

Assignment #5: 排序和超时

Updated 1343 GMT+8 Oct 10, 2023

2023 fall, Compiled by Hongfei Yan (请改为同学的姓名、院系)

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

(请改为同学的操作系统、编程环境等)

操作系统: macOS Ventura 13.4.1 (c)

Python编程环境: Spyder IDE 5.2.2, PyCharm 2023.1.4 (Professional Edition)

C/C++编程环境: Mac terminal vi (version 9.0.1424), g++/gcc (Apple clang version 14.0.3, clang-1403.0.22.14.1)

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_i coordinate, the y_i coordinate and the z_i coordinate of the force vector, applied to the body ($-100 \leq x_i, y_i, z_i \leq 100$).

Output

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

Examples

input

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

output

```
1 | NO
```

input

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

output

```
1 | YES
```

【刘思瑞，元培物理方向，2023年秋】

思路：

直接相加用列表储存

代码

```
1 | '''
2 | 刘思瑞 2100017810
3 | '''
4 | n = int(input())
5 | sum = [0,0,0]
6 | for i in range(n):
7 |     li = list(map(int,input().split()))
8 |     for j in range(3):
9 |         sum[j] += li[j]
10 | if sum == [0]*3:
11 |     print('YES')
12 | else:
13 |     print('NO')
```

By meinvader, contest: Codeforces Beta Round 63 (Div. 2), problem: (A) Young Physicist, **Accepted**,

```
'''
刘思瑞 2100017810
'''
n = int(input())
sum = [0,0,0]
for i in range(n):
    li = list(map(int,input().split()))
    for j in range(3):
        sum[j] += li[j]
if sum == [0]*3:
    print('YES')
else:
    print('NO')
```

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

```
1 | 001001
```

output

```
1 | NO
```

input

```
1 | 1000000001
```

output

【刘思瑞，元培物理方向，2023年秋】

思路：

正则表达式

代码

```

1  '''
2  刘思瑞 2100017810
3  '''
4  import re
5
6  s = input()
7  if re.search('1{7}',s) or re.search('0{7}',s):
8      print('YES')
9  else:
10     print('NO')

```

代码运行截图

By meinvader, contest: Codeforces Beta Round 77 (Div. 2 Only), problem: (A) Football, **Accepted**

```

'''
刘思瑞 2100017810
'''
import re

s = input()
if re.search('1{7}',s) or re.search('0{7}',s):
    print('YES')
else:
    print('NO')

```

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 a ($0 < a < 180$) — 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

```
1 3
2 30
3 60
4 90
```

output

```
1 NO
2 YES
3 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.

【刘思瑞，元培物理方向，2023年秋】

思路:

外角和180

代码

```
1 '''
2 刘思瑞 2100017810
3 '''
4 n = int(input())
5 for i in range(n):
6     if 360 % (180 - int(input())) == 0:
7         print('YES')
8     else:
9         print('NO')
```

代码运行截图

```
'''
刘思瑞 2100017810
'''
n = int(input())
for i in range(n):
    if 360 % (180 - int(input())) == 0:
        print('YES')
    else:
        print('NO')
```

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

```
1 2
2 3 3
```

output

```
1 | 2
```

input

```
1 | 3
2 | 2 1 2
```

output

```
1 | 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.

【刘思瑞，元培物理方向，2023年秋】

思路:

贪心，判断是否过半

代码

```
1  '''
2  刘思瑞 2100017810
3  '''
4  n = int(input())
5  coin = list(map(int, input().split()))
6  coin.sort(reverse=True)
7  summ = sum(coin)
8  half = summ//2
9  num = 0
10 sumcoin = 0
11 for i in coin:
12     sumcoin += i
13     num += 1
14     if sumcoin > half:
15         break
16 print(num)
```

