

主题三

用算法解决问题

壮壮和小伙伴决定运用算法设计一款可以查询时间且能与人互动的小程序“小智”。在设计过程中，他们学会了运用加密算法保护信息安全，用二分法快速猜数，用多分支结构判断闰年与平年……

任务导航

学习简单的加密算法，了解其在保护信息安全方面的意义。

互传密信有诀窍

探究闰年与平年的判断规则，设计并描述判断闰年与平年的算法，体会分支结构在生活实例中的应用。

闰年平年我知道（1）

猜数游戏有捷径

参与猜数游戏，运用二分法策略完成猜数挑战，感受利用算法解决问题的优势。

闰年平年我知道（2）

学习并运用多分支结构的程序语句，完整经历用算法解决问题的全流程。



学习评价

目标挑战	评价指标			收获 ★ 数量
	★	★★	★★★	
互传密信 有诀窍	了解移位密码基本规则，能识别简单示例，看懂加密算法流程图，区分明文和密文，有初步的信息安全意识	掌握移位密码规则，能阐述加密原理，完成简单的加密操作，理解明密文转换关系，能举例说明加密算法在生活中的应用	熟练掌握移位密码及其算法，能根据不同移位值进行加密解密，并能针对不同场景下的加密应用提出建议	
猜数游戏 有捷径	能说出二分法的基本思想，从简单情境中确定查找对象、范围和判断条件	知道二分法查找的基本步骤，在查找时能明确算法执行次数和数据规模的关系，能举例说明二分法在生活中的应用	深入理解二分法思想，能分析二分法效率和数据规模的关系，熟练用二分法解决实际问题，清晰阐述用算法解决问题的优势	
闰年平年 我知道（1）	能从简单问题情境中提取关键因素，确定判断条件，通过描述判断闰年平年的算法，知晓算法输入、判断条件及输出	清晰描述两个双分支结构，准确识别其中判断条件，结合生活实例说明分支结构的应用	理解算法多分支结构，能分析判断条件对算法结果的影响，熟练用分支结构解决复杂问题，知道算法的输入、判断条件以及输出	
闰年平年 我知道（2）	能识别 Python 分支结构判断条件，区分不同执行路径，初步了解利用计算机解决问题的基本方法	能用 Python 程序验证简单算法，补全多分支结构程序并运行，会用多分支结构解决简单的实际问题	掌握 Python 多分支结构，能灵活运用多分支结构解决复杂问题	
在本主题的学习中，我一共获得_____颗★				



任务一 互传密信有诀窍

壮壮团队首先为“小智”设计用户注册账号并设置密码的功能。壮壮试用时将密码设置成“zhuang”，经过计算机加密处理后，后台存储密码的信息变成了“ckxdqj”，这是为什么呢？



学一学

在信息加密过程中，原始数据称为明文，加密后的数据称为密文。壮壮设置的密码明文是“zhuang”，密文是“ckxdqj”。

壮壮团队选用了移位加密算法存储用户信息。这里的加密规则采用的是凯撒密码，它是一种简单的移位加密算法。

加密时，明文中的所有字母在字母表中向右移动三位后被替换成密文，如果移动后的字母超过了字母表最后的字母 Z，则回到字母表的开头，继续向右移动。

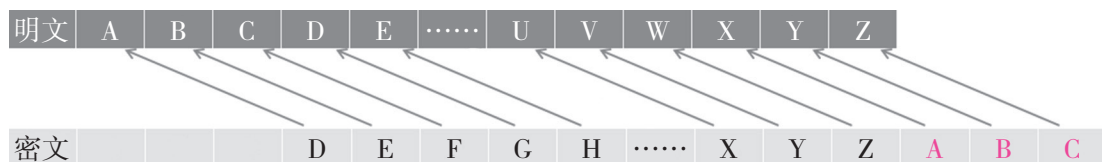
加密过程

明文	A	B	C	D	E	U	V	W	X	Y	Z			
密文				D	E	F	G	H	X	Y	Z	A	B	C



解密是加密的反向操作。解密时，将密文中的所有字母在字母表中向左移动三位后就得到明文。如果移动后的字母超过了字母表最前面的字母 A，则需要回到字母表的最后，继续向左移动。

解密过程



练一练

1. 壮壮设计了一段密文“XZFSKF”，已知明文中的字母向右移动五位后可以得到密文，那么明文是 SUANFA。
2. 假设你收到一封用凯撒密码加密的信，部分密文是“oryh”，且你知道这部分密文对应的明文是“love”，那么加密规则是 将明文中的字母向右移动三位。根据此规则加密“world”这个单词，得到的是 zruog。



试一试

我们以移动三位的加密算法为例，用自然语言进行如下描述，请你根据下面左侧自然语言的描述补全右侧的流程图。可以看出，这是一个 循环（选填“顺序”、“循环”或“分支”）结构的算法。



