信息技术重难点2--字符编码

一、编码概述

编码(Encoding)是信息按照某种规则或格式，从一种形式转换为另一种形式的过程。解码是编码的逆过程。
计算机对信息进行存储、加工、传递等处理，实际上是对信息的载体——数据进行处理。数据的表现形式可以是文本、图形、图像、声音、视频等，但不管是哪种形式的数据，最终存储在计算机中的都是经过一定规则编码后的二进制数字。

二、ASCII码

ASCII(American Standard Code for Information Interchange，美国信息交换标准代码)是一套基于拉丁字母的计算机编码系统，主要用于显示现代英语和其他西欧语言。它由电报码发展而来，是现今最通用的单字节编码系统。
基本的ASCIII码共有128个，用1个字节中的低7位编码。二进制范围为000000111111，即十六进制的00H~7FH。
基本的ASCIII码由33个控制字符、10个阿拉伯数字、26个英文大写字母、26个英文小写字母与些标点符号、运算符号组成。
ASCII码值及对应的字符，建议收藏。
常用的有空格为32(20H)，'0'为48(30H)，'A'为65(41H)，'a'为97(61H)，'A'与'a'相差32(20H)。
给定一个字符，想要获取其ASCII码怎么做呢？可以使用Python内置的ord函数，字符型->整型可以用ord函数。对应题目为TZOJ5885: ASCII表

ord()函数

> ord函数以一个字符（长度为1的字符串）作为参数，返回对应的 ASCII 数值，或者 Unicode 数值

TZOJ5885参考代码

```
a=input()
print(ord(a))
```

若知道ASCII值，能转换为字符吗？可以使用Python内置的chr函数，整型->字符型可以用chr函数。对应题目为TZOJ5889: 打印字符

chr函数

> chr() 用一个范围在256内的整数作参数，返回一个对应的字符。

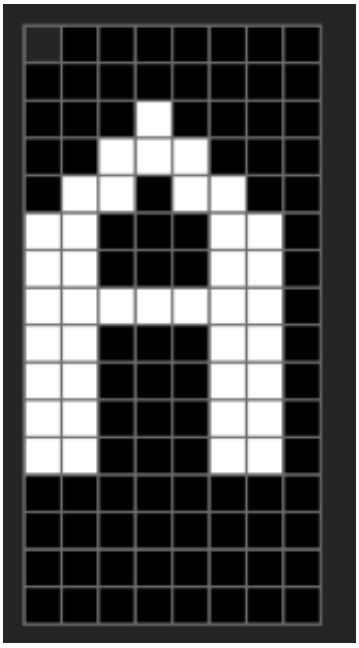
5889参考代码

```
# 注意输入的类型为int
a=int(input())
print(chr(a))
```

你可以尝试下以下题目

TZOJ7178 小z与ASCII加密、TZOJ6446 小z与ASCII解密、TZOJ1492 C语言实验题——大小写转换、TZOJ1489 C语言实验题——字符编码

要想显示出字母，系统中还必须装有相应的字库(Font，又称字型)。字库中存储了每个字符的形状信息，如字母'A'的点阵字符如下所示



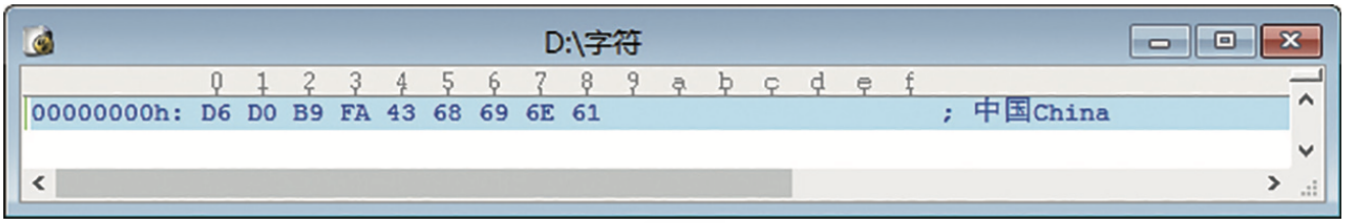
TZOJ6972: 点阵字符 可以尝试下。
字库分为点阵字符和矢量字符，矢量字符记录的是字符的笔画，具有存储空间小、美观的优点。

三、汉字编码

计算机中的汉字是采用二进制进行编码的。汉字编码分为外码、交换码、机内码和字形码。
外码：又称**输入**码，是用来将汉字输入到计算机中的一组键盘符号。常用的输入码有拼音码、五笔字形码等。
交换码：又称国标码，是和别的计算机**交换**信息时使用的编码。
机内码：根据国标码的规定，每一个汉字都有确定的二进制代码，在计算机内部汉字代码都用机内码，在磁盘上记录**存储**汉字代码也使用机内码。它们的集合就叫字符集。
字形码：又称输出码、汉字字模，把二进制的汉字编码还原为可视的图形，将其**显示**出来，它们的集合又叫字库。

