

1. 题目名称：批阅奏章(pass)

某朝皇帝有大臣 n 名 ($1 \leq n \leq 1000$)，分别编号大臣 $1 \sim n$ 。某日皇帝身体抱恙，奏章堆积如山无法及时一一批阅，便命身旁内侍帮他按指定顺序排序后再阅。于是皇帝亲自挑选了几个值得信赖的重臣并排好序，要求把他们的奏章按排好的顺序放到前面，其他的按照编号升序排列即可。现在要求你写一个程序来帮皇上解决这个问题，即已知奏章总数和顺序、钦点重臣的排列顺序，求得皇帝查阅奏章的顺序。

输入描述：

第一行输入两个整数 p ($1 \leq p \leq 5000$) 和 q ，其中 p 表示堆积奏章的总数、 q 表示皇帝钦点重臣数

第二行输入 p 个数，表示所有按呈递顺序递上来的奏章来自于哪个大臣（大臣编号）

第三行输入 q 个数，表示皇帝钦点并排好序的重臣编号

输出描述：

输出奏章按指定顺序排好序后，皇帝按大臣编号批阅的顺序

输入样例：

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

输出样例：

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

```
p,q = list(map(int,input().split(' ')))
P = list(map(int, input().split(' ')))
Q = list(map(int, input().split(' ')))
RES=[]
for i in Q:
    cnt = P.count(i)
    RES.extend([i]*cnt)
    while i in P:
        P.remove(i)
P.sort()
RES.extend(P)
print(' '.join(list(map(str,RES))))
```

2. 题目名称：报价(pass)

时间限制：1000ms 内存限制：256M

题目描述

给定某股票每日的报价和一个目标值，请在所有报价中找出和为目标值的那两天的报价，并打印出对应的报价。假设每种输入只会对应一个答案，且每日的报价不会重复。

你需要按报价从小到大的顺序打印答案。

输入描述：

输入：第一行是某股票每日的报价，这些报价是正整数且用空格相隔，例如：17 20 33

第二行是目标值，例如：37

输出描述：

输出：对应的报价，报价之间用空格相隔，例如：17 20

示例

示例 1

输入

17 20 33

37

输出 17 20

```
P = list(map(int,input().split(' ')))
n = int(input())
n1,n2 = 0,0
for i in range(len(P)-1):
    n1 = P[i]
    for j in range(i+1, len(P)):
        if n1+P[j]==n:
            n2=P[j]
            break
    if n2>0:
        break
print('%d %d'%(min(n1,n2),max(n1,n2)))
```

3. 题目名称：水仙花数(pass)

时间限制：1000ms 内存限制：256M

题目描述

如果一个 3 位数等于其各位数字的立方和，则称这个数为水仙花数。

例如： $153 = 1^3 + 5^3 + 3^3$ ，因此 153 就是一个水仙花数

输入描述：

输入一个整数 a 与一个整数 b，用空格分隔

输出描述：

输出 a 到 b 区间内的水仙花数

示例 示例 1

输入 100 170

输出 153

```
def f(n):
    N = list(map(int,str(n)))
    if n==N[0]**3+N[1]**3+N[2]**3:
        return True
    else:
        return False
n1,n2 = list(map(int,input().split()))
RES=[]
for n in range(n1,n2+1):
    if f(n):
        RES.append(n)
print(' '.join(list(map(str,RES))))
```

4. 题目名称：字符串查找和比较(pass)

时间限制：1000ms 内存限制：256M

题目描述

写函数实现如下功能，给定字符串 A 和 B, 输出 A 和 B 中的最长公共子串。比如 A="aocdfe" B="pmcdfa" 则输出 "cdf"。

输入描述:

输入待处理的两个字符串 str1, str2

输出描述:

找出两个字符串最长的公共子串

示例 示例 1

输入

aocdfe

pmcdfa

输出

Cdf

```
s1 = input().strip()
s2 = input().strip()

if len(s1)>len(s2):
    s2,s1 = s1,s2
res = ""
for i in range(len(s1)):
    j=len(s1)
    while j>i:
        _str = s1[i:j]
        if s2.find(_str)>=0:
            if len(_str)>len(res):
                res=_str
        j-=1
print(res)
```