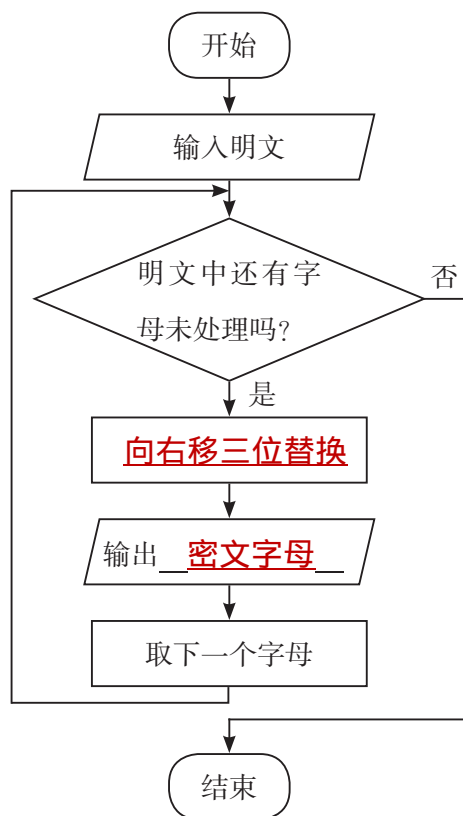
自然语言描述

第1步：输入要传送的明文。

第2步：查看第一个字母，将字母向右移动三位并输出对应的字母。

第3步：重复第2步操作，分别判断并输出第二个字母、第三个字母、第四个字母……直到处理完成明文中的所有字母。

算法的流程图



小贴士

凯撒密码是移位加密算法中移位规则较为简单、固定的一种类型，是历史上真实存在的一个经典加密方法。在罗马共和时期的战争中，军事指挥官为了预防情报所表达的信息在传递过程中被敌军截获，往往会依据指定的规则对情报进行加密，确保只有知道规则的人才能解密，从而保护了重要的军情信息。

凯撒密码为移位加密算法的发展提供了基础思路，后续很多复杂的移位加密算法都是在凯撒密码的基础上，通过改进移位规则、增加密钥的复杂性等方式发展而来。



做一做

移位算法不仅可以移动字母，还可以移动汉字、数字、符号等不同类型的信息。以中文为例，体验用多行多列的方格实现更为复杂的移位加密算法。

第1步：将今天学习移位算法之后的心得体会（20～40个字）写在下面4行10列的方格中，作为明文。

学	了	移	位	算	法	,	和	同	学
们	一	起	加	密	和	解	密	真	是
太	有	趣	了	!					

第2步：把奇数列的文字、标点符号依次下移1行，找到对应的方格填入，下移超过最后一行的文字，回到第一行继续下移替换，构成密文。

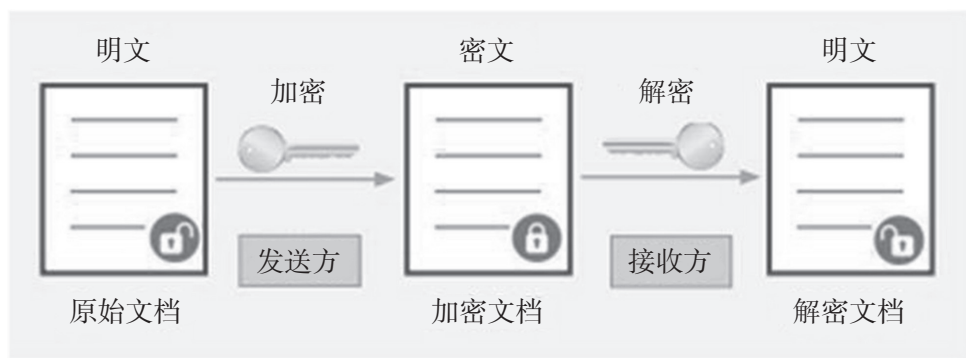
太	了	趣	位	!	法	解	和	真	学
学	一	移	加	算	和	,	密	同	是
们	有	起	了	密					

第3步：和组内的同学交流，解出对方的密文。



议一议

信息加密是保障信息安全的核心技术。在信息传输过程中，发送方通过一定的算法规则对数据进行加密后再传输，将原始信息转化为不可直接读取的密文形式，确保只有接收方凭借一定的算法规则才能将密文还原为明文而获得信息。



案例一：壮壮和小桂为同一段信息加密时，壮壮将明文的字母在字母表中移动三位，而小桂将明文的字母在字母表中移动七位，谁的加密算法更安全呢？

案例二：小漓、小柳和小邕正在为明文设计加密算法。小漓提议，可以把明文中的英文字母替换成指定中文段落中的文字；小柳却觉得，将明文中的汉字替换成字母表中特定的字母才好。请帮小邕参谋参谋，小漓和小柳的想法到底对不对？为什么？

请根据这两个案例，仔细思考分析：若想提升加密算法的安全性，可以 **增加移动的位数，使用多种类型的数据设计加密算法。**



试一试

移位密码（如凯撒密码）的密钥只有 25 种可能，现代计算机可瞬间破解，因此需要更加复杂的加密算法来保护我们的信息安全。请设计一种加密算法，然后用这个加密方式将一则信息传送给同学，并给出关于密钥的提示，看看谁能最快解读出来。

我设计的加密算法是这样的：每个字的拼音首字母替换为数字，按字母表顺序代表数字，例如：“密”的拼音首字母是m，代表数字13。

我想传递的内容是：10+8=18，用加密算法处理后的内容是：鸡+猴=肉。

小贴士

中国的“密码卫士”

国密算法是国家密码管理局发布的密码算法标准，是我国自主研发的一系列密码算法，像“数字长城”一样保护着我们的网络安全。

从 2006 年开始，我国逐步公布了 SM2、SM3、SM4、SM9、ZUC 等商用密码算法，构成了我国完整的商用密码算法体系。截至 2024 年，这些算法已全部通过 ISO/IEC 等国际标准认证，为促进国际密码学发展、丰富产业选择和保障供应链安全提供了中国方案。这也预示着，我国的密码技术已经达到国际领先水平。



