

2024 年 7 月学考复习指南

信息技术

学考得 A 学考得 A

使用须知：

本材料中出现的内容均为学考的常考内容，基本覆盖学考考点的 75%。本资料分为两个部分：理论部分和代码部分。理论部分中大部分内容是要求掌握且能够熟练应用的，代码部分要求能够理解。本资料建议配合《A+学考清单》一起使用，所有知识点以 A+上所写的为准。

第一部分 理论部分

一. 概览

以下罗列的是学考的重点部分（考点对应 A+上的考点）：

必修一

第一章 数据与信息

考点一 数据、信息与知识

考点三 编码

考点四 数据管理与安全，大数据

第四章 数据处理与应用

考点十五 大数据的基本思想与框架

考点十六 利用 Pandas 模块处理数据

考点十七 利用 matplotlib 模块处理绘图

第五章 人工智能及其应用

考点二十 人工智能的产生、应用与发展

必修二

第二章 信息系统的支撑技术

考点二十八 硬件编程开发基础

考点三十一 网络应用软件开发与实践

第三章 信息系统安全

考点三十三 数据加密与安全（理论部分）

考点三十四 身份认证与安全（仅了解）

第四章 信息系统搭建实例

考点三十六 搭建信息系统的前期准备

二. 重点罗列

• 数据、信息、知识与智慧

数据：事实内容的记录，抽象内容，未经解释的数据没有意义。

信息：数据经过储存、分析及解释后所产生的意义。其具有载体依附性、时效性、共享性、可加工处理性、真伪性和价值性。

知识：对信息进行人为的归纳、演绎和比较，并与存在的人类知识体系相结合。知识可以被继承和传递。

智慧：收集，加工，应用，传播知识的能力，以及对事物发展的前瞻性看法，卓越的判断力。

• 进制与进制转换

1. 进制标识符

二进制	八进制	十进制	十六进制
B	O	D	H

表 1-1 进制标识符

2. 进制转换

(1) 十进制转其他进制——短除法：

以 13D 转为二进制为例（过程如图 1-2 所示）：

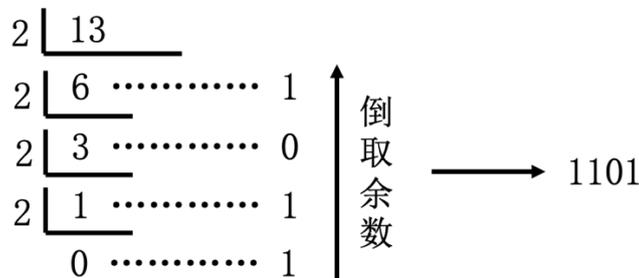


图 1-2 短除法

得到 13D=1101B

(2) 其他进制转十进制——位权相加：

以 1101B 转十进制为例（过程如图 1-3 所示）：

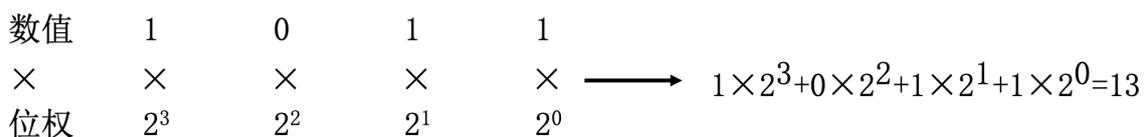


图 1-3 位权相加

得到 1101B=13D。

(3) 十六进制直接转二进制：

以 B2H 转二进制为例（过程如图 1-4 所示）：

首先，将十六进制每位拆开，将每一位转换为二进制，再按照十六进制的顺序把它们拼在一起。



图 1-4 十六进制直接转二进制

得到 B2H=11010010B。

(4) 二进制直接转十六进制：

以 101011 为例（过程如图 1-5 所示）：

首先将二进制以四个为一组从后往前（即从右往左分开）最后一组不足四位的在前面补 0，然后按照从前往后（即从左往右）的顺序依次转换为十六进制，再按照二进制顺序把它们拼在一起。

