

Assignment #5: 排序和超时

Updated 1343 GMT+8 Oct 10, 2023

2023 fall, Compiled by Xinjie Song, Phy

说明:

- 1) 请把每个题目解题思路 (可选), 源码Python, 或者C++/C (已经在Codeforces/Openjudge上AC), 截图 (包含Accepted, 学号), 填写到下面作业模版中 (推荐使用 typora <https://typoraio.cn>, 或者用word)。AC 或者没有AC, 都请标上每个题目大致花费时间。
- 3) 提交时候先提交pdf文件, 再把md或者doc文件上传到右侧“作业评论”。Canvas需要有同学清晰头像、提交文件有pdf、作业评论有md或者doc。
- 4) 如果不能在截止前提交作业, 请写明原因。

编程环境

操作系统: Windows 11 22H2

Python编程环境: PyCharm 2023.2 (Community Edition)

C/C++编程环境: g++ (x86_64-win32-seh-rev0, Built by MinGW-W64 project) 8.1.0

1. 必做题目

69A. Young Physicist

implementation/math, 1000, <https://codeforces.com/problemset/problem/69/A>

A guy named Vasya attends the final grade of a high school. One day Vasya decided to watch a match of his favorite hockey team. And, as the boy loves hockey very much, even more than physics, he forgot to do the homework. Specifically, he forgot to complete his physics tasks. Next day the teacher got very angry at Vasya and decided to teach him a lesson. He gave the lazy student a seemingly easy task: You are given an idle body in space and the forces that affect it. The body can be considered as a material point with coordinates $(0; 0; 0)$. Vasya had only to answer whether it is in equilibrium. "Piece of cake" — thought Vasya, we need only to check if the sum of all vectors is equal to 0. So, Vasya began to solve the problem. But later it turned out that there can be lots and lots of these forces, and Vasya can not cope without your help. Help him. Write a program that determines whether a body is idle or is moving by the given vectors of forces.

Input

The first line contains a positive integer n ($1 \leq n \leq 100$), then follow n lines containing three integers each: the x -coordinate, the y -coordinate and the z -coordinate of the force vector, applied to the body ($-100 \leq x, y, z \leq 100$).

Output

Print the word "YES" if the body is in equilibrium, or the word "NO" if it is not.

Examples

input

```
3
4 1 7
-2 4 -1
1 -5 -3
```

output

NO

input

```
3
3 -1 7
-5 2 -4
2 -1 -3
```

output

YES

【宋昕杰，物理学院，2023年秋】

思路：读取同时求和，减少存储量和循环量

代码

```
x, y, z = 0, 0, 0

n = int(input())
for i in range(0, n):
    datas = [int(i) for i in input().split()]
    x += datas[0]
    y += datas[1]
    z += datas[2]

if x == y == z == 0:
    print('YES')
else:
    print('NO')
```

代码运行截图

By xjsongphy, contest: Codeforces Beta Round 63 (Div. 2), problem: (A) Young Physicist, **Accepted**, <#>, [Copy](#)

```

x, y, z = 0, 0, 0
n = int(input())
for i in range(0, n):
    datas = [int(i) for i in input().split()]
    x += datas[0]
    y += datas[1]
    z += datas[2]
if x == y == z == 0:
    print('YES')
else:
    print('NO')

```

→Judgement Protocol

Test: #1, time: 30 ms., memory: 0 KB, exit code: 0, checker exit code: 0, verdict: OK

Input

```

3
4 1 7
-2 4 -1
1 -5 -3

```

Output

```

NO

```

Answer

```

NO

```

Checker Log

```

ok NO

```

Test: #2, time: 30 ms., memory: 0 KB, exit code: 0, checker exit code: 0, verdict: OK

Input

```

3
3 -1 7
-5 2 -4
2 -1 -3

```

Output

```

YES

```

Answer

```

YES

```

96A. Football

implementation/strings, 900, <http://codeforces.com/problemset/problem/96/A>

Petya loves football very much. One day, as he was watching a football match, he was writing the players' current positions on a piece of paper. To simplify the situation he depicted it as a string consisting of zeroes and ones. A zero corresponds to players of one team; a one corresponds to players of another team. If there are at least 7 players of some team standing one after another, then the situation is considered dangerous. For example, the situation 0010011011111101 is dangerous and 11110111011101 is not. You are given the current situation. Determine whether it is dangerous or not.

