements in *A*<sub>3</sub>, from top to bottom, to *answers*.

*answers* = [2, 4, 6, 3, 5, 7]. Return this list.

Function Description

Complete the waiter function in the editor below.

waiter has the following parameters:

- int number[n]: the numbers on the plates
- int q: the number of iterations

Returns

- int[n]: the numbers on the plates after processing

Input Format

The first line contains two space separated integers, *n* and *q*.

The next line contains *n* space separated integers representing the initial pile of plates, i.e., *A*.

Constraints

$1 \leq n \leq 5 \times 10^4$   
 $2 \leq number[i] \leq 10^4$   
 $1 \leq q \leq 1200$

Sample Input

```
5 1
3 4 7 6 5
```

Sample Output

```
4
6
5
7
3
2
```

Change ThemeLanguagePython 3

```
16 #
17 def getQprimes(q):
18     n = 2
19     count = 0; out = []
20     while(count < q):
21         isPrime = True
22         for i in range(2,n//2+1):
23             if n%i == 0:
24                 isPrime = False
25                 break
26         if isPrime:
27             out.append(n)
28             count += 1
29         n += 1
30     return out
31
32
33 def waiter(number, q):
34     primes = getQprimes(q); answer = []
35     # Write your code here
36     for i in primes:
37         B = []; A = []
38         for j in reversed(number):
39             if j%i == 0:
40                 B.append(j)
41             else:
42                 A.append(j)
43         number = A
44         answer += reversed(B)
45
46     answer += reversed(A)
47     return answer
48
```

Line: 26 Col: 20

Upload Code as FileTest against custom input

Run CodeSubmit Code

Congratulations

You solved this challenge. Would you like to challenge your friends?

Test case 6

Test case 7

Test case 8

Test case 9

Test case 10

Test case 11

Test case 12

Compiler Message

Success

Input (stdin)

|   |           |
|---|-----------|
| 1 | 5 1       |
| 2 | 3 4 7 6 5 |

Expected Output

|   |   |
|---|---|
| 1 | 4 |
| 2 | 6 |
| 3 | 3 |
| 4 | 7 |