

python 高年级

设置安全密码

题目描述：

我们登录各种网站时，常常需要注册用户名和密码。有的同学会产生疑问，要怎样才能设置一个更安全一点的密码呢？

一般来说，一个比较安全的密码应该至少满足以下两个条件：

- (1) 密码长度大于等于 8，且不要超过 16；
- (2) 密码中的字符应该至少来自下面“字符类别”四组中的三组。

这四组字符类别分别为：

- 1. 大写字母：A、B、C、...、Z
- 2. 小写字母：a、b、c、...、z
- 3. 数字：0、1、2、...、9
- 4. 特殊符号：~、!、@、#、\$、%、^

给你一个密码，你的任务是判断它是不是一个安全的密码。

输入：

第一行，一个整数 T，

接下来 T 行，每行一个字符串，表示一个密码。

输出：

共 T 行，每行输出 True 或 False，表示这个密码安全或不安全。

输入样例：

3 0123Ab! abcAbc!@# 12ab09cd3412

输出样例:

False True False

用时/内存:

1000MS/100MB

```
t = int(input())
```

```
for _ in range(t):
    s = input()
    if not (8 <= len(s) <= 16):
        print("False")
        continue

    count = 0
    count += any(c.isupper() for c in s)
    count += any(c.islower() for c in s)
    count += any(c.isdigit() for c in s)
    count += any(c in "~!@#$%^" for c in s)
    if count >= 3:
        print("True")
    else :
        print("False")
```

小蓝的零食柜

题目描述:

小蓝非常喜欢吃零食，他的零食柜里有许多零食。每天，他可能会买一些零食放进柜子里，也可能会吃掉一些零食。你的任务是帮助小蓝计算到最后，他的零食柜里还剩多少零食。

输入:

第一行输入一个整数 n ，表示零食柜初始的零食数量。 ($1 \leq n \leq 1000$)

第二行输入一个整数 m ，表示接下来有 m 天操作。 ($1 \leq m \leq 10$)

接下来的 m 行，每行包含一个整数：

正数表示当天购买的零食数量。

负数表示当天吃掉的零食数量。（不会出现负库存的情况）

输出：

输出一个整数，表示零食柜里最终剩下的零食数量。

输入样例：

50 4 20 -10 15 -5

输出样例：

70

用时/内存：

1000MS/100MB

```
n = int(input())  
m = int(input())
```

```
cur = n
```

```
for _ in range(m):  
    operation = int(input())  
    cur += operation
```

```
print(cur)
```

100 以内加法

题目描述：

实现 100 以内加法，输入两个自然数 A 和 B，输出他们的和。

输入：

第一行输入 A，第二行输入 B

输出：

如果 A+B、A、B 不在范围 0 至 100（包括 0 和 100）以内，则输出“不符合要求”，满足要求则输出 A+B 的和

输入样例：

27 29

输出样例:

56

用时/内存:

1000MS/100MB

```
a = int(input())
```

```
b = int(input())
```

```
total = a + b
```

```
if 0 <= a <= 100 and 0 <= b <= 100 and 0 <= total <= 100:
```

```
    print(total)
```

```
else:
```

```
    print("不符合要求")
```

数位相加

题目描述:

输入一个正整数 A (小于 1000000)，把 A 所有数位上的数字相加得 A1，如果 A1 大于 10 则继续将 A1 所有数位上的数字相加得 A2，依次类推，直到最后的和小于 10。

输入:

第一行输入 A

输出:

把 A 所有数位上的数字的相加得 A1，如果 A1 大于 10 则继续将 A1 所有数位上的数字相加得 A2，依次类推，直到最后的和小于 10，输出该值

输入样例:

11

输出样例:

2

用时/内存:

1000MS/100MB

```
n = int(input())
while n >= 10:
    sum_digits = 0
    while n > 0:
        sum_digits += n % 10
        n //= 10
    n = sum_digits
print(n)
```

图像加密

题目描述:

加密系统收到 $m \times m$ 的平面图后，需要把这个图像顺时针旋转 90 度，完成加密。

请你编写程序，实现加密效果。

具体旋转效果，参见展示样例。

输入:

第一行，一个整数 m

接下来 m 行，每行 m 个整数，整数之间用空格隔开

(输入的 $m * m$ 个整数表示一幅图像)

输出:

顺时针旋转 90 度后的图像

输入样例:

3 1 2 3 4 5 6 11 13 45

输出样例:

11 4 1 13 5 2 45 6 3

用时/内存:

1000MS/100MB

```
m = int(input())
grid = []
for i in range(m):
    grid.append(input().split())

for j in range(m):
    for i in range(m - 1, -1, -1):
        print(grid[i][j], end=" ")
    print()
```