Input

The first input line contains a non-empty string consisting of characters "0" and "1", which represents players. The length of the string does not exceed 100 characters. There's at least one player from each team present on the field.

Output

Print "YES" if the situation is dangerous. Otherwise, print "NO".

Examples

input

```
001001
```

output

```
NO
```

input

1000000001

output

YES

【宋昕杰，物理学院，2023年秋】

思路：利用Python提供的in关键字可以简洁地完成任务

代码

```
str_input = input()
if '1111111' in str_input or '0000000' in str_input:
    print('YES')
else:
    print('NO')
```

代码运行截图

The screenshot shows a web browser displaying the Codeforces submission page for problem 96A. The page includes the Codeforces logo, navigation links, and a table of submission details. The source code is visible, and the test results show a successful verdict (OK) for the first test case.

Submission Details:

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged	
221254710	Practice: xjsongphy	96A - 10	Python 3	Accepted	92 ms	0 KB	2023-08-31 14:15:18	2023-08-31 14:15:18	★ Compare

Source Code:

```
str_input = input()
if '1111111' in str_input or '0000000' in str_input:
    print('YES')
else:
    print('NO')
```

Test Results:

1
Time: 60 ms, memory: 0 KB
Verdict: OK
Input: 001001
Participant's output: NO
Jury's answer: NO
Checker comment: ok NO

2
Time: 60 ms, memory: 0 KB
Verdict: OK

270A. Fancy Fence

geometry/implementation/math, 1100, x23265, <https://codeforces.com/problemset/problem/270/A>

Emuskald needs a fence around his farm, but he is too lazy to build it himself. So he purchased a fence-building robot.

He wants the fence to be a regular polygon. The robot builds the fence along a single path, but it can only make fence corners at a single angle α .

Will the robot be able to build the fence Emuskald wants? In other words, is there a regular polygon which angles are equal to α ?

Input

The first line of input contains an integer t ($0 < t < 180$) — the number of tests. Each of the following t lines contains a single integer α ($0 < \alpha < 180$) — the angle the robot can make corners at measured in degrees.

Output

For each test, output on a single line "YES" (without quotes), if the robot can build a fence Emuskald wants, and "NO" (without quotes), if it is impossible.

Examples

input

```
3
30
60
90
```

output

```
NO
YES
YES
```

Note

In the first test case, it is impossible to build the fence, since there is no regular polygon with angle 30° .

In the second test case, the fence is a regular triangle, and in the last test case — a square.

