**SimpleOne**Problem Code: **ARGCD**

******

All submissions for this problem are available.

Instead of making a long story , Coming to the question directly . You are given an array **A**of size**N**, what you have to do , is to list out all the **subarrays** of the array , take the gcd's of each subarrays and push back the result into some container(1 indexed) . Now Q queries will be given to you , in each query you will be given a number**X**, you have to answer the **Xth**smallest number from the container.

**Input**

 In the first line you will be given , two numbers **N**and **Q** In the second line you will be given**N**numbers , the array elements. The next **Q**lines contain 1 query each, a number**X**as described above.

**Output**

* For each Query, output a single line containing the required answer .

**Constraints**

* **1** ≤ **N** ≤ **20000**
* **1** ≤ **Q** ≤ **100000**
* **1** ≤ **X** ≤ **( N \*( N+1 ) ) / 2**
* **0** ≤ **A[i]** ≤ **2^63-1**

**Subtask#1   (20 Points)**

* **1** ≤ **N** ≤ **2000**

Other tasks remains Same as Original . 

**Subtask#2   (80 Points)**

Original Constraints .

**Example**

**Input:**

3 3

1 3 6

1

5

6

**Output:**

1

3

6

**Explanation**

Subarray's GCD is defined as

GCD(a,b,c,d) = GCD( a , GCD( b, GCD( c , d ) ) );

subarrays are ::

gcd

{ 1 } -- 1

{ 3 } -- 3

{ 6 } -- 6

{ 1 , 3 } -- 1

{ 3 , 6 } -- 3

{ 1 , 3 , 6 } -- 1

so after arranging the increasing order of gcd's

1 . 1

2 . 1

3 . 1

4 . 3

5 . 3

6 . 6

Author:[6★abhi\_1595](https://www.codechef.com/users/abhi_1595)

Tags:[abhi\_1595](https://www.codechef.com/tags/problems/abhi_1595)

Date Added:23-05-2017

Time Limit:3 secs

Source Limit:50000 Bytes

Languages:ADA, ASM, BASH, BF, C, C99 strict, CAML, CLPS, CPP 4.3.2, CPP 4.9.2, CPP14, D, ERL, FORT, FS, HASK, ICK, ICON, JAVA, LISP clisp, LISP sbcl, LUA, NEM, NICE, NODEJS, PAS fpc, PAS gpc, PERL, PERL6, PHP, PIKE, PRLG, PYPY, PYTH, PYTH 3.4, RUBY, SCALA, SCM chicken, SCM guile, SCM qobi, ST, TCL, TEXT, WSPC

<https://www.codechef.com/LOCMAY17/problems/ARGCD>

*------------MI SOLUCION 20 PUNTOS-------------*

*/\**

* *\* To change this template, choose Tools | Templates*
* *\* and open the template in the editor.*
* *\*/*
* *//package javaapplication216;*
* */\*\**
* *\**
* *\* @author Administrador*
* *\*/*
* import java.io.\*;
* import java.math.BigInteger;
* import java.util.ArrayList;
* import java.util.Collections;
* **public** **class** Main{
* */\*\**
* *\* @param args the command line arguments*
* *\*/*
* */\**
* *int gcd(int a, int b)*
* *{*
* *if (a == 0)*
* *return b;*
* *return gcd(b%a, a);*
* *}\*/*
* */\**
* *static BigInteger gcd(BigInteger a, BigInteger b)*
* *{*
* *if (a.compareTo(BigInteger.valueOf( 0))==0)*
* *return b;*
* *return gcd(b.mod(a), a);*
* *}\*/*
* **public** static void main(**[String](http://www.google.com/search?q=allinurl%3AString+java.sun.com&bntl=1)**[] args) **throws** [**IOException**](http://www.google.com/search?q=allinurl%3AIOException+java.sun.com&bntl=1) {
* *// TODO code application logic here*
* [**BufferedReader**](http://www.google.com/search?q=allinurl%3ABufferedReader+java.sun.com&bntl=1) br = **new** java.io.BufferedReader (**new** java.io.InputStreamReader ([**System**](http://www.google.com/search?q=allinurl%3ASystem+java.sun.com&bntl=1).in));
* */\**
* *BigInteger a = BigInteger.valueOf(48);*
* *BigInteger b = BigInteger.valueOf(60);*
* *System.out.println(gcd(a,b));\*/*
* [**String**](http://www.google.com/search?q=allinurl%3AString+java.sun.com&bntl=1)[] input = br.readLine().split(" ");
* int n = [**Integer**](http://www.google.com/search?q=allinurl%3AInteger+java.sun.com&bntl=1).parseInt(input[0]);
* int q = [**Integer**](http://www.google.com/search?q=allinurl%3AInteger+java.sun.com&bntl=1).parseInt(input[1]);
* [**String**](http://www.google.com/search?q=allinurl%3AString+java.sun.com&bntl=1)[] numbers = br.readLine().split(" ");
* [**BigInteger**](http://www.google.com/search?q=allinurl%3ABigInteger+java.sun.com&bntl=1)[] arr\_big = **new** [**BigInteger**](http://www.google.com/search?q=allinurl%3ABigInteger+java.sun.com&bntl=1)[n];
* for(int i =0; i<n; i++) {
* arr\_big[i] = **new** [**BigInteger**](http://www.google.com/search?q=allinurl%3ABigInteger+java.sun.com&bntl=1)(numbers[i]);
* }
* ArrayList<BigInteger> lista = **new** ArrayList<BigInteger>();
* for(int i =0; i<arr\_big.length; i++) {
* [**BigInteger**](http://www.google.com/search?q=allinurl%3ABigInteger+java.sun.com&bntl=1) GCD = arr\_big[i];
* *//lista.add(GCD);*
* for(int j = i; j<arr\_big.length; j++) {
* GCD = GCD.gcd(arr\_big[j]);
* *//System.out.println(GCD);*
* lista.add(GCD);
* }
* }
* [**Collections**](http://www.google.com/search?q=allinurl%3ACollections+java.sun.com&bntl=1).sort(lista);
* */\**
* *for(int i =0; i<lista.size(); i++) {*
* *System.out.println(lista.get(i) + " ");*
* *}\*/*
* while(q-- > 0) {
* int query = [**Integer**](http://www.google.com/search?q=allinurl%3AInteger+java.sun.com&bntl=1).parseInt(br.readLine());
* [**System**](http://www.google.com/search?q=allinurl%3ASystem+java.sun.com&bntl=1).out.println(lista.get(query-1) );
* }