5. 题目名称：尼姆博弈(pass)

时间限制：1000ms 内存限制：256M

题目描述

你和你的朋友，两个人一起玩 Nim 游戏：

桌子上有一堆石头。

你们轮流进行自己的回合，你作为先手。

每一回合，轮到的人拿掉 1 - 3 块石头。

拿掉最后一块石头的人就是获胜者。

假设你们每一步都是最优解。请编写一个函数，来判断你是否可以在给定石头数量为 n 的情况下赢得游戏。如果可以赢，返回 true；否则，返回 false。

输入描述:

整数 n

输出描述:

true 或 false

示例

示例 1

输入

4

复制

输出

false

```
n = int(input().strip())
def f(n):
    if n%4==0:
        return False
    else:
        return True
if f(n):
    print('true')
else:
    print('false')
```

6. 题目名称：罗马数字转整数(pass)

时间限制：1000ms 内存限制：256M

题目描述

罗马数字包含以下七种字符：I， V， X， L， C， D 和 M。

字符	数值
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

例如， 罗马数字 2 写做 II ，即为两个并列的 1 。12 写做 XII ，即为 X + II 。 27 写做 XXVII，即为 XX + V + II 。

通常情况下，罗马数字中小的数字在大的数字的右边。但也存在特例，例如 4 不写做 IIII，而是 IV。数字 1 在数字 5 的左边，所表示的数等于大数 5 减小数 1 得到的数值 4 。同样地，数字 9 表示为 IX。这个特殊的规则只适用于以下六种情况：

I 可以放在 V (5) 和 X (10) 的左边，来表示 4 和 9。

X 可以放在 L (50) 和 C (100) 的左边，来表示 40 和 90。

C 可以放在 D (500) 和 M (1000) 的左边，来表示 400 和 900。

给定一个罗马数字，将其转换成整数。

输入描述：

罗马数字

输出描述：

转换后的整数

示例

示例 1

输入 III

输出 3

```
d_rm={'I':1, 'V':5, 'X':10, 'L':50, 'C':100, 'D':500, 'M':1000}
in_rm = input().strip()
def rm2num(in_rm):
    if not in_rm:
        return 0
    if len(in_rm)==1:
        return d_rm.get(in_rm, 0)
    res = 0
    for i in range(len(in_rm)-1):
        n1 = d_rm.get(in_rm[i],0)
        n2 = d_rm.get(in_rm[i+1],0)
        if n1<n2:
            res-=n1
        else:
            res+=n1
    res+=n2
    return res
print(rm2num(in_rm))
```

7. 题目名称：员工排序(pass)

时间限制：1000ms 内存限制：256M

题目描述

某公司要针对员工情况统一调薪，现有一个员工列表，包含姓名、职级、工作年限、工资信息。现要求将该列表按以下规则重新排序：

- 1、职级高的排在前面
- 2、若职级相同，按工资少的排前面
- 3、若工资相同，工作年限长的排前面

输入描述：

第一行输入一个整数 n，表示员工列表中员工的人数

接下来的 n 行，分别输入员工的姓名、职级、工作年限和工资信息，各项信息用空格分隔

输出描述：

输出按要求重新排列的员工列表，每名员工信息（姓名、职级、工作年限和工资）占一行

示例

示例 1

输入

6

张三 3 3 3000

李四 3 4 3000

王五 3 3 4000

赵六 4 3 3000

陆奇 4 4 4000

闫八 4 4 3980.99

复制

输出

赵六 4 3 3000.00
闫八 4 4 3980.99
陆奇 4 4 4000.00
李四 3 4 3000.00
张三 3 3 3000.00
王五 3 3 4000.00

```
n = int(input())
P = []
for i in range(n):
    _p = input().split(' ')
    P.append([_p[0],int(_p[1]),int(_p[2]),float(_p[3])])
P.sort(key=lambda x:(-x[1],x[3],-x[2]))
for p in P:
    print('%s %d %d %.2f'%(p[0],p[1],p[2],p[3]))
```