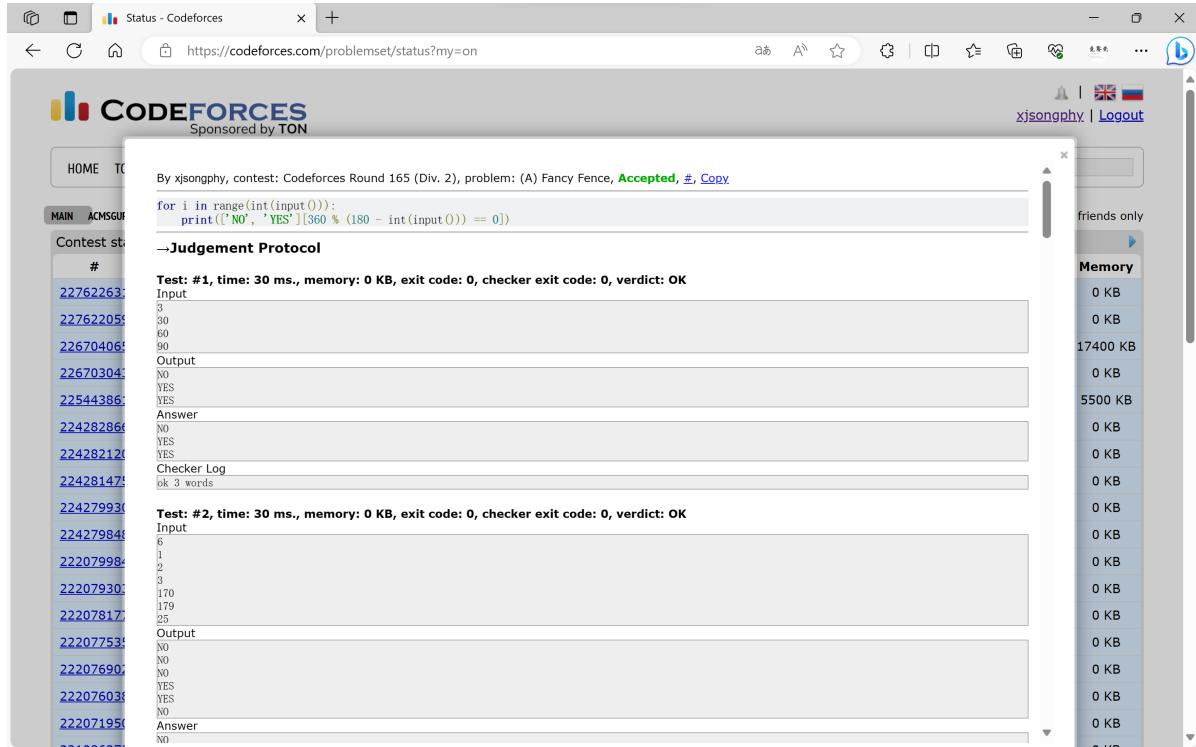
【宋昕杰，物理学院，2023年秋】

思路：找出数学公式并求解

代码

```
for i in range(int(input())):
    print(['NO', 'YES'][360 % (180 - int(input())) == 0])
```

代码运行截图



160A. Twins

greedy, sortings, 900, <https://codeforces.com/problemset/problem/160/A>

Imagine that you have a twin brother or sister. Having another person that looks exactly like you seems very unusual. It's hard to say if having something of an alter ego is good or bad. And if you do have a twin, then you very well know what it's like.

Now let's imagine a typical morning in your family. You haven't woken up yet, and Mom is already going to work. She has been so hasty that she has nearly forgotten to leave the two of her darling children some money to buy lunches in the school cafeteria. She fished in the purse and found some number of coins, or to be exact, n coins of arbitrary values a_1, a_2, \dots, a_n . But as Mom was running out of time, she didn't split the coins for you two. So she scribbled a note asking you to split the money equally.

As you woke up, you found Mom's coins and read her note. "But why split the money equally?" — you thought. After all, your twin is sleeping and he won't know anything. So you decided to act like that: pick for yourself some subset of coins so that the sum of values of your coins is **strictly larger** than the sum of values of the remaining coins that your twin will have. However, you correctly thought that if you take too many coins, the twin will suspect the deception. So, you've decided to stick to the following strategy to avoid suspicions: you take the **minimum number of coins**, whose sum of values is strictly more than the sum of values of the remaining coins. On this basis, determine what **minimum** number of coins you need to take to divide them in the described manner.

Input

The first line contains integer n ($1 \leq n \leq 100$) — the number of coins. The second line contains a sequence of n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 100$) — the coins' values. All numbers are separated with spaces.

Output

In the single line print the single number — the minimum needed number of coins.

Examples

input

```
2
3 3
```

output

```
2
```

input

```
3
2 1 2
```

output

```
2
```

Note

In the first sample you will have to take 2 coins (you and your twin have sums equal to 6, 0 correspondingly). If you take 1 coin, you get sums 3, 3. If you take 0 coins, you get sums 0, 6. Those variants do not satisfy you as your sum should be strictly more than your twin's sum.

In the second sample one coin isn't enough for us, too. You can pick coins with values 1, 2 or 2, 2. In any case, the minimum number of coins equals 2.