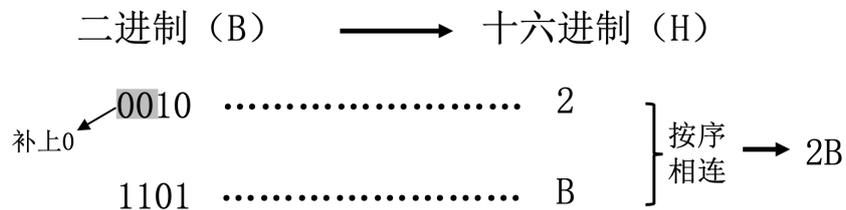


图 1-5 二进制直接转十六进制

得到 101101B=2BH。

• 编码

1. 英文或数字在计算机中使用 ASCII 码编码。ASCII 码共 128 个用一个字节 (Byte) 中的低 (前) 7 位 (bit) 编码，范围十进制 0-127，十六进制 00-7F。

2. 中文汉字在计算机中使用 GB2312 编码。一个汉字占两个字节。

无论以上哪种，最终都以二进制储存在电脑中。

3. 编码计算文件大小（仅无损文件可使用公式计算，括号中是单位）：

(1) 声音 (WAVE 为无损，MP3、WMA 为有损)

$$\text{采样频率 (Hz)} \times \text{量化位数 (bit)} \times \text{声道数 (个)} \times \text{时长 (s)} \quad (\text{单位: bit})$$

(2) 图像 (BMP 为无损，JPEG、PNG、GIF 为有损)

$$\text{横向像素 (个)} \times \text{纵向像素 (个)} \times \text{颜色位数深度 (bit)} \quad (\text{单位: bit})$$

颜色深度位数与颜色数转换公式：

$$\text{颜色深度位数} = \log_2 \text{颜色数 (种)} \quad (\text{单位: bit})$$

(3) 视频 (AVI 为无损，MP4 为有损)

$$\text{单帧容量 (计算公式见图像)} \times \text{帧速率 (帧/s)} \times \text{时长 (s)} \quad (\text{单位: bit})$$

4. 大小转换

8bit(位)=1B(字节); 1024B=1KB; 1024KB=1MB; 1024MB=1GB; 1024GB=1TB; 1024TB=1PB

• 大数据

1. 大数据的特点：数据体量巨大，速度快，数据类型多，价值密度低。

2. 大数据处理：处理时一般采用分治思想。

(1) 批处理计算：适用于静态数据，用 Hadoop 或 Spark 等处理，Hadoop 是一个可用于大规模计算机集群上的分布式系统基础架构。

(2) 流计算：适用于流数据，用 Storm 或 Heron 等处理

(3) 图计算（仅了解）：分析传染病传播路径等，适用于图数据（非图像数据）

• Pandas 数据处理（次次都考）

此部分见文件《Pandas 复习》，可在 <https://delt40.voin.ink/technology/pandas.pdf> 上获取。

• 使用 matplotlib 画图（次次都考）

以下是一些常用代码：

代码	含义
plot()	折线图
bar()	竖直柱状图
barh()	横向柱状图
scatter()	散点图
title()	设置标题
xlable();ylable()	设置 x 轴和 y 轴的标签

表 1-6 matplotlib 绘图函数常用代码

代码写法（以 plot() 为例）：

```
plt.plot(x,y) #此处 x 和 y 是两个列表或两个 Series 结构数据
```

• 人工智能

1. 人工智能主要方法：

(1) 符号主义：认为学习或其他智能特征能被符号描述，从而被机器仿真。例子：专家系统（记忆方法：去看专家要挂号，所以专家系统是符号主义）。

(2) 联结主义：模仿人脑中的神经元复杂交互来进行认知推理。例子：深度学习。

(3) 行为主义：智能体能在与环境的交互中不断学习，从而提升自己的智力水平。例子：试错学习（扫地机器人的试错学习）

2. 人工智能概念：

- (1) 领域人工智能：依赖领域知识和数据，具有强大的存储，记忆和搜索功能。应用举例：“深蓝”、“沃森”。
- (2) 跨领域人工智能：从一个领域快速跨越到另一个领域。应用举例：“Alpha Go”、“沃森”
- (3) 混合增强智能：人和机器一起干活，人类智能回路的总开关。应用举例：“机器客服与人类客服合作”。

- 硬件编程开发（最近不常考）

见《A+》第 68 页-第 71 页

- 网络应用软件开发

有 Web 服务器就是 B/S。

以下是一段代码实例：

```
import Flask
app=Flask(__name__)
@app.route("/")
def index():
    # 此处代码略
@app.route("/input",methods=['GET'])
def input():
    idm=request.args.get('id')
    cmd=request.args.get('cmd')
    # 此处代码略
if __name__=="__main__":
    app.run(host="192.168.0.16",port="8080")
```

