凯撒密码

凯撒大帝是杰出的军事家，为后来的罗马帝国开疆拓土，打了无数胜仗。他在行军时经常要给士兵们传达指令，而万一这些指令被敌人的间谍截获，敌人获悉了他们的举动，打仗就非常被动。

公元前 58 年左右，凯撒大帝想出了一个办法，他决定把自己的指令加密。他告诉自己的士兵们，出征后，指令的每个字母都会往右移 3 格。这样就算敌人截获了信息，看到的也是无法理解的一堆字母。这是早期流传下来的密码。

至此之后，但凡要传递不想被别人知道信息，大家都开始用密码。大家约定好一个规则，给**明文加密**，明文变成**密文**传递。收到密文后，根据规则**解密就可以**转回明文了。

只要学好逻辑和数学，早期密码还是比较容易被破译的。

到了第二次世界大战时，纳粹德国发明出英格玛机器（Enigma），可以用机器加密，速度和复杂度比之前的密码都强大得多。这个加密方法成了纳粹领袖希特勒的重要武器。

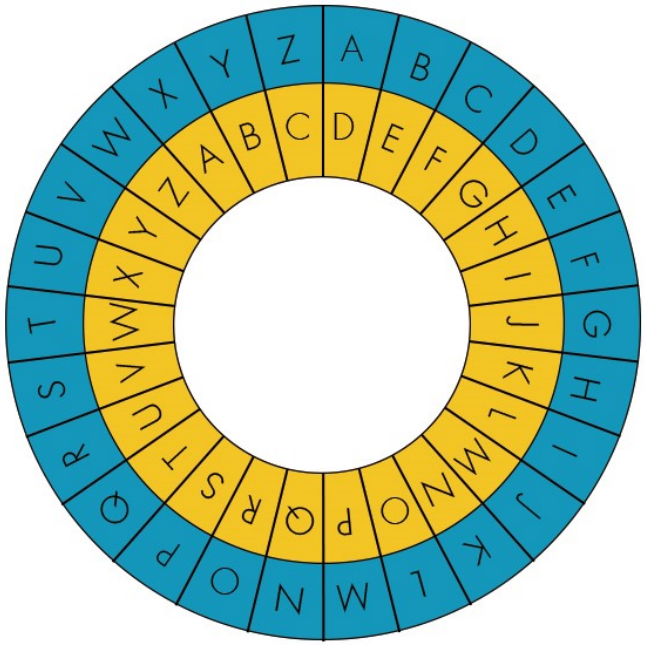
英国的通信处召集了包括“计算机科学家之父”艾伦·图灵（Alan Turing）在内的多位计算机科学家、数学家、语言学家，经过艰难的工作，最终成功设计出能够破解英格玛的机器，帮助盟军取得二战的胜利。

除了军事用途外，通信行业也使用编码，比方说早期电报就用摩斯密码。现如今计算机和互联网渗透我们生活中的方方面面，如何保护好银行卡、网络账户的密码等重要信息，是现代密码学的关键。

1. 凯撒密码介绍

凯撒密码是一种替换密码，明文里的每个字母都用其它字母替换。替换的规律是将字母往右移动几格。具体移动的数字我们称之为**密匙**。

相传凯撒大帝每次会移动字母 3 格，他的密匙是 3。所以它的 a 会变成 d，b 会变成 e，以此类推。



为了方便数格子，我们列出每个字母所对应的数字，也就是字母表字符串的索引值位置。

alphabet = “abcdefghijklmnopqrstuvwxyz”
alphabet[0]会返回“a”

alphabet[15]会返回“p”

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12
n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

如果明文是“apple”按密匙为 3 来看，对应的密文是“dssoh”。我们找到每个字母对应的数字，加上 3，得到密文对应的数字，找到对应数字的字母。

明文	a	p	p	l	e
明文对应数字	0	15	15	11	4
密匙 3，移动 3 格	+3	+3	+3	+3	+3
密文对应数字	3	18	18	14	7
密文	d	s	s	o	h

2. 编写凯撒密码

编写一段代码，让用户输入明文，再让用户输入密匙的数字，然后打印出密文。

```
请输入明文>>>apple
请输入密匙数字>>>3
你的密文是： dssoh
```

凯撒密码程序 1

1. 定义函数 encrypt，带参数明文字符串 original 和密匙 shift
   1) 将字母表存在变量 alphabet 里
   2) 用变量 code 装密文，初始时为空字符串
   3) 对于明文 original 里的每个字，用循环重复下面的代码：
      a) 找到每个字在字母表里的索引值，存进 num 里
      b) 给这个索引值加上密匙数字 shift
      c) 找到这个数字除以 26 的余数，得到密文对应的数字，存进 newNum 里
      d) 找到 newNum 对应的字母，添加进 code 里
   4) 当 original 里的每一个字母都经过 for 循环加密后，返回装密文的 code

2. 用 input()让用户输入明文，将明文字符串存进变量 text
3. 用 input()让用户输入密匙，将密匙字符串存进变量 key
4. 用 int()将 key 里的字符串转换成数字
5. 调用 encrypt 函数，将 text 变量里的明文字符串和 key 里的密匙数字作为参数，将返回的密文值存进变量 secret
6. 打印密文

```
def encrypt(original, shift):
    alphabet =
"abcdefghijklmnopqrstuvwxyz"
    code = ""
    for x in original:
        num = alphabet.find(x)
        newNum = num + shift
        newNum = newNum % 26
        code = code + alphabet[newNum]
    return code


text = input("请输入明文>>>")
key = input("请输入密匙数字>>>")
key = int(key)
secret = encrypt(text, key)
print("你的密文是：", secret)
```

注意，这里 newNum 有一行 newNum % 26 的处理。原因是 alphabet 里对应字母的数字只有 0 ~ 25。但我们在做加法时，偶尔会超出 25， 这时候我们需要重新回到 0 来数数。

比方说如果要将 w 往右移动 5 格。w 移动完 3 格后，第 4 格 回到 a，第 5 格到 b。按照我们的算法，22 + 5 = 27。我们用 27 % 26 可以得到 1，1 对应 b。不论数字多大，只要使用%26，我们就可以得到 0 ~ 25 之间和它对应的数字和字母。

编写好后保存，尝试运行代码。看看如果输入 attack tomorrow，是否可以得到下面的结果？

```
请输入明文>>>attack tomorrow
请输入密匙数字>>>5
你的密文是： fyyfhpeytrtwwtb
```