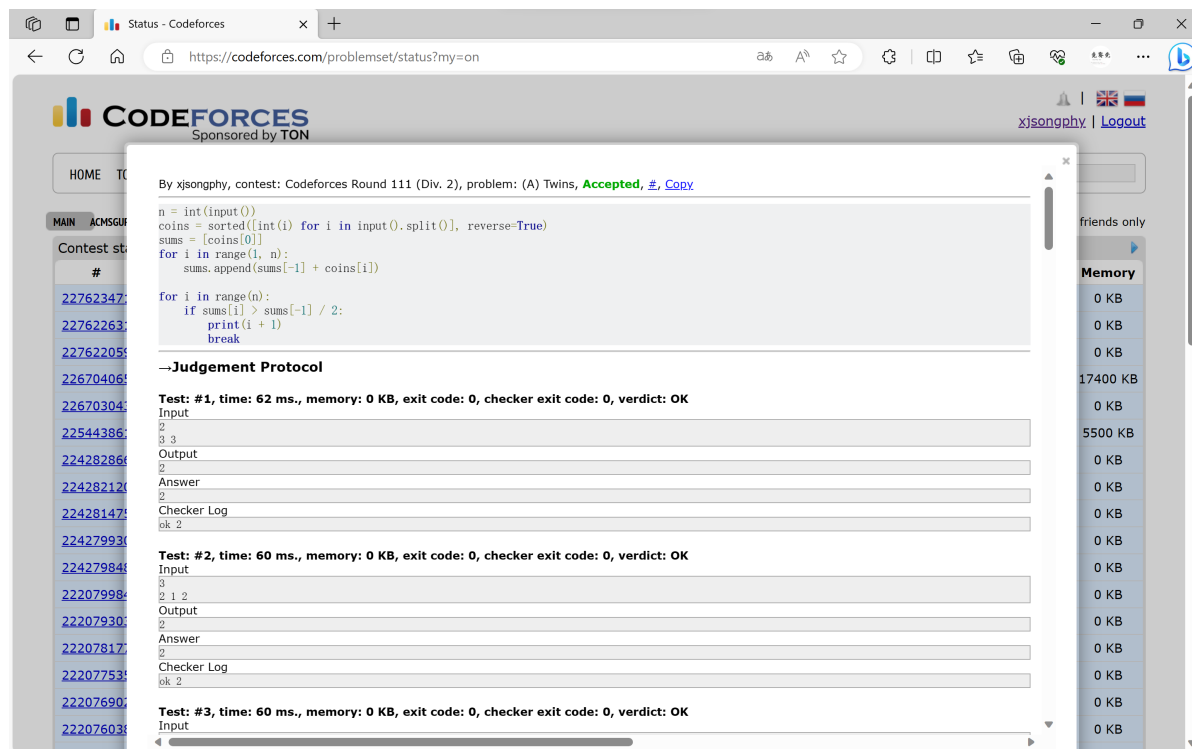
【宋昕杰，物理学院，2023年秋】

思路：简单贪心策略，每次选取最大的硬币，即依次由大至小求和，大于总数的一半时停止并输出

代码

```
n = int(input())
coins = sorted([int(i) for i in input().split()], reverse=True)
sums = [coins[0]]
for i in range(1, n):
    sums.append(sums[-1] + coins[i])
for i in range(n):
    if sums[i] > sums[-1] / 2:
        print(i + 1)
        break
```

代码运行截图



2. 选做题目

12559: 最大最小整数 v0.3

greedy/strings/sortings, <http://cs101.openjudge.cn/practice/12559>

假设有n个正整数，将它们连成一片，将会组成一个新的大整数。现要求出，能组成的最大最小整数。

比如，有4个正整数，23，9，182，79，连成的最大整数是97923182，最小的整数是18223799。

输入

第一行包含一个整数n， $1 \leq n \leq 1000$ 。

第二行包含n个正整数，相邻正整数间以空格隔开。

输出

输出为一行，为这n个正整数能组成的最大的多位整数和最小的多位整数，中间用空格隔开。

样例输入

```
Sample1 in:
4
23 9 182 79

Sample1 out:
97923182 18223799
```

样例输出

```
Sample2 in:
2
11 113

Sample2 out:
11311 11113
```

【宋昕杰，物理学院，2023年秋】

思路：一开始写了个插入排序，运行时间6000+ms，后来看历年同学的解法改成了冒泡排序，减小了时间复杂度

代码

```
n = int(input())
nums = input().split()
for i in range(n - 1):
    for j in range(i+1, n):
        if nums[i] + nums[j] < nums[j] + nums[i]:
            nums[i], nums[j] = nums[j], nums[i]
print(''.join(nums), end=' ')
nums.reverse()
print(''.join(nums))
```

