

HW#5

经济学院 刘安澜 1700015495

112A. Petya and Strings, implementation/strings, 800, x75000+

<http://codeforces.com/problemset/problem/112/A>

解题思路：先把所有的字母都更改为小写字母方便比较；python 的字符串可以比较大小，运用该命令即可。

```
a=input().lower()
b=input().lower()
if a>b:
    print(1)
elif a<b:
    print(-1)
else:
    print(0)
```

263A. Beautiful Matrix, implementation, 800, x75000+

<http://codeforces.com/problemset/problem/263/A>

解题思路：需要分成行列两次循环来确定元素 1 的位置；输出时需要注意列表是从 0 开始，所以与中间元素的距离实际是与 2 做比较。

```
for i in range(5):
    l=input().split()
    for j in range(5):
        if l[j]=='1':
            print(abs(i-2)+abs(j-2))
            break
```

339A. Helpful Maths, greedy/implementation/sortings/strings, 800, x71000+

<http://codeforces.com/problemset/problem/339/A>

解题思路：按照 '+' 号先对字符串进行切分，然后再将切分后的字符串排序，最后再用 join 函数将排序后的字符串连起来。此题切分的仅为 1, 2, 3，这样的做法没有问题；但是需要注意到如果数字大于 10 的话，字符串的排序和阿拉伯数字排序不同！

```
l=input().split('+')
print('+'.join(sorted(l)))
```

281A. Word Capitalization, implementation/strings, 800, x68000+

<http://codeforces.com/problemset/problem/281/A>

解题思路：需要将字符串第一个元素与之后的元素分开做；只将第一个元素大写，再加上原字符串的后半部分即可。注意如果使用 capitalize() 函数会将后面的元素便为小写！

```
l=input()
print(l[0].upper()+l[1:])
```

//选做:

230B. T-primes, binary search, implementation, math, number theory, 1300, x29581

<http://codeforces.com/problemset/problem/230/B>

解题思路:

需要打出数据范围内的质数表, 然后输入的数据根据下标查找它的平方根是不是质数即可。质数表需要按照 sieve of Eratosthenes 素数筛选法, 循环找出 10^6 内的质数即可。

根据提示, 我先自己在网上学习了 sieve of Eratosthenes 素数筛选法, 并且根据筛法的思路自己写了一遍代码, 然后反复修改程序运行时间长的问题, 最后终于在 python3 上 AC, 过程还是比较困难的, 但是也让我对算法的时间复杂度有了更深入的认识。

