

Assignment #4: 国庆节月考

Updated 1641 GMT+8 Oct 5, 2023

2023 fall, Compiled by 刘思瑞 元培学院

说明:

- 1) 国庆节月考: AC8。摸底题目都在“练习”里面, 按照数字题号能找到, 可以重新提交。作业中提交自己最满意版本的代码和截图。
- 2) 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 22621.2283

Python编程环境: Visual Studio (1.82.2); python 3.11.3

C/C++编程环境: 无

1. 必做题目

02701: 与7无关的数

math, <http://cs101.openjudge.cn/practice/02701>

一个正整数,如果它能被7整除,或者它的十进制表示法中某一位上的数字为7,则称其为与7相关的数.现求所有小于等于 n ($n < 100$)的与7无关的正整数的平方和.

输入

输入为一行,正整数 n ($n < 100$)

输出

输出一行, 包含一个整数, 即小于等于 n 的所有与7无关的正整数的平方和。

样例输入

1 | 21

样例输出

来源

计算概论05

【刘思瑞，元培物理方向，2023年秋】

思路：

直接遍历即可，但是对于更大的数目应该考虑计算所有平方和再减掉和7有关的这样可以节省很多计算量

代码

```
1  '''
2  刘思瑞 2100017810
3  '''
4  n = int(input())
5  sum = 0
6  for i in range(1,n+1):
7      if i % 7 != 0 and not('7' in str(i)):
8          sum += i**2
9  print(sum)
```

代码运行截图

状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
n = int(input())
sum = 0
for i in range(1,n+1):
    if i % 7 != 0 and not('7' in str(i)):
        sum += i**2
print(sum)
```

©2002-2022 POJ 京ICP备20010980号-1

02712: 细菌繁殖

math, <http://cs101.openjudge.cn/practice/02712>

一种细菌的繁殖速度是每天成倍增长。例如：第一天有10个，第二天就变成20个，第三天变成40个，第四天变成80个，……。现在给出第一天的日期和细菌数目，要你写程序求出到某一天的时候，细菌的数目。

输入

第一行有一个整数n，表示测试数据的数目。其后n行每行有5个整数，整数之间用一个空格隔开。第一个数表示第一天的月份，第二个数表示第一天的日期，第三个数表示第一天细菌的数目，第四个数表示要求的那一天的月份，第五个数表示要求的那一天的日期。已知第一天和要求的一天在同一年并且该年不是闰年，要求的一天一定在第一天之后。数据保证要求的一天的细菌数目在长整数（long）范围内。

输出

对于每一组测试数据，输出一行，该行包含一个整数，为要求的一天的细菌数。

样例输入

```
1 2
2 1 1 1 1 2
3 2 28 10 3 2
```

样例输出

```
1 2
2 40
```

来源

2005~2006医学部计算概论期末考试

【刘思瑞，元培物理方向，2023年秋】

思路：

写了一个列表来计算月份，实在是麻烦这种题。。

代码

```
1 '''
2 刘思瑞 2100017810
3 '''
4 def culcu(t):
5     global long
6     m0 , d0 , ini , m , d = map(int,input().split())
7     day = 0
8     if m0 == m:
9         times = d - d0
10    else:
11        for i in range(m0+1,m):
12            day += long[i]
13        times = long[m0] - d0 + day + d
14    return ini * (2 ** times)
15
16 long = [0 , 31,28,31,30,31,30,31,31,30,31,30,31]
17 n = int(input())
18 for i in range(n):
19     print(culcu(1))
```

代码运行截图

状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
def culcu(t):
    global long
    m0 , d0 , ini , m , d = map(int,input().split())
    day = 0
    if m0 == m:
        times = d - d0
    else:
        for i in range(m0+1,m):
            day += long[i]
        times = long[m0] - d0 + day + d
    return ini * (2 ** times)

long = [0 , 31,28,31,30,31,30,31,31,30,31,30,31]
n = int(input())
for i in range(n):
    print(culcu(1))
```

©2002-2022 POJ 京ICP备20010980号-1

02753: 菲波那契数列

math, <http://cs101.openjudge.cn/practice/02753>

菲波那契数列是指这样的数列: 数列的第一个和第二个数都为1, 接下来每个数都等于前面2个数之和。
给出一个正整数a, 要求菲波那契数列中第a个数是多少。

输入

第1行是测试数据的组数n, 后面跟着n行输入。每组测试数据占1行, 包括一个正整数a(1 <= a <= 20)