任务拓展

1. 有一段采用移位加密算法加密后的密文“BYFFI”，移位值未知，但知道明文是一个常见的英文单词，尝试推理出移位值和明文。

2.《武经总要》记载，北宋时期曾公亮搜集了 40 个常用军事短语，并按顺序对这些短语进行编码，用作军事保密通信的代码，如下表所示。

编码	含义	编码	含义	编码	含义	编码	含义
一	请求弓箭	十一	请求车辆和牛	二十一	敌人众多	三十一	被敌人包围
二	请求箭矢	十二	请求船只	二十二	敌人稀少	三十二	敌人的包围解除
三	请求刀具	十三	请求攻城和守城的器具	二十三	敌我力量相当	三十三	战斗未能取胜
四	请求盔甲	十四	请求增派兵力	二十四	敌人增兵	三十四	战斗大胜
五	请求枪和旗帜	十五	请求转移阵地	二十五	敌人转移营地	三十五	战斗取得大捷
六	请求锅和帐篷	十六	请求进攻	二十六	敌人进攻	三十六	将士投降
七	请求马匹	十七	请求撤退	二十七	敌人撤退	三十七	将士叛变
八	请求衣物	十八	请求坚守	二十八	敌人坚守	三十八	士兵生病
九	请求粮食	十九	未发现敌人	二十九	包围了敌人的城池	三十九	高级将领生病
十	请求草料	二十	已发现敌人	三十	解除对城池的包围	四十	战斗小胜

军队出征前，约定用一首不含重复文字的五言律诗作为密码本，即律诗的文字顺序与常用军事短语的编码一一对应。比如本次使用的五言律诗为《使至塞上》，全文如下。

使至塞上
(唐·王维)
单车欲问边，属国过居延。
征蓬出汉塞，归雁入胡天。
大漠孤烟直，长河落日圆。
萧关逢候骑，都护在燕然。

将领如果需要补充马匹，先查出“请求马匹”编码为七，再在律诗中找到第七个字“国”，在普通公文中传递一个“国”字即可。

按照这次的约定，收到的信息是“骑”“河”“汉”“归”，将领想传达什么信息呢？如果需要请求刀具和衣物，将领应该传递哪些字呢？

将领想传达的信息是：战斗取得大捷，敌人撤退，请求增派兵力，请求进攻。
请求刀具传递“欲”；请求衣物传递“过”。



任务二 猜数游戏有捷径

壮壮为“小智”设计了一个猜数的互动游戏功能，并告诉小美“小智”猜对一个100以内的自然数最多只要7次，这是真的吗？



玩一玩

两个同学为一组进行猜数游戏，互相猜测对方的数字，并将猜数过程记录在下表中，看看谁能用更少的次数猜中。游戏规则如下：

1. 需要猜的数是一个自然数，范围在 $1 \sim 100$ ；
2. 猜测方说出一个数后，被猜测方要给出一个提示：说出的数相对于要猜的数是大了还是小了，或者是猜对了。

我心里想的数是_____。

猜测次数	对方猜的数	判断结果	剩余范围
1			
2			
3			
4			
5			
6			

续表

猜测次数	对方猜的数	判断结果	剩余范围
7			
8			
9			
10			
11			
12			
13			
14			

通过游戏发现：如果猜出的数不正确，猜数的范围就可以根据所猜数的位置逐渐 缩小（选填“扩大”或“缩小”），直到猜到符合的数。

同时还发现：猜数范围越 小（选填“大”或“小”），一次猜出指定数的可能性就越 大（选填“大”或“小”）。



试一试

打开“小智猜数（百）.py”程序，和“小智”玩猜数游戏，并把过程记录在下表中。

我心里想的数是_____。

猜测次数	“小智”猜的数	判断结果
1		
2		
3		
4		



2. 打开“小智猜数(万).py”程序,猜数的范围为 $1 \sim 10000$ 的自然数,并把你们的游戏结果记录在下表中。

游戏次数	我心里想的数	“小智”猜对用的次数
1		
2		
3		

组内交流:当猜数范围是 $1 \sim 10000$ 时,用二分法猜最多需要 14 次。

3. 打开“小智猜数(十万).py”程序,猜数的范围为 $1 \sim 100000$ 的自然数,并把你们的游戏结果记录在下表中。

游戏次数	我心里想的数	“小智”猜对用的次数
1		
2		
3		

组内交流:当猜数范围是 $1 \sim 100000$ 时,用二分法猜最多需要 17 次。

体验不同范围的二分法程序后,请为二分法查找数据的自然语言描述选择正确的内容。

第 1 步:在要查找的一组 有序数 (选填“有序数”或“无序数”)中,取中间位置的数与要查找的数比较。如果二者 相等 (选填“相等”或“不相等”),则查找成功。

第 2 步:若中间位置的数大于查找数,则进一步查找比这个数 小 (选填“大”或“小”)的部分,否则进一步查找比这个数 大 (选填“大”或“小”)的部分。

第 3 步:重复上述两个步骤,直到找到目标数或确定该数不存在。



趣味知识

二分法在有序数列的查找中效率非常高。如果需要猜的数是1亿，数值范围是1至1亿，那么我们用二分法猜大约只需要27次。假设一秒猜1次，那么用按顺序逐个猜的方式就需要约1157天（约为3年），用二分法只需不到27秒。