* }
* }

-------------SOLUCION POR prakuj  100 PUNTOS------------

* *//package may;*
* import java.util.ArrayList;
* import java.util.Collections;
* import java.util.Scanner;
* */\*\**
* *\* <h3>Descripton</h3>*
* *\* <p>You are given an array A of size N ,*
* *\* what you have to do , is to list out all the subarrays of the array ,*
* *\* take the gcd's of each subarrays and push back the result into some container(1 indexed)*
* *\* . Now Q queries will be given to you , in each query you will be given a number X ,*
* *\* you have to answer the Xth smallest number from the container.*
* *\* @author pankaj*
* *\* @since 27/05/2017*
* *\*/*
* **class** SimpleOne {
* static long ONE=1l,ZERO=0l;
* **public** static void main(**[String](http://www.google.com/search?q=allinurl%3AString+java.sun.com&bntl=1)**[] args) {
* Scanner sc=**new** Scanner([**System**](http://www.google.com/search?q=allinurl%3ASystem+java.sun.com&bntl=1).in);
* int i,j,n,q,x;
* long hcf;
* n=sc.nextInt();
* q=sc.nextInt();
* long arr[]=**new** long[n];
* for(i=0;i<n;i++){
* arr[i]=sc.nextLong();
* }
* ArrayList<Long> al=**new** ArrayList<>();
* int count=0;
* boolean flag;
* for(i=0;i<n;i++){
* hcf=arr[i];
* flag=**true**;
* for(j=i;j<n;j++){
* if(flag){
* hcf=gcd(hcf,arr[j]);
* if(hcf==1){
* flag=**false**;
* }
* al.add(hcf);
* }
* else
* count++;
* }
* }
* [**Collections**](http://www.google.com/search?q=allinurl%3ACollections+java.sun.com&bntl=1).sort(al);
* *//System.out.println(al.toString());*
* while(q-->0){
* x=sc.nextInt();
* if(x<=count)
* [**System**](http://www.google.com/search?q=allinurl%3ASystem+java.sun.com&bntl=1).out.println(1);
* else
* [**System**](http://www.google.com/search?q=allinurl%3ASystem+java.sun.com&bntl=1).out.println(al.get(x-count-1));
* }
* sc.close();
* }
* **private** static long gcd(long a,long b){
* if(b==ZERO) **return** a;
* if(b==ONE) **return** ONE;
* **return** gcd(b,a%b);
* }
* }