输出

输出有n行, 每行输出对应一个输入。输出应是一个正整数, 为菲波那契数列中第a个数的大小
样例输入

1	4
2	5
3	2
4	19
5	1

样例输出

```
1 5
2 1
3 4181
4 1
```

【刘思瑞，元培物理方向，2023年秋】

思路：

用三位的列表来储存数列，节省最小的内存，不过一开始oj报了CE，好像是必须在函数前面声明全局变量？很奇怪

代码

```
1 '''
2 刘思瑞 2100017810
3 '''
4 f = [1, 1, 2]
5
6 def forward(t):
7     global f
8     f = f[1:]
9     f.append(f[0]+f[1])
10
11 n = int(input())
12 for i in range(n):
13     a = int(input())
14     f = [1, 1, 2]
15     if a <= 3:
16         print(f[a-1])
17     else:
18         for j in range(a-3):
19             forward(1)
20         print(f[2])
```

状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
f = [1 ,1 ,2]

def forward(t):
    global f
    f = f[1:]
    f.append(f[0]+f[1])

n = int(input())
for i in range(n):
    a = int(input())
    f = [1 ,1 ,2]
    if a <= 3:
        print(f[a-1])
    else:
        for j in range(a-3):
            forward(1)
        print(f[2])
```

02810: 完美立方

bruteforce, <http://cs101.openjudge.cn/practice/02810>

形如 $a^3 = b^3 + c^3 + d^3$ 的等式被称为完美立方等式。例如 $12^3 = 6^3 + 8^3 + 10^3$ 。编写一个程序，对任给的正整数N ($N \leq 100$)，寻找所有的四元组 (a, b, c, d)，使得 $a^3 = b^3 + c^3 + d^3$ ，其中 a,b,c,d 大于1, 小于等于N, 且 $b \leq c \leq d$ 。

输入

一个正整数N ($N \leq 100$)。

输出

每行输出一个完美立方。输出格式为：

Cube = a, Triple = (b,c,d)

其中a,b,c,d所在位置分别用实际求出四元组值代入。

请按照a的值，从小到大依次输出。当两个完美立方等式中a的值相同，则b值小的优先输出、仍相同则c值小的优先输出、再相同则d值小的先输出。

样例输入

1 | 24

样例输出

```
1 Cube = 6, Triple = (3,4,5)
2 Cube = 12, Triple = (6,8,10)
3 Cube = 18, Triple = (2,12,16)
4 Cube = 18, Triple = (9,12,15)
5 Cube = 19, Triple = (3,10,18)
6 Cube = 20, Triple = (7,14,17)
7 Cube = 24, Triple = (12,16,20)
```

【刘思瑞，元培物理方向，2023年秋】

思路：

我直接按大小顺序遍历了，但是感觉处理i, j, k的范围还可以优化

代码

```
1 '''
2 刘思瑞 2100017810
3 '''
4 import math
5 def find(num):
6     result = []
7     for i in range(1,num+1):
8         for j in range(2,i):
9             for k in range(j,i):
10                 rest = i**3 - j**3 - k**3
11                 if rest >= k**3:
12                     for m in range(int(rest**(1/3))+2):
13                         if m**3 == rest:
14                             result.append([i,j,k,m])
15     return result
16
17 n = int(input())
18 re = find(n)
19 for i in re:
20     print('Cube = %d, Triple = (%d,%d,%d)' % (i[0],i[1],i[2],i[3]))
```

代码运行截图

状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
import math
def find(num):
    result = []
    for i in range(1,num+1):
        for j in range(2,i):
            for k in range(j,i):
                rest = i**3 - j**3 - k**3
                if rest >= k**3:
                    for m in range(int(rest**(1/3))+2):
                        if m**3 == rest:
                            result.append([i,j,k,m])
    return result

n = int(input())
re = find(n)
for i in re:
    print('Cube = %d, Triple = (%d,%d,%d)' % (i[0],i[1],i[2],i[3]))
```