任务拓展

某软件的最新版本（版本号303）突然崩溃，但最旧版本（版本号1）正常。请帮助开发者快速找到最早引入该问题的版本号。

1. 先检查最中间的版本： $(1+303) \div 2=152$ 。
若版本号152崩溃：问题出在1-152之间，缩小范围到1-152。
若版本号152正常：问题出在153-303之间，缩小范围到153-303。
2. 重复步骤1，每次取当前范围的中间版本测试，逐步缩小范围，直到找到第一个崩溃的版本，即为最早引入问题的版本。



任务三 闰年平年我知道（1）

壮壮在给“小智”设计查询时间功能时，发现每年的天数并不相同。团队成员小桂说：“这我可是深有体会，虽然我已经 11 岁了，但是我只过了 2 次生日。”这是为什么呢？



查一查

使用搜索引擎工具查询近 12 年的天数，并与同伴交流，你发现了什么？

年份						
天数						
年份						
天数						

我发现：366 天每 四 年出现一次，其余年份的天数都是 365 天，闰年多的那一天是 2 月 29 日。

小贴士

地球绕太阳运行的实际周期约为 365.2422 天，而平年按 365 天计算，这样每四年就会多出将近一天的时间。为弥补历法年度（365 天）与地球实际公转周期的时间差，历法规定每 4 年增加 1 天，即闰年按 366 天计算。闰年的 2 月为 29 天，平年的 2 月为 28 天。



探一探

闰年通常每四年一次，但是不是只要能被 4 整除的年份就是闰年？我国的俗语“四年一闰，百年不闰，四百年又闰”揭示了更为复杂的规则。下面我们通过具体年份探究闰年的判断条件。

先计算这些年份是否能分别被 4、100 或 400 整除，再用网络检索查询该年是否是闰年。

年份	2025	1900	2000	2020	2100
被 4 整除	否	能	能	能	能
被 100 整除	否	能	能	否	能
被 400 整除	否	否	能	否	否
是否是闰年	否	否	是	是	否

通过探究，我们发现：

1. 如果一个年份能被 4 整除，但不能被 100 整除，那么这一年是闰年。
2. 如果一个年份能被 400 整除，那么这一年也是闰年。
3. 不满足以上条件的年份是 平 年。



填一填

用自然语言和流程图来描述判断闰年与平年的算法。

1. 自然语言描述。

(1) 根据以上结论，我们可以用“如果……那么……，否则……”的语句来描述判断闰年和平年的算法。

如果“一个年份能被 4 整除”但“不能被 100 整除”，或者“一个年份能被 400 整除”，那么结论为“是 闰”（选填“闰”