---------------SOLUCION POR  shpalasara 100 PUNTOS----------------------

* import java.io.IOException;
* import java.io.PrintWriter;
* import java.util.ArrayList;
* import java.util.Collections;
* import java.util.InputMismatchException;
* **public** **class** Main {
* **public** static void main(**[String](http://www.google.com/search?q=allinurl%3AString+java.sun.com&bntl=1)**[] args){
* FasterScanner sc = **new** FasterScanner();
* [**PrintWriter**](http://www.google.com/search?q=allinurl%3APrintWriter+java.sun.com&bntl=1) out = **new** [**PrintWriter**](http://www.google.com/search?q=allinurl%3APrintWriter+java.sun.com&bntl=1)(**[System](http://www.google.com/search?q=allinurl%3ASystem+java.sun.com&bntl=1)**.out);
* int n = sc.nextInt();
* int q = sc.nextInt();
* long[] data = **new** long[n];
* for(int i=0;i<n;i++)
* data[i] = sc.nextLong();
* ArrayList<Long> list = **new** ArrayList<Long>();
* long g;
* for(int i=0;i<n;i++)
* {
* g = data[i];
* list.add(g);
* for(int j=i+1;j<n;j++)
* {
* g = gcd(g,data[j]);
* if(g==1)
* **break**;
* list.add(g);
* }
* }
* [**Collections**](http://www.google.com/search?q=allinurl%3ACollections+java.sun.com&bntl=1).sort(list);
* int x;
* int size=(n\*(n+1))>>1;
* int one\_gap = size-list.size();
* while(q-->0)
* {
* x = sc.nextInt();
* if(x<=one\_gap)
* out.println(1);
* else
* out.println(list.get(x-one\_gap-1));
* }
* out.close();
* }
* **public** static long gcd(long a,long b){
* a=**[Math](http://www.google.com/search?q=allinurl%3AMath+java.sun.com&bntl=1)**.abs(a);
* b=**[Math](http://www.google.com/search?q=allinurl%3AMath+java.sun.com&bntl=1)**.abs(b);
* long temp;
* while(b!=0)
* {
* temp = a%b;
* a = b;
* b = temp;
* }
* **return** a;
* }
* static **class** FasterScanner {
* **private** byte[] buf = **new** byte[8192];
* **private** int curChar;
* **private** int numChars;
* **public** int read() {
* if (numChars == -1) {
* **throw** **new** InputMismatchException();
* }
* if (curChar >= numChars) {
* curChar = 0;
* **try** {
* numChars = [**System**](http://www.google.com/search?q=allinurl%3ASystem+java.sun.com&bntl=1).in.read(buf);
* } **catch** (**[IOException](http://www.google.com/search?q=allinurl%3AIOException+java.sun.com&bntl=1)** e) {
* **throw** **new** InputMismatchException();
* }
* if (numChars <= 0) {
* **return** -1;
* }
* }
* **return** buf[curChar++];
* }
* **public** [**String**](http://www.google.com/search?q=allinurl%3AString+java.sun.com&bntl=1) nextLine() {
* int c = read();
* while (isSpaceChar(c)) {
* c = read();
* }
* StringBuilder res = **new** StringBuilder();
* do {
* res.appendCodePoint(c);
* c = read();
* } while (!isEndOfLine(c));
* **return** res.toString();
* }
* **public** [**String**](http://www.google.com/search?q=allinurl%3AString+java.sun.com&bntl=1) nextString() {
* int c = read();
* while (isSpaceChar(c)) {
* c = read();
* }
* StringBuilder res = **new** StringBuilder();
* do {
* res.appendCodePoint(c);
* c = read();
* } while (!isSpaceChar(c));
* **return** res.toString();
* }
* **public** long nextLong() {
* int c = read();
* while (isSpaceChar(c)) {
* c = read();
* }
* int sgn = 1;
* if (c == '-') {
* sgn = -1;
* c = read();
* }
* long res = 0;
* do {
* if (c < '0' || c > '9') {
* **throw** **new** InputMismatchException();
* }
* res \*= 10;
* res += c - '0';
* c = read();
* } while (!isSpaceChar(c));
* **return** res \* sgn;
* }
* **public** int nextInt() {
* int c = read();
* while (isSpaceChar(c)) {
* c = read();
* }
* int sgn = 1;
* if (c == '-') {
* sgn = -1;
* c = read();
* }
* int res = 0;
* do {
* if (c < '0' || c > '9') {
* **throw** **new** InputMismatchException();
* }
* res \*= 10;
* res += c - '0';
* c = read();
* } while (!isSpaceChar(c));
* **return** res \* sgn;
* }
* **public** int[] nextIntArray(int n) {
* int[] arr = **new** int[n];
* for (int i = 0; i < n; i++) {
* arr[i] = nextInt();
* }
* **return** arr;
* }
* **public** long[] nextLongArray(int n) {
* long[] arr = **new** long[n];
* for (int i = 0; i < n; i++) {
* arr[i] = nextLong();
* }
* **return** arr;
* }
* **private** boolean isSpaceChar(int c) {
* **return** c == ' ' || c == '**\n**' || c == '**\r**' || c == '**\t**' || c == -1;
* }
* **private** boolean isEndOfLine(int c) {
* **return** c == '**\n**' || c == '**\r**' || c == -1;
* }
* }
* }