四、GB2312编码

为了解决计算机的汉字显示问题，国家标准化管理委员会发布了GB/T 2312-1980，简称GB 2312。在GB 2312字符集中，1个汉字在计算机中用2个字节表示。如下图中的"中国China"这几个字符，其中"中国"两个汉字的内码为D6 DO B9 FA，而英文字符"China"是ASCII字符，其中每个字母都用1个字节编码表示。



图片中使用的软件是UltraEdit，使用Python可以直接将其内码依次显示。

```
a='中国China'
# 使用gb2312进行编码
a=a.encode('gb2312')
# 依次输出每个字节的值
for i in a:
    # 使用16进制输出
    print("%02X" % i,end=" ")
```

GB 2312标准共收录6763个汉字，其中一级汉字3755个，二级汉字3008个。GB 2312的出现，基本满足了汉字的计算机处理需要，它所收录的汉字已经覆盖中国大陆99.75%的使用频率。
使用GB 2312时采用EUC-CN(Extended Unix Code)储存方法，以便兼容于ASCII。ASCII取值范围00H~7FH，**两个连续的大于7FH的字符连接在一起就表示汉字**。
同时，GB 2312收录了包括一般符号、数字、希腊字母、日文假名、俄文字母等在内的682个字符。ASCII码内本来就有的标点、数字、字符等在GB 2312中依旧被编码了，我们把他们称为**全角字符**，ASCII内就叫做半角字符，他们在计算机显示是不一样的，全角字符往往要宽一点。

想要更进一步了解GB2312的区位码可以尝试下 TZOJ7334: GB2312区位码

五、* GBK编码

GB 2312中虽然已有6763个汉字，但是像有些生僻字"珽"以及繁体字"甦"等均无法显示。如果系统所用的是GB 2312，遇到这些字往往不能显示。还有1万多个汉字和需要编码，于是人们就想到了扩充，发布了《汉字内码扩展规范》，简称GBK(国标扩展)。GBK 向下与 GB 2312 编码兼容，向上支持 ISO 10646.1国际标准(国际标准组织ISO公布的通用字符集UCS)，，它与 Unicode 组织的 Unicode 编码完全兼容。**第一个字节还满足大于7FH**，第二个字节范围为40H到FEH。
使用Python查看其gbk内码时把上文代码中的encode设为gbk即可。

```
a='甦'
# 使用gbk进行编码
a=a.encode('gbk')
# 依次输出每个字节的值
for i in a:
    # 使用16进制输出
    print("%02X" % i,end=" ")
```

六、* GB18030编码

为了解决中文、国内少数民族文字、日文以及朝鲜语等的编码，国家标准化管理委员会又发布了GB 18030，现行标准为GB18030-2005，GB/T 2312-1980自2017年3月23日起转化为推荐性标准，不再强制执行。
GB 18030包含三种长度的编码：单字节的ASCII、**双字节的GBK（略带扩展）**、以及用于填补所有Unicode码位的四字节UTF区块。

七、Unicode字符集

随着互联网的快速发展，各国之间的交流由于语言不同成为障碍，能否设计出让650种语言都兼容的字符集呢，Unicode协会将所有这些语言字符统一为一套字符集解决了这个问题。
Unicode字符集又称万国码，往往用U+一个16进制数字表示，目前已经编码到10FFFFH，货币元¥为U+00A5，实心电话☎为U+260E，汉字烦为U+70E6，表情喜极而泣😂为U+1F602。

具体每个字符编码值 可以到https://unicode-table.com/cn查询

使用Python代码可以查到每个字符的Unicode符号值。

```
a='中国¥☎😂'
# 使用unicode-escape编码集
a=a.encode('unicode-escape')
print(a)
```

八、* UTF-8编码

Unicode字符集只规定了字符与码表的一一对应关系，却没有规定该如何实现，因此这个字符集有多种实现方式(UTF-8,UTF-18,UTF-32)，常见的实现方式是UTF-8。
UTF-8采用变长的编码方式实现，其节省空间，兼容ASCII标准的优点， 在互联网上使用最广的一种Unicode的实现方式。

UTF-8 的编码规则很简单，只有二条：
1.对于单字节的符号，字节的第一位设为0，后面7位为这个符号的 Unicode 码。因此对于英语字母，UTF-8 编码和 ASCII 码是相同的。
2.对于n字节的符号（n > 1），第一个字节的前n位都设为1，第n + 1位设为0，后面字节的前两位一律设为10。剩下的没有提及的二进制位，全部为这个符号的 Unicode 码。

Python代码依旧可以查出其utf-8编码值

```
a='烦'
# 使用utf-8进行编码
a=a.encode('utf-8')
# 依次输出每个字节的值
for i in a:
    # 使用16进制输出
    print("%02X" % i,end=" ")
```

汉字的长度从2字节变为3字节了，所以实际在网络传输中还是会使用其Unicode值进行传输的，而且我们也不知道用户使用什么编码，比如json中汉字为\u+四位十六进制数。

规定UTF-8编码规则能不能写出其转换程序呢？TZOJ7335: UTF-8编码等你来尝试。

九、Base64编码

Base64编码是计算机中常见的一种编码方式，规则是把3个字节(24位)的数据按6位1组分成4组(24÷6=4)，然后将每组数据分别转换为十进制，根据下图将这些十进制数所对应的字符连接，即为Base64编码。