或‘平’)年”，否则结论为“是 平 (选填‘闰’或‘平’)年”。

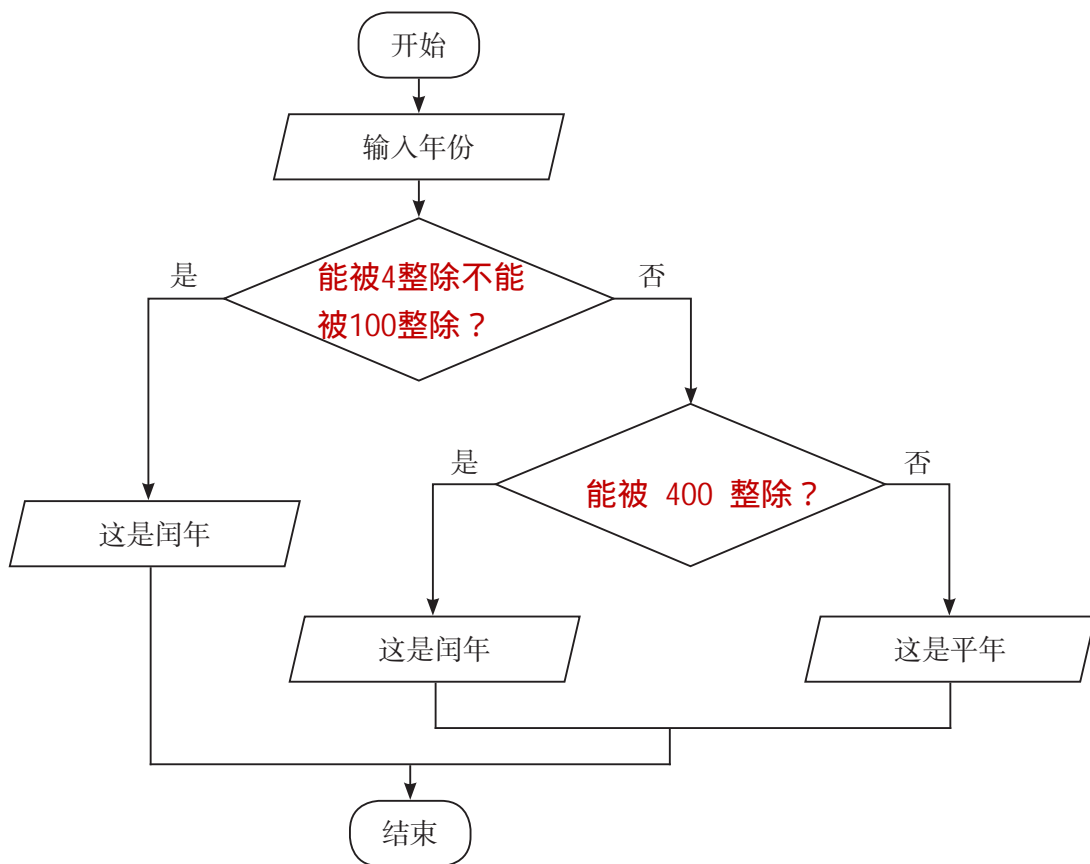
(2) “如果”后面有三个条件，这三个条件之间有什么逻辑关系呢？

前两个条件是并列关系，第三个条件和前两个条件又形成一次并列关系。

用括号可以表示为：如果(“一个年份能被4整除”但“不能被100整除”)或者“一个年份能被400整除”，那么结论为“是闰年”，否则结论为“是平年”。

2. 流程图描述。

(1) 请根据自然语言描述，把流程图中的判断条件补充完整。



(2) 选择一个年份进行判断，并根据补充好的流程图中指出执行的判断路径。

我选择需要判断的年份是_____，是_____ (选填“闰”或“平”)年。在流程图中，它的路径是_____，

_____。

（3）这个流程图中有 2 个判断条件，形成了 两 个双分支结构。

（4）组内交流，从条件判断的次数、执行内容、适用场景等方面说一说两个双分支结构、一个双分支结构和一个单分支结构的不同。

结构类型	算法流程图
两个双分支结构	<pre> graph TD Start(()) --> Cond1{条件 1 成立?} Cond1 -- 是 --> S1[语句组 1] Cond1 -- 否 --> Cond2{条件 2 成立?} Cond2 -- 是 --> S2[语句组 2] Cond2 -- 否 --> S3[语句组 3] S1 --> Join(()) S2 --> Join S3 --> Join Join --> End(()) </pre>
一个双分支结构	<pre> graph TD Start(()) --> Cond{是否满足条件?} Cond -- 是 --> S1[语句组 1] Cond -- 否 --> S2[语句组 2] S1 --> Join(()) S2 --> Join Join --> End(()) </pre>
一个单分支结构	<pre> graph TD Start(()) --> Cond{是否满足条件?} Cond -- 是 --> S[语句组] Cond -- 否 --> End(()) S --> End </pre>

①单分支结构和一个双分支结构都只判断一次，两个双分支结构要判断两次。②单分支结构条件成立时执行语句组，不成立则什么都不做；一个双分支结构条件成立执行其中一个语句组，不成立执行另一个语句组，必选其一；两个双分支结构可能有多种执行路径，最终执行其中一条路径。③单分支结构适用于“满足条件才做某事”；一个双分支结构适用于“二选一”的情况；两个双分支结构适用于“多重判断”的场景。