My Submissions							
#	When	Who	Problem	Lang	Verdict	Time	Memory
97045116	Oct/29/2020 10:44UTC+8	anlanamy	B - T-primes	Python 3	Accepted	1622 ms	9300 KB
97042290	Oct/29/2020 09:12UTC+8	anlanamy	B - T-primes	Python 3	Accepted	1402 ms	12800 KB
97042247	Oct/29/2020 09:10UTC+8	anlanamy	B - T-primes	Python 3	Accepted	1372 ms	12700 KB
97042207	Oct/29/2020 09:08UTC+8	anlanamy	B - T-primes	Python 3	Wrong answer on test 7	1184 ms	12600 KB
97009088	Oct/28/2020 22:26UTC+8	anlanamy	B - T-primes	Python 3	Time limit exceeded on test 7	2000 ms	9300 KB
97008842	Oct/28/2020 22:23UTC+8	anlanamy	B - T-primes	Python 3	Time limit exceeded on test 7	2000 ms	9300 KB
97008571	Oct/28/2020 22:19UTC+8	anlanamy	B - T-primes	Python 3	Time limit exceeded on test 7	2000 ms	9300 KB
97007480	Oct/28/2020 22:06UTC+8	anlanamy	B - T-primes	Python 3	Time limit exceeded on test 7	2000 ms	9800 KB
97007411	Oct/28/2020 22:05UTC+8	anlanamy	B - T-primes	PyPy 3	Accepted	1152 ms	23900 KB
97006598	Oct/28/2020 21:56UTC+8	anlanamy	B - T-primes	PyPy 3	Runtime error on test 5	404 ms	2100 KB
96861990	Oct/27/2020 21:31UTC+8	anlanamy	B - T-primes	PyPy 3	Accepted	1028 ms	27300 KB
96861917	Oct/27/2020 21:31UTC+8	anlanamy	B - T-primes	Python 3	Time limit exceeded on test 18	2000 ms	12900 KB
96855844	Oct/27/2020 20:23UTC+8	anlanamy	B - T-primes	Python 3	Time limit exceeded on test 18	2000 ms	12700 KB
96855696	Oct/27/2020 20:22UTC+8	anlanamy	B - T-primes	PyPy 3	Accepted	964 ms	27300 KB
96831357	Oct/27/2020 14:38UTC+8	anlanamy	B - T-primes	PyPy 3	Time limit exceeded on test 7	2000 ms	24300 KB
96831223	Oct/27/2020 14:36UTC+8	anlanamy	B - T-primes	Python 3	Time limit exceeded on test 7	2000 ms	24700 KB
96830935	Oct/27/2020 14:32UTC+8	anlanamy	B - T-primes	Python 3	Time limit exceeded on test 1	2000 ms	20400 KB
96828997	Oct/27/2020 14:01UTC+8	anlanamy	B - T-primes	Python 3	Time limit exceeded on test 5	2000 ms	19800 KB
96828758	Oct/27/2020 13:57UTC+8	anlanamy	B - T-primes	Python 3	Wrong answer on test 2	186 ms	300 KB

在尝试的过程当中总结如下:

- 1、质数表中的数据直接设为 bool 类型, 可以节省内存;
- 2、直接按照下标修改全表 (并查询全表) 比建立新的质数列表 (再进行在质数列表中查询) 快;
- 3、由于大部分的数不是完全平方数, 所以分两步的判断条件比一步判断更快;
- 4、将输入列表排序找最大数可能反而会增加程序运行时长, 因为排序的时间复杂度大概在 $O(n)$ 左右
- 5、解决 python3 超时的关键原因: 其实找到 10^6 以内的质数只需要在 10^3 以内遍历就可以, 因为根据 sieve of Eratosthenes 素数筛选法, 我们把不大于 i 质数的倍数都去除后, 实际上已经将 i^2 内的全部非质数剔除。这样只循环 10^3 次, 就大大节省时间了。

解题代码:

```
//1. (python3 1372ms)
```

```
l=[True]*1000001
l[0]=l[1]=False
for i in range(1001):
    if l[i]==True:
```

```

        for j in range(2*i,1000001,i):
            l[j]=False
n=int(input())
k=[int(x) for x in input().split()]
for i in range(n):
    if k[i]**0.5!=int(k[i]**0.5):
        print('NO')
    else:
        print('YES' if l[int(k[i]**0.5)]==True else 'NO')

```

//2. (python3 1402ms)

```

l=[True]*1000001
l[0]=l[1]=False
for i in range(1001):
    if l[i]==True:
        for j in range(2*i,1000001,i):
            l[j]=False
n=int(input())
k=[int(x) for x in input().split()]
for i in range(n):
    print('YES' if int(k[i]**0.5)==k[i]**0.5 and
l[int(k[i]**0.5)]==True else 'NO')

```

//3. (python3 1622ms)

```

n=int(input())
k=[int(x) for x in input().split()]
m=max(k)
m1=int(m**0.5+1)
l=[True]*m1
l[0]=l[1]=False
for i in range(int(m1**0.5+1)):
    if l[i]==True:
        for j in range(i,m1-i,i):
            l[i+j]=False
for i in range(n):
    if k[i]**0.5!=int(k[i]**0.5):
        print('NO')
    else:
        print('YES' if l[int(k[i]**0.5)]==True else 'NO')

```
