Finals 2017

sponsored by:

Problem 7: Debug

Seán is trying to debug a piece of his code. First he creates an array of **N** integers and fills it with zeros. Then he repeatedly calls the following procedure which he has written in C++:

```
void something( int jump ) {
  int i = 0;
  while( i < N ) {
    seq[i] = seq[i] + 1;
    i = i + jump;
  }
}
```

As you can see, this procedure increases by one all elements in the array whose indices are divisible by `jump`.

Seán calls the procedure exactly **K** times, using the sequence $\mathbf{X_1}$ $\mathbf{X_2}$ $\mathbf{X_3}$ **...** $\mathbf{X_k}$ as arguments.

After this, Seán has a list of **Q** special parts of the array he needs to check to verify that his code is working as it should be. Each of this parts is defined by two numbers, **L** and **R** (**L** ≤ **R**) the left and right bound of the special part. To check the code, Seán must compute the sum of all elements of `seq` between and including **L** and **R**. In other words `seq`[**L**] + `seq`[**L**+1] + `seq`[**L**+2] + … + `seq`[**R**]. Since he needs to know the answer in advance in order to check it, he asked you to help him.

Input

The first line of input contains two integers, **N** ($1 \le N \le 10^6$), size of the array, and **K** ($1 \le K \le 10^6$), number of calls to `something` Seán makes.The second line contains **K** integers: $\mathbf{X_1}$ $\mathbf{X_2}$ $\mathbf{X_3}$ **...** $\mathbf{X_k}$, arguments passed to the procedure. ($1 \le X_i < N$).

Next line contains one integer **Q** ($1 \le Q \le 10^6$), number of special parts of the array Seán needs to check.

Next **Q** lines contain two integers each $\mathbf{L_i}$ and $\mathbf{R_i}$ ($0 \le L_i \le R_i < N$), bounds of each special part.

Output

The output should contain exactly **Q** lines. The i^{th} line should contain the sum of the elements `seq`[**L**$_i$] + `seq`[**L**$_i$ +1] + `seq`[**L**$_i$ +2] + … + `seq`[**R**$_i$].

Examples

Input Example 1	Input Example 2	Input Example 3

10 4
1 1 2 1
3
0 9
2 6
7 7
	11 3
	3 7 10
	3
	0 10
	2 6
	7 7
		1000000 6
		12 3 21 436 2 19
		2
		12 16124
		692 29021

Output Example 1	**Output Example 2**	**Output Example 3**
35 18 3	8 2 1	16422 28874

Example 1 description: The procedure is called with arguments 1, 1, 2, 1. After that the array contains values {4, 3, 4, 3, 4, 3, 4, 3, 4, 3}. Sum of indices 2 to 6 (inclusive) is 4+3+4+3+4 = 18.

Example 2 description: After the procedure calls, the array is {3, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1}.