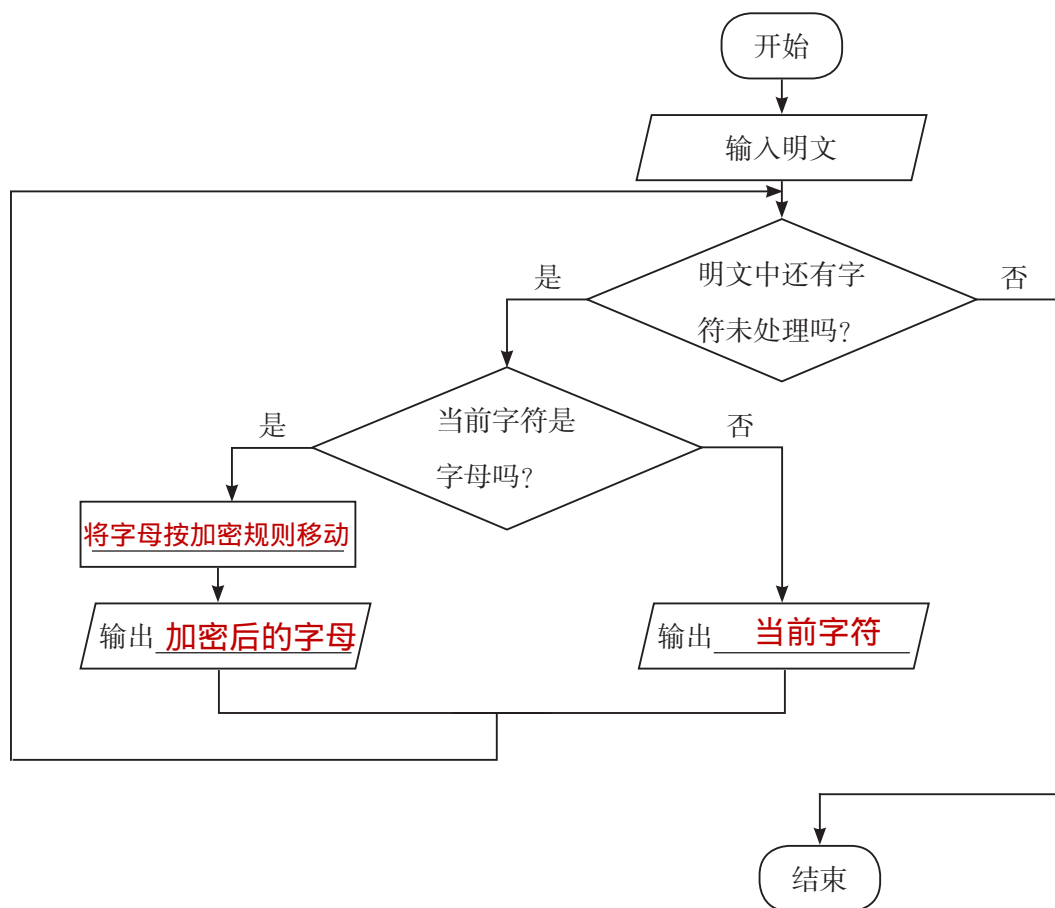
试一试

在移位加密算法中，如果遇到非字母符号（如数字、标点、空格等），应保持原样输出。我们可以通过两个双分支结构完善这个算法，请根据完善后的自然语言描述补全流程图。

第1步：输入要传送的明文。

第2步：判断第一个字符是否为字母。如果不是字母，则直接输出当前字符；如果是字母，将字母移动加密后输出对应的字母。

第3步：重复第2步的操作，分别判断并输出第二个字符、第三个字符……直到处理完明文中的所有字符。





任务拓展

1. 壮壮一家七口计划在“十一”假期乘坐动车去桂林游玩。壮壮和妈妈一起做攻略的时候，发现动车票的价格和儿童的年龄有关。

相关规定：随同成年人旅客乘车的儿童，年满6周岁且未满14周岁的，应当购买儿童优惠票；年满14周岁的，应当购买全价票。每一名持票成年人旅客可免费携带一名未满6周岁且不单独占用席位的儿童乘车；超过一名时，应按超过人数购买儿童优惠票。儿童年龄按乘车日期计算。

壮壮一家出行的人员有：壮壮11岁，妹妹5岁，爸爸40岁，妈妈39岁，堂哥16岁，伯伯43岁，伯母41岁。

(1) 请根据相关规定，讨论其中包含了哪些条件判断。

(2) 根据出行人员的情况，帮壮壮一家确定购票方案。

(1) ①判断儿童是否年满14周岁（含14周岁），是则购买全价票，否则进入第二次判断。②判断儿童是否年满6周岁，是则购买儿童优惠票，否则进入第三次判断。③判断未满六周岁儿童的数量是否少于持票成年人的数量，是则未满六周岁儿童无需买票；否则按超过人数购买儿童优惠票。

(2) 购票方案为：购买5张全价票（4名成年人+堂哥）和1张儿童优惠票（壮壮），妹妹可免费随成年人乘车。

2. 小桂去超市买东西时，发现超市推出如下优惠活动：购买商品总价低于100元，没有折扣；总价大于等于100元且低于200元，打9折；总价大于等于200元，打8折。最后结账的时候，小桂一共付了162元。

(1) 请描述该购物场景中存在哪些条件判断，列出所有的分支情况，并绘制流程图。

(2) 根据流程图计算小桂购买商品的原总价可能是多少。

(1) ①判断商品总价是否低于100元，是则无折扣，否则进入第二次判断。

②判断商品总价是否低于200元，是则打9折，否则打8折。

(2) 根据流程图，以此执行以下路径：

①若商品总价低于100元，原价输出，但 $162\text{元} > 100\text{元}$ ，因此不属于该分支。

②若商品总价低于200元，打九折后为162元，可解出此时原价为180元，满足“ $100\text{元} \leq 180\text{元} < 200\text{元}$ ”，符合该分支条件。

③若商品总价高于200元，打八折后为162元，可解出此时原价为202.5元，满足“ $202.5\text{元} \geq 200\text{元}$ ”，符合该分支条件。

所以小桂购买商品的原总价可能是180元或202.5元。□□□□

【流程图见PPT】



任务四 闰年平年我知道（2）

壮壮团队知道“判断闰年与平年”的算法后，想用程序实现它。经过查阅资料，他们发现通常用“if……elif……else”语句实现多分支结构。



学一学

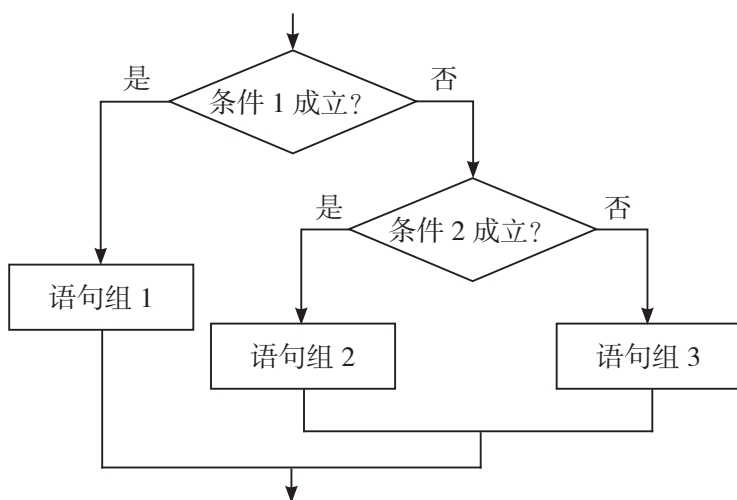
多分支结构是一种依据多个不同条件判断得到不同结果的程序结构。“判断闰年与平年”的算法有两个判断条件、两个双分支，这样的嵌套结构属于多分支结构。

