

题目描述

疫情过后，希望小学终于又重新开学了，三年二班开学第一天的任务是将后面的黑板报重新制作。

黑板上已经写上了N个正整数，同学们需要给这每个数分别上一种颜色。

为了让黑板报既美观又有学习意义，老师要求同种颜色的所有数都可以被这种颜色中最小的那个数整除。

现在请你帮帮小朋友们，算算至少需要多少种颜色才能给这N个数进行上色。

输入描述

第一行有一个正整数N，其中，

第二行有N个int型数(保证输入数据在[1,100]范围内)，表示黑板上各个正整数的值。

输出描述

输出只有一个整数，为最少需要的颜色种数。

用例

输入	3 2 4 6
输出	1
说明	所有数都能被2整除
输入	4 2 3 4 9
输出	2
说明	2与4除一种颜色，4能被2整除；3与9除另一种颜色，9能被3整除。不能4个数涂同一种颜色，因为3与9不能被2整除。所以最少的颜色是两种。

题目解析

简单的逻辑题，题目要求：“同种颜色的所有数都可以被这种颜色中最小的那个数整除”。

因此我们可以直接将输入数列进行升序排序，则数列从左到右，元素依次增大，我们每次取最左边的数arr[i]，然后遍历它后面的所有数arr[j]去验证，若可以整除，则为一种颜色，若不可以整除，则为不同颜色。

本题难点主要在于，如何标记一个元素已经涂色了，我这里直接定义了一个长度和输入数列arr相同的数据color，color所有元素默认未初始化，一旦arr[j]可以整除arr[i]，则color[j] = true。

Java算法源码

```
1 import java.util.Arrays;
2 import java.util.Scanner;
3
4 public class Main {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         int n = Integer.parseInt(sc.nextLine());
9
10        int[] arr = Arrays.stream(sc.nextLine().split(" ")).mapToInt(Integer::parseInt).toArray();
11
12        System.out.println(getResult(n, arr));
13    }
14
15    public static int getResult(int n, int[] arr) {
16        Arrays.sort(arr);
17
18        if (arr[0] == 1) {
19            return 1;
20        }
21
22        boolean[] color = new boolean[n];
23        int count = 0;
24
25        for (int i = 0; i < n; i++) {
26            if (color[i]) continue;
27
28            color[i] = true;
29            for (int j = i + 1; j < n; j++) {
30                if ((color[j] && arr[j] % arr[i] == 0) {
31                    color[j] = true;
32                }
33            }
34
35            count++;
36        }
37
38        return count;
39    }
40 }
```

JS算法源码

```
1 // JavaScript Node ACM模式 控制台输入读取 参考
2 const readline = require("readline");
3
4 const rl = readline.createInterface({
5     input: process.stdin,
6     output: process.stdout,
7 });
8
9 const lines = [];
10 rl.on("line", (line) => {
11     lines.push(line);
12
13     if (lines.length === 2) {
14         let n = parseInt(lines[0]);
15         let arr = lines[1].split(" ").slice(0, n);
16
17         console.log(getMinColorCount(arr));
18
19         lines.length = 0;
20     }
21 });
22
23 function getMinColorCount(arr) {
24     arr.sort((a, b) => a - b);
25
26     if (arr[0] === 1) {
27         return 1;
28     }
29
30     let color = new Array(arr.length);
31     let count = 0;
32
33     for (let i = 0; i < arr.length; i++) {
34         if (color[i]) continue;
35         color[i] = true;
36         for (let j = i + 1; j < arr.length; j++) {
37             if ((color[j] && arr[j] % arr[i] === 0) {
38                 color[j] = true;
39             }
40         }
41         count++;
42     }
43
44     return count;
45 }
```

Python算法源码

```
1 # 输入数据
2 n = int(input())
3 arr = list(map(int, input().split()))
4
5
6 # 算法入口
7 def getResult():
8     arr.sort()
9
10    if arr[0] == 1:
11        return 1
12
13    color = [False]*n
14    count = 0
15
16    for i in range(n):
17        if color[i]:
18            continue
19
20        color[i] = True
21        for j in range(i+1, n):
22            if not color[j] and arr[j] % arr[i] == 0:
23                color[j] = True
24
25        count += 1
26
27    return count
28
29 # 调用算法
30 print(getResult())
```