==

04138: 质数的和与积

math, <http://cs101.openjudge.cn/practice/04138>

两个质数的和是S，它们的积最大是多少？

输入

一个不大于10000的正整数S，为两个质数的和。

输出

一个整数，为两个质数的最大乘积。数据保证有解。

样例输入

```
1 | 50
```

样例输出

```
1 | 589
```

来源

《奥数典型题举一反三（小学五年级）》(ISBN 978-7-5445-2882-5) 第三章 第二讲 例1

【刘思瑞，元培物理方向，2023年秋】

思路：

肯定是差越小积越大因此从中间遍历，筛选质数还可以优化不过数据量太小没必要

代码

```
1  '''
2  刘思瑞 2100017810
3  '''
4  def detect(num):
5      for i in range(2,num):
6          if num % i == 0:
7              return False
8      return True
9
10 num = int(input())
11 for i in range(num//2,num-1):
12     if detect(i) and detect(num - i):
13         print(i*(num-i))
14         break
```

代码运行截图

状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
def detect(num):
    for i in range(2,num):
        if num % i == 0:
            return False
    return True

num = int(input())
for i in range(num//2,num-1):
    if detect(i) and detect(num - i):
        print(i*(num-i))
        break
```

04146: 数字方格

math, <http://cs101.openjudge.cn/practice/04146>

a1	a2	a3
----	----	----

如上图，有3个方格，每个方格里面都有一个整数 a_1 , a_2 , a_3 。已知 $0 \leq a_1, a_2, a_3 \leq n$ ，而且 $a_1 + a_2$ 是2的倍数， $a_2 + a_3$ 是3的倍数， $a_1 + a_2 + a_3$ 是5的倍数。你的任务是找到一组 a_1 , a_2 , a_3 ，使得 $a_1 + a_2 + a_3$ 最大。

输入

一行，包含一个整数 n ($0 \leq n \leq 100$)。

输出

一个整数，即 $a_1 + a_2 + a_3$ 的最大值。

样例输入

1 | 3

样例输出

1 | 5

【刘思瑞，元培物理方向，2023年秋】

思路：

直接暴力枚举了，肯定有更好的做法hhhhh

代码

```
1  '''
2  刘思瑞 2100017810
3  '''
4  n = int(input())
5  max = 0
6  for i in range(1,n+1):
7      for j in range(1,n+1):
8          for k in range(1,n+1):
9              if (i+j)%2 == 0 and (j+k)%3 == 0 and (i+j+k)%5 == 0 and
10 i+j+k>max:
11                 max = i+j+k
12 print(max)
```

状态: Accepted

源代码

```
'''  
刘思瑞 2100017810  
'''  
n = int(input())  
max = 0  
for i in range(1, n+1):  
    for j in range(1, n+1):  
        for k in range(1, n+1):  
            if (i+j)%2 == 0 and (j+k)%3 == 0 and (i+j+k)%5 == 0 and i+j+k == n:  
                max = i+j+k  
print(max)
```

2. 选做题目

02746: 约瑟夫问题

implementation, <http://cs101.openjudge.cn/practice/02746>

约瑟夫问题：有 n 只猴子，按顺时针方向围成一圈选大王（编号从 1 到 n ），从第 1 号开始报数，一直数到 m ，数到 m 的猴子退出圈外，剩下的猴子再接着从 1 开始报数。就这样，直到圈内只剩下一只猴子时，这个猴子就是猴王，编程求输入 n, m 后，输出最后猴王的编号。

输入

每行是用空格分开的两个整数，第一个是 n ，第二个是 m ($0 < m, n \leq 300$)。最后一行是：

0 0

输出

对于每行输入数据（最后一行除外），输出数据也是一行，即最后猴王的编号

样例输入

1	6 2
2	12 4
3	8 3
4	0 0

样例输出

1	5
2	1
3	7

【刘思瑞，元培物理方向，2023年秋】

思路：

在循环链表和列表之间选择了置标志位QAQ

代码

```
1  '''
2  刘思瑞 2100017810
3  '''
4  def delete(n,m):
5      flag = [True]*n
6      j = 0
7      times = 0
8      sum = 0
9      while True:
10         if flag[j] == True:
11             sum +=1
12             if sum == m:
13                 flag[j] = False
14                 times += 1
15                 sum = 0
16             if times == n-1:
17                 break
18             if j == n-1:
19                 j = 0
20             else:
21                 j+=1
22         return flag.index(True) + 1
23
24
25 while True:
26     n,m = map(int,input().split())
27     if [n,m] == [0,0]:
28         break
29     print(delete(n,m))
```

状态: Accepted

源代码

