**Stacking Plates**

You work at a plate stacking factory and they need a program that stacks the plates in a nice fashion using prime numbers. The way it works is that there will be a set number of iterations that you will sort the plates by. You will group the plates into groups **A** and **B**.

In the first iteration, you will compare all the plates in the stack to the first prime number in the prime number sequence, which is 2 (this pattern will continue as 3, 5, 7, 11, 13…). If it is divisible by the current prime number, it goes into stack **B**, otherwise it goes into stack **A**. You will then store the values of stack **B** into the “final” stack.

In the subsequent iterations, you will continue to do the same process with the elements of stack **A**. You will repeat this until all required iterations are complete. If after the last iteration there are still plates in stack **A**, you will put the remaining plates into the “final” stack. You will then print out the “final” stack from bottom to top.

**Input:** The first line of input contains **N**, the number of elements on the second line and **Q**, the number of iterations you will sort the plates by. The second line contains **N** space-separated integers which represent a plate and the size of the plate.

**Output:** The “final” stack printed from bottom to top.

**Example Input:**

5 2

3 3 4 4 9

**Example Output:**

4

4

9

3

3

**Explanation:** Since Q = 2, we will iterate only two times.

Initially, the stack of plates is (bottom to top of stack): [3, 3, 4, 4, 9]. We then check the divisibility of all the integers by the first prime number which is 2. So stack A and B look like:

A = [9, 3, 3] and B = [4, 4]. So the final answer appears as [4, 4] currently. In the second iteration, we then compare the divisibility of the plates of A to the second prime number, which is 3. We then get: A = [] and B = [3, 3, 9]. We then move all elements of B into the answer stack, which then becomes [4, 4, 9, 3, 3] (with 3 being the top of the stack).

We then output the stack from bottom to top which returns us 4, 4, 9, 3, 3. This is the final answer.