分析以下这段代码，首先看到最下面的 app.run() 括号内写的内容，host 代表 IP 地址，而 port 代表端口号。本代码中 IP 地址为 192.168.0.16，端口号为 8080。加下来上面两处@app.route() 是路由，每个路由对应一个视图函数，本代码中的对应关系如表 1-7 所示。

路由	视图函数
@app.route("/")	def index():
@app.route("/input",methods=['GET'])	def input():

表 1-7 路由与视图函数的对应关系

可以看到视图函数 input() 的路由除了路由符号外，还有一个 methods 参数，这个参数代表了获取信息的方式。网页获取信息的方式有两种：GET 和 POST。GET 获取方式是从网址（URL）中直接获取

数据。例如当我输入网址 (URL): `http://192.168.0.16:8080/input?id=001&cmd='clear'` (加粗下划线处可修改) 时视图函数 `input()` 中的 `idm` 就会被赋值为 `001`, `cmd` 会被赋值为 `"clear"`。另一种方式时 `POST`, 是通过浏览器后台来传输数据, 这种数据不会直接显示在网址 (URL) 中, 所以 `POST` 传输适用于传输保密信息, 例如你的帐号和密码等 [本质区别一个 URL 中有? _____ (这种是 `GET`), 一个 URL 中没有? _____ (这种是 `POST`)]。

网址栏的书写非常简单, 分为以下几个部分:

`http://192.168.0.16:8080/input?id=001&cmd='clear'`
IP地址 路由 请求信息
(没有问号) (可能没有)

图 1-8 URL 的组成

其中路由部分就是在代码中路由的部分后面的参数, 请求信息的有无需要看代码中是否有 `request.args.get()` 之类的话。

数据库代码也是非常重要的内容以下是一些常用操作 (见表 1-10):

语句	说明
Python 中的操作	
<code>sqlite3.connect(database)</code>	连接到一个数据库, 其中 <code>database</code> 处应写为数据库名称。
<code>connection.execute(sql, 命令)</code>	执行 SQL 查询
sqlite3 中的常用查询命令	
<code>SELECT * FROM 表名 WHERE 筛选条件</code>	查询表名中的所有符合筛选条件的数据
<code>DELETE FROM 表名 WHERE 筛选条件</code>	删除表中的所有符合筛选条件的数据
<code>INSERT INTO 表名(要插入的条目) VALUES (条目对应数据)</code>	插入数据

表 1-10 SQL 常用操作

`sqlite3` 中的常用查询命令只需要知道表名在哪里, Python 常用操作中只需要知道数据库名称在哪里就行。

• 加密与解密 (理论部分)

P: 明文 C: 密文 K: 密钥

加密过程: $C = E_{K1}(P)$

解密过程: $P = D_{K2}(C)$

异或加密: 符号 \oplus , 代码中写作 \wedge , $1 \oplus 1 = 0$, $0 \oplus 0 = 0$, $1 \oplus 0 = 1$, $0 \oplus 1 = 1$

• 身份认证与安全

1. 访问控制: 解决用户对数据访问权限的问题。

2. 认证方式（三个）：用户名+口令（常说的帐号密码）、生物特征识别、USB key。

• 搭建信息系统的前期准备

1. 需求分析

（1）目标期待

（2）功能需求：

①核心需求：必须有的功能

②拓展需求：现有条件可使信息系统更令人满意的需求

③创新需求：要用新技术新方法

（3）性能需求：储存容量，运行时间，传输速度，安全性

（4）资源、环境需求：

①硬件种类、型号

②什么软件编写

③用户的技术水平需达到何种程度

2. 可行性分析

（1）必要性分析：是否应该马上开始搭建信息系统

（2）可行性分析：主要从技术、经济和社会意义等方面考虑

①技术：现有技术是否可以实现

②经济：对搭建所需费用与效益的评估

③社会意义：考察社会因素对它的制约作用

可行性分析结束后，要用可行性报告形式写出

3. 开发模式选择：B/S 或 C/S

4. 概要设计

（1）模块结构设计

①模块化结构设计，把系统分为若干模块

②模块中无序，不一定是树形

③模块结构设计主要包括：划分系统模块，确定模块功能，决定模块间的调用关系，制定模块间的接口

（2）系统物理配置：硬件设备配置、应用软件选择以及通信网络的选择和设计

（3）在数据库管理系统的选择上，主要考虑数据库的性能，类型，管理系统的选择

5. 详细设计：

（1）输入设计：输入内容，设备与正确性

- (2) 输出设计：输出内容，设备与介质，格式
- (3) 人机介面：美观、友好、简洁、统一
- (4) 数据库设计
- (5) 代码设计
- (6) 安全设计