表1.5.1　Base64编码表

索引	字符	索引	字符	索引	字符	索引	字符	索引	字符	索引	字符	索引	字符
0	A	10	K	20	U	30	e	40	o	50	y	60	8
1	B	11	L	21	V	31	f	41	p	51	z	61	9
2	C	12	M	22	W	32	g	42	q	52	0	62	+
3	D	13	N	23	X	33	h	43	r	53	1	63	/
4	E	14	O	24	Y	34	i	44	s	54	2		
5	F	15	P	25	Z	35	j	45	t	55	3		
6	G	16	Q	26	a	36	k	46	u	56	4		
7	H	17	R	27	b	37	l	47	v	57	5		
8	I	18	S	28	c	38	m	48	w	58	6		
9	J	19	T	29	d	39	n	49	x	59	7		

以编码字符“Web”为例，如下图所示，字符“Web”对应的ASCI编码分别是87，101，98，分别转换为8位二进制数，按6位二进制数分组后再转换成十进制，查找它们的对应字符，得到“Web”的Base64编码为“V2Vi”。

表1.5.2　Base64编码方法

文本	W								e								b							
ASCII编码	87								101								98							
二进制位	0	1	0	1	0	1	1	1	0	1	1	0	0	1	0	1	0	1	1	0	0	0	1	0
索引	21				54				21				34											
Base64编码	V				2				V				i											

TZOJ7212: Base64编码简单版 给你一个长度是3的倍数的字符串，请你找出其Base64编码的值。
这个过程实际上就是进制转换的过程，可以写出如下代码

7212参考代码1

```
# base64表
base64_table="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+-"
```

```
a=input()
# 3位一处理
for i in range(0,len(a),3):
    # 连续的3位按照256进制转换为数字
    x=ord(a[i])*256*256+ord(a[i+1])*256+ord(a[i+2])
    # 按照64进制转换回去
    print(base64_table[x//64//64//64],
    base64_table[x//64//64%64],
    base64_table[x//64%64],
    base64_table[x%64],
    end="",sep="")
```

也可以使用Python内置的base64模块

7212参考代码

```
from base64 import b64encode
data = input()
print(b64encode(data.encode()).decode())
```

如果原始的字节数不是3的倍数，比如4字节（剩余1字节）或者5字节（剩余2字节），则需要再末尾处理，一共可以分为2种情况。

1.如果剩余1字节，需要补2字节（16bit）才是3的倍数。也就是要则补16个0，加上前面剩余的1字节（8bit），一共是24个bit，其中24个bit的前12bit（26）*构成两个base64字符，剩下的12bit（26）*全部是0，用两个等号 == 表示。

2.如果剩余2字节，需要补1字节（8bit）才是3的倍数。也就是要则补8个0，加上前面剩余的2字节（16bit），一共是24个bit，其中24个bit的前18bit（36）*构成三个base64字符，剩下的6bit（16）*全部是0，用一个等号 = 表示。

了解了以上规则，TZOJ 7214: Base64编码 等你来AC。

编码会了，解码要不要尝试一下呢？TZOJ7213 Base64编码解码简单版、TZOJ7215 Base64编码解码

十、小结

计算机表示数字数字非常方便，但是遇到字符却不太好表示，发明计算机的美国人使用ASCII码表示英文以及控制字符等。等到中国人要用时，并没有我们的汉字，国家标准化管理委员会发布了GB 2312，但是其囊括汉字有限，微软推出了GBK。国家标准化管理委员会又发布了GB 18030，完全兼容GBK。后来发布的均兼容之前的，所以只要是上述出现的编码，使用GB 18030就不会出现乱码了。

每个国家都可以自己制定一套编码自己使用，为了世界不同语言的可以交流，我们发明了Unicode字符集。Unicode为了包含所有文字，使用更多位的字节的表示。对于有些文字使用多字节是比较浪费的，比如数字1，在ASCII码中使用一个字节就能够表示，在Unicode中有可能就是多字节表示，非常浪费存储。UTF-8是Unicode的一种实现方式。实际情况使用变字节的来表示文字。使用1~4个字节表示一个字符，根据不同的字符而变化字节长度，当字符在ASCII 码的范围时，就用一个字节表示，而一个中文字符占3个字节。这个和我们中文GB的编码是不同的，所以会出现乱码。不过UTF-8编码也兼容ASCII码，所以里面的英文是可以正常显示的。

参考：

1. GB2312、GBK、GB18030 这几种字符集的主要区别是什么？https://www.zhihu.com/question/19677619

2. 标准号：GB/T 2312-1980 中文标准名称：信息交换用汉字编码字符集 基本集 http://www.gb688.cn/bzgk/gb/newGbInfo?hcno=5664A728BD9D523DE3B99BC37AC7A2CC
3. 标准号：GB 18030-2005 中文标准名称：信息技术 中文编码字符集 http://www.gb688.cn/bzgk/gb/newGbInfo?hcno=C344D8D120B341A8DD328954A9B27A99