8. 题目名称：汉诺塔 (pass)

时间限制：1000ms 内存限制：256M

题目描述

有三个立柱 A、B、C。A 柱上穿有大小不等的圆盘 N 个，较大的圆盘在下，较小的圆盘在上。要求把 A 柱上的圆盘全部移到 C 柱上，保持大盘在下、小盘在上的规律（可借助 B 柱）。每次移动只能把一个柱子最上面的圆盘移到另一个柱子的最上面。请输出移动过程。

输入描述：

输入一个整数 n

输出描述：

输出移动过程

示例 示例 1

输入 3

输出

a->c

a->b

c->b

a->c

b->a

b->c

a->c

```
def hanoi(n, A, B, C):
    if n == 1:
        print('%s->%s'%(A,C))
    else:
        hanoi(n - 1, A, C, B)
        print('%s->%s'%(A,C))
        hanoi(n - 1, B, A, C)

n = int(input())
```

9. 题目名称：交换数值(pass)

时间限制：1000ms 内存限制：256M

题目描述

将整数 a、b 的值进行交换，并且不使用任何中间变量。

输入描述：

输入待处理的两个整数 a 和 b

输出描述：

a 和 b 的值交换

示例

示例 1

输入

a=100

b=200

复制

输出

a=200

b=100

```
a = int(input().split('=')[1])
b = int(input().split('=')[1])
a,b = b,a
print('a=%d'%a)
print('b=%d'%b)
```

10. 题目名称：进程排序(pass)

某系统中有 n 个进程，每个进程都有自己唯一的进程 id (PID)，同时每个进程最多还有一个父进程，父进程 id 为 (PPID)，和一个或多个子进程。

若某进程没有父进程，则 PPID 为 0。当某一进程被终止时，其子进程也将被终止。

现给出进程 id 列表和起对应的父进程 id 列表，当要终止某一进程时，计算最终会终止哪些进程，并将要终止的 PID 按升序排列。

输入描述：

第一行输入两个整数 n 和 k，n 表示当前系统中运行的进程数；k 表示要终止进程的 PID

第二行输入 n 个正整数，表示进程列表，每个整数表示进程的 PID

第三行输入 n 个正整数，表示进程列表中的进程对应的父进程 PPID 列表。

输出描述：

输出当进程 k 终止时，所有会被终止的进程 PID，并按 PID 升序排列，每个 PID 用空格分隔。

输入样例：

4 5

1 3 10 5

3 0 5 3

输出样例：5 10

```
n,k = map(int, input().split(' '))
PID = list(map(int, input().split(' ')))
```

```

PPID = list(map(int, input().split(' ')))
d_ppid = {}
for i in range(len(PPID)):
    _ppid = PPID[i]
    _pids = d_ppid.get(_ppid, [])
    _pids.append(PID[i])
    d_ppid[_ppid] = _pids
KILL_PPID=set([k])
KILL_ALL=set([])
while len(KILL_PPID)>0:
    _p_id = KILL_PPID.pop()
    KILL_ALL.add(_p_id)
    _c_ids = d_ppid.get(_p_id, [])
    for _cid in _c_ids:
        KILL_PPID.add(_cid)
        KILL_ALL.add(_cid)

res = list(KILL_ALL)
res.sort()
res = list(map(str, res))
print(' '.join(res))

```

11. 题目名称：丑数(pass)

时间限制：1000ms 内存限制：256M

题目描述

给你一个整数 n ，请你判断 n 是否为丑数。如果是，返回 true；否则，返回 false。

丑数 就是只包含质因数 2、3 和/或 5 的正整数。(1 通常被视为丑数)

输入描述：

整数 n

输出描述：true 或 false

示例 示例 1

输入 6

输出 true

```

n = int(input())
def f(n):
    if n<1:
        return False
    PN = set([])
    while n>1:
        for i in range(2,n+1):
            if n%i==0:
                n=n//i
                PN.add(i)
            break

    for i in [2,3,5]:

```



```
        if i in PN:
            PN.remove(i)
    if len(PN)>0:
        return False
    else:
        return True
if f(n):
    print('true')
else:
    print('false')
```