代码运行截图

```
'''
刘思瑞 2100017810
'''
n = int(input())
coin = list(map(int, input().split()))
coin.sort(reverse=True)
summ = sum(coin)
half = summ//2
num = 0
sumcoin = 0
for i in coin:
    sumcoin += i
    num += 1
    if sumcoin > half:
        break
print(num)
```

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

```
1 Sample1 in:
2 4
3 23 9 182 79
4
5 Sample1 out:
6 97923182 18223799
```

样例输出


```
1 Sample2 in:
2 2
3 11 113
4
5 Sample2 out:
6 11311 11113
```

【刘思瑞，元培物理方向，2023年秋】

思路:

一开始我想的是先按不同首数字分组再排序，但是这样内存会超，所以还是直接冒泡排序了

需要注意的是这次的冒泡排序需要与后面所有的作比较

代码

```
1 '''
2 刘思瑞 2100017810
3 '''
4 #冒泡排序
5 def compare(s1,s2):
6     if int(s1 + s2) > int(s2 +s1):
7         return True
8     else:
9         return False
10 n = int(input())
11 number = list(input().split())
12 #max number:
13 for i in range(n-1):
14     for j in range(i+1,n):
15         if not compare(number[i],number[j]):
16             temp = number[j]
17             number[j] = number[i]
18             number[i] =temp
19 for i in number:
20     print(i,end='')
21 print(' ',end='')
22 #min number:
23 number.reverse()
24 for i in number:
25     print(i,end='')
```

状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
#冒泡排序
def compare(s1,s2):
    if int(s1 + s2) > int(s2 +s1):
        return True
    else:
        return False
n = int(input())
number = list(input().split())
#max number:
for i in range(n-1):
    for j in range(i+1,n):
        if not compare(number[i],number[j]):
            temp = number[j]
            number[j] = number[i]
            number[i] =temp
for i in number:
    print(i,end='')
print(' ',end='')
#min number:
number.reverse()
for i in number:
    print(i,end='')
```

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

```
1 3
2 4 5 6
```

output

```
1 YES
2 NO
3 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".

【刘思瑞，元培物理方向，2023年秋】

思路:

欧拉筛加上平方筛选

代码

```
1 '''
2 刘思瑞 2100017810
3 '''
4 def euler(n):
5     filter, primers = [False for i in range(n + 1)], []
6     for i in range(2, n + 1):
7         if not filter[i]:
8             primers.append(i)
9             for prime in primers:
10                 if i * prime > n:
11                     break
12                 filter[i * prime] = True
13                 if i % prime == 0:
14                     break
15     return filter
16
17 def search(num):
18     global prime_
19     sq = int(num**(0.5))
20     if int(sq**2) != num or num == 1:
21         return 'NO'
22     if not prime_[sq]:
23         return 'YES'
24     return 'NO'
25
```

```

26 n = int(input())
27 num = list(map(int, input().split()))
28 n = int(max(num)**(0.5))
29 prime_ = euler(n)
30 for i in num:
31     print(search(i))

```

代码运行截图

By meinvader, contest: Codeforces Round 142 (Div. 2), problem: (B) T-primes, **Accepted**,

```

'''
刘思瑞 2100017810
'''
def euler(n):
    filter, primers = [False for i in range(n + 1)], []
    for i in range(2, n + 1):
        if not filter[i]:
            primers.append(i)
            for prime in primers:
                if i * prime > n:
                    break
                filter[i * prime] = True
                if i % prime == 0:
                    break
    return filter

def search(num):
    global prime_
    sq = int(num**(0.5))
    if int(sq**2) != num or num == 1:
        return 'NO'
    if not prime_[sq]:
        return 'YES'
    return 'NO'

n = int(input())
num = list(map(int, input().split()))
n = int(max(num)**(0.5))
prime_ = euler(n)
for i in num:
    print(search(i))

```

Judgment Protocol

3. 学习总结和收获

最后一题的超时实在令人头疼，最后想到了排队里面用的tf数组，应该是主要超时在a in b的计算里面了，改用列表和索引之后python过了