• 信息系统搭建实例



图 1-9 完整的信息系统

如图 1-9 所示的是一套信息系统大致所含的内容。前面提到过有 Web 服务器就是 B/S。一般如果网络线路一开始是好的，后来突然某个数据长时间不更新，但是网站可以正常登录，说明可能是 IoT 模块和 Wi-Fi 断开连接，或者智能终端和 IoT 断开连接，或者传感器和 IoT 断开连接。无论 IoT 模块坏没坏，智能终端连没连网，只要智能终端不坏且智能终端和执行器连接良好，那么只要传感器感受到的数据不正常，执行器还是可以正常浇水的。

第二部分 代码部分

代码部分中分为三类代码：必修一算法代码、必修一 Pandas 代码、必修二加密代码。以下提供一些示例代码以供参考。

一. 基础知识

在 Python 中编写代码需要知道一些基础知识。

1. Python 中的数据类型

类型（关键字）	含义
int	整型：存放整数，如 4, 5, 6。
float	实型：存放小数，如 1.2, 2.5, 3.0
str	字符串型：存放字符串，如"del40.voin.ink"
bool	布尔型：只有两个值 True 和 False

表 2-1 Python 中的数据类型

2. Python 中的运算符

符号	名称	含义
----	----	----

+	加	对于整型和实型：数相加，如 3+5 的结果是 8 对于字符串类型：字符串相连，如 "del" + "40" 的结果是 "del40"
-	减	对于整型和实型：数相减，如 5-2 的结果是 3
*	乘	对于整型和实型：数相乘，如 5*2 的结果是 10 对于字符串类型：字符串多个相连，如 "del"*2 的结果是 "deltdel"
/	除	对于整型和实型：数相除，如 10/2 的结果是 5.0（结果为实型）
//	整除	对于整型和实型：数相除取整，如 10//2 的结果是 5（结果为整型）
%	取余	对于整型和实型：数相除取余数，如 5%2 的结果是 1（结果为整型）
^	异或	见理论部分“加密与解密（理论部分）”
not	非	not True=False , not False=True
and	与	True and True=True , True and False=False , False and False=False
or	或	True and True=True , True and False=True , False and False=False
round(x, n)	四舍五入	将 x 进行四舍五入，保留 n 位小数

表 2-2 Python 中的运算符

*需要主义的是，最后三个运算符的优先级是 not, and, or，也就是说程序中应先计算 not 再计算 and 最后再计算 or。

3. Python 中的关系运算符

关系运算符	含义
A==B	判断 A、B 是否相等，相等返回 True，反之 False
A>B 或 A<B	判断 A 是否大于 B（或 A 小于 B），是返回 True，反之 False
A<=B 或 A<=B	判断 A 是否大于等于 B（或 A 小于等于 B），是返回 True，反之 False
A!=B	判断 A 是否不等于 B，是返回 True，反之 False
A in B	判断 A 是否在 B 这个序列中，是返回 True，反之 False

表 2-3 Python 中的关系运算符

接下来可以看一些简单的关键词。

关键词	含义
input()	输入，括号中的参数是输入时显示的提示词
int()	将数据强行转换为整型
str()	将数据强行转换为字符串型
print()	输出括号中的内容

if A: B elif C: D else: E	分支结构，如果满足 A 则运行 B，如果不满足 A 但是满足 C，则运行 D，如果 A 与 B 都不满足，则运行 E。其中可以没有 elif 也可以没有 else
while A: B	在满足 A 条件时，重复执行 B 代码段
for i in A: B	让 i 遍历（一个一个取得）序列 A（一边是列表或者一个 range）的每一个值的同时，i 每取一个值运行一遍 B
range(s, e, p)	s 代表起始值，可以取到，e 代表结束值，取不到，p 代表步长，意思几个数取一个。例如：range(0, 5, 1)=[0, 1, 2, 3, 4]，range(1, 6, 2)=[1, 3, 5]。当步长为 1 时可以缩写为 range(s, e)；当步长为 1 且从 0 开始可以缩写为 range(e)（只有这两种缩写方式）。如 range(0, 5, 1) 可以缩写为 range(5)
ord('A')	取得字符'A'的 ASCII 值
chr(65)	将十进制 65 转换为 ASCII 对应的字母