```
'''
刘思瑞 2100017810
'''
def delete(n,m):
    flag = [True]*n
    j = 0
    times = 0
    sum = 0
    while True:
        if flag[j] == True:
            sum +=1
            if sum == m:
                flag[j] = False
                times += 1
                sum = 0
            if times == n-1:
                break
        if j == n-1:
            j = 0
        else:
            j+=1
    return flag.index(True) + 1

while True:
    n,m = map(int,input().split())
    if [n,m] == [0,0]:
        break
    print(delete(n,m))
```

CF1364A: A. XXXXX

brute force/data structures/number theory/two pointers, 1200, <https://codeforces.com/problemset/problem/1364/A>

Ehab loves number theory, but for some reason he hates the number x . Given an array a , find the length of its longest subarray such that the sum of its elements **isn't** divisible by x , or determine that such subarray doesn't exist.

An array a is a subarray of an array b if a can be obtained from b by deletion of several (possibly, zero or all) elements from the beginning and several (possibly, zero or all) elements from the end.

Input

The first line contains an integer t ($1 \leq t \leq 5$) — the number of test cases you need to solve. The description of the test cases follows.

The first line of each test case contains 2 integers n and x ($1 \leq n \leq 10^5$, $1 \leq x \leq 10^4$) — the number of elements in the array a and the number that Ehab hates.

The second line contains n space-separated integers a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^4$) — the elements of the array a .

Output

For each testcase, print the length of the longest subarray whose sum isn't divisible by x . If there's no such subarray, print -1 .

Example

input

```
1 3
2 3 3
3 1 2 3
4 3 4
5 1 2 3
6 2 2
7 0 6
```

output

```
1 2
2 3
3 -1
```

Note

In the first test case, the subarray $[2,3]$ has sum of elements 5, which isn't divisible by 3.

In the second test case, the sum of elements of the whole array is 6, which isn't divisible by 4.

In the third test case, all subarrays have an even sum, so the answer is -1 .

【刘思瑞，元培物理方向，2023年秋】

思路：

我首先想到的是全部取余数，因为0的区间是不需要考虑的这样可以节省很多时间，接下来我一开始是从非零元素开始遍历，这样的话我在test9左右就超时了，后来想到这个最长列表一定是从一端开始的，因为不可能加左边的剩余求和被整除，加右边的也被整除。

代码

```
1 '''
2 刘思瑞 2100017810
3 '''
4 def calcu(lx,x,array):
5     nonzero = []
6     for i in range(lx):
7         array[i] %= x
8         if array[i] != 0:
9             nonzero.append(i)
10    sum = -1
11    if len(nonzero) == 0:
12        return sum
```

```

13     if len(nonzero) ==1:
14         return le
15     sum = nonzero[0] +1
16     nsum = 0
17     for j in range(0,len(nonzero)):
18         nsum += array[nonzero[j]]
19         if nsum % x != 0:
20             if j+1 < len(nonzero):
21                 if nonzero[j+1] >sum:
22                     sum = nonzero[j+1]
23             else:
24                 return le
25     return sum
26
27
28 n = int(input())
29 for i in range(n):
30     le , x = map(int,input().split())
31     array = list(map(int,input().split()))
32     print(max(calcu(le,x,array),calcu(le,x,array[::-1])))

```

代码运行截图

By meinvader, contest: Codeforces Round 649 (Div. 2), problem: (A) XXXXX, **Accepted**, #, [Copy](#)

```

...
刘思瑞 2100017810
...
def calcu(le,x,array):
    nonzero = []
    for i in range(le):
        array[i] %= x
        if array[i] != 0:
            nonzero.append(i)
    sum = -1
    if len(nonzero) ==0:
        return sum
    if len(nonzero) ==1:
        return le
    sum = nonzero[0] +1
    nsum = 0
    for j in range(0,len(nonzero)):
        nsum += array[nonzero[j]]
        if nsum % x != 0:
            if j+1 < len(nonzero):
                if nonzero[j+1] >sum:
                    sum = nonzero[j+1]
            else:
                return le
    return sum

n = int(input())
for i in range(n):
    le , x = map(int,input().split())
    array = list(map(int,input().split()))
    print(max(calcu(le,x,array),calcu(le,x,array[::-1])))

```

3. 学习总结和收获

如果作业题目简单，有否额外练习题目，比如：OJ“每日选做”中每天推出的3个题目、CF、洛谷等网站题目。

月考前面七个题目还是比较简单的，第八题我感觉我做起来纯属侥幸，还是对代码结构和题目不够熟练，我是10号ddl做的大概一共两个半小时，当然主要在最后一题上