代码运行截图

OpenJudge - 提交状态 x +

不安全 | cs101.openjudge.cn/practice/solution/41584111/

OpenJudge 题目ID, 标题, 描述 23n2300011524 信箱 账号

CS101 / 题库

题目 排名 状态 提问

#41584111提交状态 查看 提交 统计 提问

状态: Accepted

源代码

```
n = int(input())
nums = input().split()
for i in range(n - 1):
    for j in range(i + 1, n):
        if nums[i] + nums[j] < nums[j] + nums[i]:
            nums[i], nums[j] = nums[j], nums[i]
print(''.join(nums), end=' ')
nums.reverse()
print(''.join(nums))
```

基本信息

#: 41584111
题目: 12559
提交人: 23n2300011524
内存: 3660kB
时间: 176ms
语言: Python3
提交时间: 2023-10-11 17:19:24

©2002-2022 POJ 京ICP备20010980号-1 English 帮助 关于

230B. T-primes

binary search/implementation/math/number theory, 1300, <http://codeforces.com/problemset/problem/230/B>

We know that prime numbers are positive integers that have exactly two distinct positive divisors. Similarly, we'll call a positive integer t T-prime, if t has exactly three distinct positive divisors.

You are given an array of n positive integers. For each of them determine whether it is T-prime or not.

Input

The first line contains a single positive integer, n ($1 \leq n \leq 10^5$), showing how many numbers are in the array. The next line contains n space-separated integers x_i ($1 \leq x_i \leq 10^{12}$).

Please, do not use the %lld specifier to read or write 64-bit integers in C++. It is advised to use the cin, cout streams or the %I64d specifier.

Output

Print n lines: the i -th line should contain "YES" (without the quotes), if number x_i is T-prime, and "NO" (without the quotes), if it isn't.

Examples

input

```
3
4 5 6
```

output

YES

NO

NO

Note

The given test has three numbers. The first number 4 has exactly three divisors — 1, 2 and 4, thus the answer for this number is "YES". The second number 5 has two divisors (1 and 5), and the third number 6 has four divisors (1, 2, 3, 6), hence the answer for them is "NO".

【宋昕杰，物理学院，2023年秋】

思路：欧拉筛求出质数，质数的平方是所求T-prime，这是因为T-prime除以多出来的因数只能仍是这个因数，即T-prime是完全平方数，又要求这个的因数没有其他因数，因此这个因数必须是质数。使用Python实现的时候，由于Python执行速度较慢，因此选择字典作为数据结构方便查找和删除（即当值取某一个特定值时，键被视为删除），再考虑到sqrt函数可能会较慢（这是因为sqrt返回值有多位小数），因此起初我手动寻找最大数的平方根作为筛质数的限制，但提交后发现测试数据很不友好，因此直接取 x_i 的上线 10^{12} 的平方根作为限制，这样虽然会导致简短的测试数据用时较长，但不会让复杂的测试数据超时；接下来，找到小于等于 10^6 的所有质数，开始使用埃氏筛发现超时，便学习了欧拉筛后使用欧拉筛筛出质数；这里，找到某一个质数后直接在输入数据中查找是否有其平方存在，若存在则记录，这样省去了生成质数表后再次查找的时间；最后，按输入顺序输出即可（本来想将输入存为字典，但测试数据中会有 x_i 重复的情况出现，因此只能先存为列表，在另存为字典，最后根据列表的数据和字典中存储的对应数据的结果输出最终的结果）。