表 2-4 Python 常用关键词

同时 Python 中还有一种特殊的东西，叫做自定义函数，请看以下代码。

```
def ou(x):    #自定义函数
    if x%2==0:    #如果是偶数
        return True
    else:
        return False
for i in range(10):
    if ou(i):
        print(i)
```

这段代码中出现了自定义函数 ou(x)，其中包含两个内容：函数名 ou 和参数 x。其中函数名和参数都可以自定义。要注意的是参数中的变量的作用域（发挥作用区域）仅限于函数内。即 ou(x) 这一自定义函数中的 x 仅在函数内起作用，如果在最后写上一句 print(x)，则该代码会报错。

当主程序中出现 ou() 这一句话时，该自定义函数就会被运行，例如上面这一段程序可以看成是这

样（这两段代码等价，为了方便模拟返回值，这里使用 flag 来表示返回值）：

```
#等价代码
for i in range(10):
    if i%2==0:
        flag=True
    else:
        flag=False
    if flag:
        print(i)
```

可以发现，自定义函数当中的参数被替换为了 i，这就说明我们传入的参数在自定义函数中会自动变成参数对应位置的内容，例如现有这么一个自定义函数 def f(x, y, z)我在主程序中调用时是这样写的：f(1, 2, 3)那么在运行自定义函数 f 时 x=1, y=2, z=3。最后在运行完毕自定义函数后会有返回值（不是所有自定义函数都有返回值），这个返回值会直接返回到函数出现的地方。比如我们原始的代码中，如果 x 是偶数，那么返回 True，则 True 就会返回到 if 后面函数的位置（即此时的 if 判断会变为 if True），则就会执行 if 下面的代码。如果无法理解返回值，可以使用等价代码来理解。

在了解了这些东西后，我们就可以开始看代码了。

二. 必修一算法代码

例 1:

```
s=0
x=int(input())
while x>0:
    m=x%10
    if m%2==0:
        s=s+m
    x=x//10
print(s)
```

阅读代码的过程如下：先创建变量 s 并赋值 0，输入 x，进入循环当 x 大于 0。m=x%10 取得 x 的个位，if m%2==0:如果 x 的个位（即变量 m）是偶数，让 s 加上 m。最后除去 x 的个位让原来的十位变成个位。

所以本段代码的作用，是将 x 中每一位偶数相加求和然后输出。例如，当 x 输入为 2137348 时，2、4 和 8 这三位是偶数，所以 s=2+4+8=14。

在阅读代码的过程中，有几个点是非常重要的。首先是你要看懂每个变量的含义，部分变量的含义可以通过变量名看出，表 2-5 中展示了常见的变量名对应变量内容

变量名	常见含义
s/sumn	求和
r	余数/随机数
t/tmp/temp	临时变量
f/flag	布尔型变量，一般是一个标记
m/max	最大值
m/min	最小值

表 2-5 常见变量名对应变量内容

*此表中所展现的只是一般情况，并不是所有题目中的变量都是这个意思，在做题时要具体题目具体分析。

在本代码中，s 变量用于求和，m 变量用于取得个位数。

例 2:

```
i=1 ; k=1 ; c=0
while i<7:
    c=c+k*(2*i-1)
    k=-k
    i=i+2
print(c)
```

阅读代码的过程如下：声明变量 i, k, c 分别为 1, 1, 0，当 $i < 7$ 的时候，c 加上 $k \times (2 \times i - 1)$ ，让 k 取相反数，让 $i += 2$ 。这个代码咋一看没有什么突破点，所以只能一点点分析。i 从 1 到 6 (7 取不到) 步长为 2，可以取到 1, 3, 5 三个数，所以 $2 \times i - 1$ 的值会是 1, 5, 9，这三个数对应的 k 值分别是 1, -1, 1，所以相乘得到的值是 1, -5, 9，所以 c 最后的值是 $1 - 5 + 9 = 5$ 。

这两段代码中，例 1 是功能性代码，你可以通过代码看出这段代码的功能，这种代码的解题思路就是看出功能，然后脱离代码直接手算。例 2 这种代码是抽象型代码，没有具体的功能，这种只能一点点分析，不能求快。