下面是多分支结构的基本格式和算法流程图，请找出基本格式各语句在流程图对应的部分，并完成填空。

多分支结构基本格式

```
if 条件 1:  
    语句组 1  
elif 条件 2:  
    语句组 2  
else:  
    语句组 3
```

多分支结构算法流程图





在多分支结构的程序执行时，首先判断 条件1（选填“条件1”或“条件2”）是否成立，如果 条件1（选填“条件1”或“条件2”）成立，转向 是（选填“是”或“否”）分支，并执行 语句组1（选填“语句组1”、“语句组2”或“语句组3”）；如果不成立，则转向 否（选填“是”或“否”）分支。

接下来判断 条件2（选填“条件1”或“条件2”）是否成立，如果 条件2（选填“条件1”或“条件2”）成立，转向 是（选填“是”或“否”）分支，并执行 语句组2（选填“语句组1”、“语句组2”或“语句组3”）；如果不成立，则转向 否（选填“是”或“否”）分支，并执行 语句组3（选填“语句组1”、“语句组2”或“语句组3”）。



做一做

打开“超市优惠.py”程序，尝试输入不同区间的价格，观察输出提示，并判断执行的是哪一部分代码。

```
1  # 输入商品总价
2  total_price = float(input("请输入商品的总价："))
3
4  # 根据不同的价格区间计算实际付款金额
5~ if total_price < 100:
6      actual_payment = total_price
7~ elif 100 <= total_price < 200:
8      actual_payment = total_price * 0.9
9~ else:
10     actual_payment = total_price * 0.8
11
12 # 输出实际付款金额
13 print(f"实际需要支付的金额是：{actual_payment} 元")
14
```



续表

比较运算符	说明	举例
>	大于	print(100>4) 输出为 True
>=	大于等于	print(100>=4) 输出为 True
==	等于	print(100==4) 输出为 False
!=	不等于	print(100!=4) 输出为 True

3. 逻辑运算符：用于组合多个布尔表达式。

逻辑运算符	说明	举例
and	“与”运算符，表示要同时满足两个条件	print((100<4) and (4<100)) 输出为 False
or	“或”运算符，表示两个条件中只要满足其中一个即可	print((100<4) or (4<100)) 输出为 True
not	“非”运算符，用于对一个布尔值取反	print(not(100<4)) 输出为 True



练一练

先写出下列代码运行后的结果，再借助编程工具验证结果是否正确。

1. 填空题。

- (1) 计算表达式“10 // 4”的结果是 2。
- (2) 表达式“80 > 11”的结果是 True。
- (3) “(5 > 3) and (2 < 7)”的结果是 True。

2. 判断题，正确的在题后的括号中画“√”，错误的画“×”。

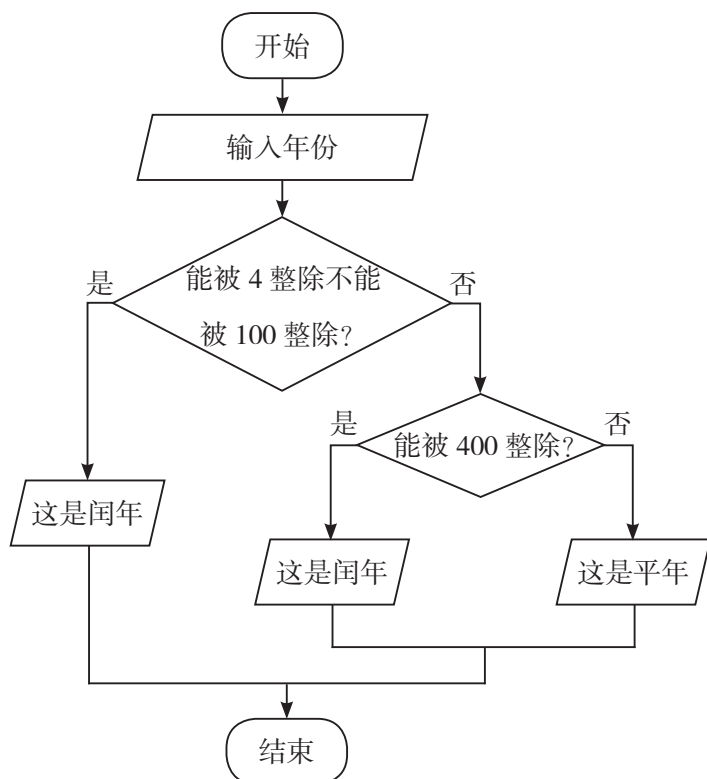
- (1) “7 % 3” 的结果是 1。 ()
- (2) “5 != 5” 的结果是 True。 (×)
- (3) “(4 < 8) and (9 > 1)” 的结果是 True。 ()





试一试

已知判断闰年与平年算法的流程图和部分代码如下：



#获取输入的年份

```
year = int(input('请输入一个年份: '))
```

判断是闰年还是平年

如果 `year % 4 == 0 and year % 100 != 0:`

```
print(year, ' 这是闰年 ')
```

又如果 `year % 400 == 0:`

```
print(year, ' 这是闰年 ')
```

否则

```
print(year, ' 这是平年 ')
```



1. 通过观察，找出判断条件对应的代码。

(1) 判断条件 1 “能被 4 整除不能被 100 整除” 对应的代码是_____

year %4==0 and year %100!=0:_____。

(2) 判断条件 2 “能被 400 整除” 对应的代码是_____

year % 400 == 0:_____。

2. 根据多分支结构程序语句，用程序代码替换红色文字的描述。

#获取输入的年份

year = int(input('请输入一个年份: '))