注：最终CF显示用时为1870ms

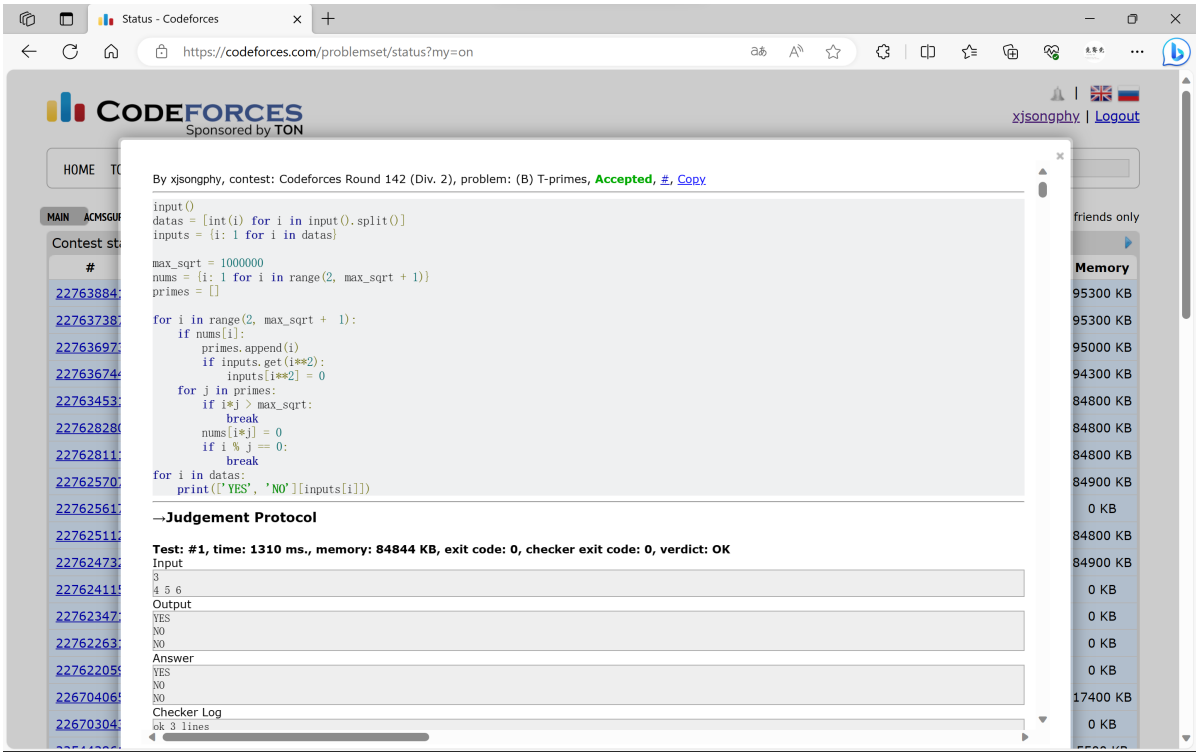
代码

```
input()
datas = [int(i) for i in input().split()]
inputs = {i: 1 for i in datas}

max_sqrt = 1000000
nums = {i: 1 for i in range(2, max_sqrt + 1)}
primes = []

for i in range(2, max_sqrt + 1):          #欧拉筛
    if nums[i]:
        primes.append(i)
        if inputs.get(i**2):
            inputs[i**2] = 0
        for j in primes:
            if i*j > max_sqrt:
                break
            nums[i*j] = 0
            if i % j == 0:
                break
for i in datas:
    print(['YES', 'NO'][inputs[i]])
```

代码运行截图



3. 学习总结和收获

第一次了解了埃氏筛和欧拉筛（其实以前会埃氏筛，只是不知道叫什么名字），优化CF230B题目的程序时综合运用了各种省时间的方法，也算是一种巩固了。

这段时间每天依然是先做新加入的题目，但做完以后分配的时间基本用尽甚至超出，而且没有深入学习贪心和动态规划算法就做题可能效果不是太好，因此下一段时间准备先学《算法基础与在线实践》，有时间再做新题。

截至2023年10月13日10:30，OJ完成题目65道，CF完成题目29道。