三. 必修一 Pandas 代码

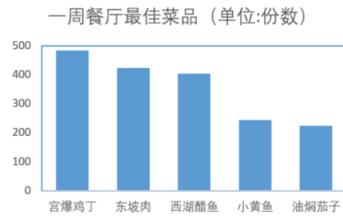
在阅读本部分前，请确认您已经阅读过《Pandas 复习》且已经能够熟练应用 Pandas 了。《Pandas 复习》这个文件可以在 <https://delt40.voin.ink/technology/pandas.pdf> 上获取。

例:

将系统数据库中一周订单数据导出为文件“week.xlsx”，如第 11 题图 b 所示，并制作图表如第 11 题图 c 所示

	A	B	C	D	E	F
1	日期	订单号	菜品	份数	单价	金额
2	2022-6-13	1	宫爆鸡丁	1	6.0	6.0
3	2022-6-13	1	西湖醋鱼	2	8.0	16.0
4	2022-6-13	1	青菜	1	3.5	3.5
5	2022-6-13	2	番茄蛋汤	1	2.5	2.5
6	2022-6-13	2	糖醋排骨	1	7.0	7.0
7	2022-6-13	2	东坡肉	3	9.0	27.0

第 11 题图 b



第 11 题图 c

```
import pandas as pd
import matplotlib.pyplot as plt
#图表显示中文字体,代码略
df = pd.read_excel("week.xlsx") #读取 Excel 文件中的数据
df1 = df.groupby("菜品", as_index = False).sum() #按“菜品”分组求和
df2 = df1.sort_values("份数", ascending = False).head(5) #按“份数”排序并取前 5 行
x = df2["菜品"] # 本题挖空的地方
y = df2["份数"]
plt.title("一周餐厅最佳菜品(单位:份数)") #设置图表标题
plt.bar(x, y) #绘制柱形图
plt.show()
```

阅读代码的过程：首先它读入了 week.xlsx 这个文件，然后按照“菜品”进行了分组并求和，然后将分组求和完的数据进行降序排序取前五，最后让图标的 x 轴是菜品名称，y 轴是菜品份数，并展示图表。

这种代码经常挖空的地方有以下几个：图表的 x 轴和 y 轴分别是什么，groupby 之后做的操作（求和.sum()、求平均.mean()、求个数.count()等），sort_values() 排序参数中 ascending 是啥，排序完之后的操作（取前五.head(5)、取后五.tail(5)（取前五或者后五的时候括号内可省略不写））这种题很简单，不要着急，仔细读题，一定可以做出来。

四. 必修二加密代码

根据加密方法的不同，题目难度不同，常见的加密方法是凯撒加密。

例：

```
p = "Tel-114"
c = ""
for ch in p:
```

```
if ch>= "0" and ch<= "9":
    c+=str(9-int(ch))
elif ch>= "a" and ch<="z":
    c+= chr(ord(ch)- ord("a")+ord("A"))
else:
    c+=ch
print(c)
```

阅读代码的过程：根据前面的理论知识（第一部分）不难知道，p 是明文，c 是密文，这里用了一个 for 循环让 ch 这个变量，遍历（一个一个取得）p 中的每一个字符。如果 ch 是数字，就让它变成 $9-\text{int}(\text{ch})$ ；如果 ch 是小写字母，就让它变成大写字母；如果既不是数字，也不是小写字母，那么不变。所以这道题 c 最后会变成：TEL-885。

做这种题的时候你需要有一点知识储备，例如 $\text{chr}(\text{ord}(\text{ch})-\text{ord}(\text{"a"})+\text{ord}(\text{"A"}))$ 这句话的意思是小写字母转大写字母， $\text{chr}((\text{ord}(\text{ch})-\text{ord}(\text{'a'})+\text{key})\%26+\text{ord}(\text{'a'}))$ 这是小写字母后移 key 位， $\text{chr}((\text{ord}(\text{ch})-\text{ord}(\text{'a'})-\text{key}+26)\%26+\text{ord}(\text{'a'}))$ 这是小写字母前移 key 位（如果要写大写字母前移或后移，把这段代码中的 $\text{ord}(\text{'a'})$ 中的 'a' 改为 'A' 即可）等。这需要做题来积累。

至于为什么在前移 key 位时要+26， $[\text{chr}((\text{ord}(\text{ch})-\text{ord}(\text{'a'})-\text{key}+26)\%26+\text{ord}(\text{'a'}))$ （语句中标粗的）]是因为如果 $\text{ch}=\text{'a'}$ ， $\text{key}=5$ ，那么减完 key 之后就是-5，为了防止是负的所以加上 26 来保证不是负的，为了消除加上 26 的影响所以要在最后%26。