判断是闰年还是平年

if year % 4 == 0 and year % 100 != 0:

print(year, ' 这是闰年 ')

elif year % 400 == 0:

print(year, ' 这是闰年 ')

else :

print(year, ' 这是平年 ')

3. 打开“判断闰年与平年 – 补充前 .py” 程序，修改程序并保存。运行程序，输入不同的年份，观察执行结果。



任务拓展

1. 小桂为程序“小智”设计了一个能根据气温给出运动建议的功能，程序如下。



如果“小智”推荐的运动为游泳、晨跑或傍晚打球，为执行判断条件“`elif 20<=temperature<30`”后的语句组，所以当天温度可能在20-30度之间（含20度不含30度）。

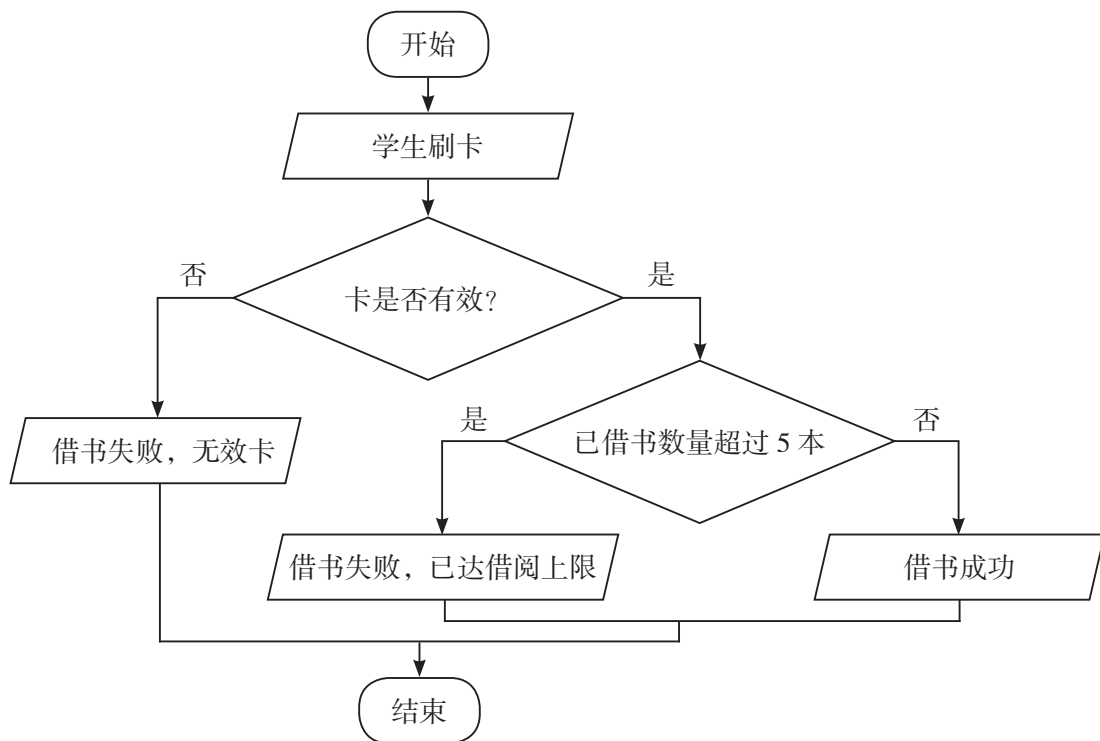
```

根据气温给出运动建议.py x
1 # 获取用户输入的温度
2 temperature = int(input("请输入当前温度(℃): "))
3
4 # 根据温度给出运动建议
5 if temperature < 0:
6     print(f"当前温度 {temperature}℃, 极寒天气! 建议选择室内运动, 如瑜伽、健身操或室内球类。")
7 elif 0 <= temperature < 10:
8     print(f"当前温度 {temperature}℃, 寒冷天气! 推荐慢跑、快走等低强度户外运动, 注意保暖防风。")
9 elif 10 <= temperature < 20:
10    print(f"当前温度 {temperature}℃, 凉爽舒适! 非常适合骑行、徒步、足球等户外运动, 享受新鲜空气。")
11 elif 20 <= temperature < 30:
12    print(f"当前温度 {temperature}℃, 温暖宜人! 推荐游泳、晨跑或傍晚打球, 注意补充水分。")
13 else:
14    print(f"当前温度 {temperature}℃, 高温预警! 建议选择室内游泳、健身房或傍晚瑜伽, 避免中暑。")

```

如果“小智”推荐的运动为游泳、晨跑或傍晚打球，那么当天温度可能是多少？打开“根据气温给出运动建议.py”，运行并检测自己的答案是否正确。

2. 查看“学生在图书馆使用自动借阅系统”的流程图，打开“图书馆自动借阅系统.py”，请把红色部分修改成代码，尝试保存并运行。





假设借书卡有效性

card = True

获取用户输入的已借书籍数量

count = int(input(" 请输入已经借出的书籍数量: "))

```
如果 not card:                                if
    print(" 借书失败, 无效卡 ")
elif 已借书的数量大于等于 5:                  count >= 5
    print(" 借书失败, 已达借阅上限 ")
否则 :                                         else
    print(" 借书成功 ")
```