浙江省普通高中

Python 程序设计教学基础实例

浙江省浦江中学 方春林

目　录

第一讲　取数问题 …… 3
第二讲　最值问题 …… 5
第三讲　累加专题 …… 7
数字累加
字符串累加
第四讲　秦九韶算法 …… 12
第五讲　对称数（回文） …… 14
第六讲　进制转换专题 …… 17
十进制数转二进制数
二进制数转十进制数
二进制与十六进制相互转换
第七讲　字符串专题 …… 23
字符串基本操作
字符串中取数字
字符串中取单词
字符串中统计
字符串加密
第八讲　“数”问题 …… 37
质数
最大公约数
斐波那契数列
第九讲　约瑟夫问题 …… 42

第一讲　取数问题

中国象棋是中国的一种二人对抗性游戏的棋戏，可以修心养性，陶冶情操，丰富文化生活，深受广大群众的喜爱。玩象棋时，必须先确定动哪一个“棋子”，然后将“棋子”移动到下一步；数学上计算两数之和，要先确定是哪 2 个数做加法运算；在 windows 中要删除某个文件，需要先选择要删除的文件，然后再执行删除操作；……我们做任何事，必须先弄清对象，然后再执行相应的操作。

编写程序，就是将任务用计算机能看懂的语言一行行地写好，然后交给计算机一步一步执行。计算机执行指令，也需要先确定要操作的对象，比如做加法指令，必须确定要操作的两个数据。而很多时候，需要指定数据中获取需要的数据——这就是取数问题。

根据目标数据类型不同，取数问题可以分为整数型中取数问题和字符串中取数问题。下面分这两种类型分别讲解如何取数。

1.整数型

对于整数型数据，利用**整除**和**取余**方法获取各位上的数字

① 已知一个两位数 x，写出该数的个位数 a 及十位数 b 的 python 表达式

```
a = x % 10        #  x 除以 10 的余数就是个位数
b = x // 10       #  整除 10 得到十位数
```

② 写出三位数 x 的个位数 a，十位数 b，百位数 c 的 python 表达式

```
a = x % 10
b = x // 10 % 10 (或者  b = x % 100 // 10)
c = x // 100
```

2. 字符串型

字符串取子串用切片；s[start:end:step]表示从索引 start 开始取，直到索引 end 为止，但不包括索引 end，步长 step。举例如下：

```
s = "abcdefghabcd"
n = len(s)                  #字符串长度
first = s[0]                #字符串第1个字符串
last = s[-1]                #最后1个字符串
last = s[n - 1]             #最后1个字符串
strs = s[5]                 #第5个字符串
strs = s[2:5]               #第3个开始到第5个字符串
strs = s[::2]               #取从头开始每隔2个字符串
```

练习

① 写出计算一个三位整数 x 的各位数之和的 python 表达式。

② 根据 18 位身份证号码 sfz，写出计算年龄的 python 表达式。

③ 写出以一个三位数 x 开头的对称数的 python 表达式，如 x=123,则输出 123321。

④ 写出以 n 位字符串 s 开头的长度 2n-1 位的对称字符串的 python 表达式,如 s="abcdef",则输出"abcdefedcba"。

第二讲　最值问题

最值指在一个数据范围中的最大值或最小值。比如 2500 名学生成绩数据中找出成绩最好的学生；在一段字符串中找出最大的字符等等。

1. 求最值一般的思路

① 假定第一个数据最大并赋值给 Maxx；

② 取出下一个数与最大值 Maxx 比较，若大于最大值 Maxx，则更新最大值 Maxx，直到枚举完所有数据；

2. 求最值程序模板

```
Maxx = List[0]                          #List 为数据集合
for item in List[1:]:                   #第 2 个元素开始枚举数据
    if item > Maxx:                     #取出的数据大于 Maxx 的更新 Maxx
        Maxx = item
```

3. 实例讲解

输入一串字符串，输出该字符串中最大的字符。

思路：该问题是最值问题，可以直接应用求最值的程序模板，代码如下：

```
strs = input("输入一串字符串\n")   #strs字符串
maxchar = strs[0]                  #字符串strs的第1个字符赋值给maxchar
for ch in strs[1:]:                #第2个元素开始枚举字符串
    if ch > maxchar:               #取出的字符大于maxchar则更新maxchar
        maxchar = ch
print(maxchar)
```

4. python 方法求最值

python 中提供了函数 max(),min()分别求最大值,最小值。

max()函数原型

max(*iterable*, *[,*key*,*default*])

参数：*iterable*,表示可迭代对象，数据集合

函数功能：取传入的多个参数中（或可迭代对象元素中）的最大值。

min()函数原型

min(*iterable*,*[,*key*,*default*])

参数：*iterable*,表示可迭代对象，数据集合

函数功能：取传入的多个参数中（或可迭代对象元素中）的最小值。

举例如下：

```
>>> max(1, 2, 3, 4, 5, 6, 7, 8)
8
>>> max("zhejiangpujiangpujiangschool")
'z'
>>> max(False,True)
True
```

上述实例，可以利用 max()函数实现，代码为：maxchar = max(strs)

练习

① 输入批量数据，编程输出该批量数据中的最大值，最小值。

② 输入一个三位数，将该数各位上的数字重新排列得到新整数，求最大整数和最小整数。

③ 输入一个正整数,将该数各位上的数字重新排列得到新的整数,求最大整数和最小整数。

第三讲　累加专题

3.1　数值累加

《道德经》：“九层高台，起于累土。” 累：积累，叠加。加：增加。累加：在原有基础上添加。我们讲累加是指批量数据做加法运算。根据数据类型的不同，分数值累加和字符串累加两类问题进行讲解。本节介绍数值累加。

1. 问题引入

①计算 s=1+2+3 的和

我们可以按如下分步运算

s=0

s=s+1

s=s+2

s=s+3

经过以上 4 个步骤，可以得到 s 的结果。

②如果计算 s=1+2+3+…+100 呢，也可以分步完成

s=0

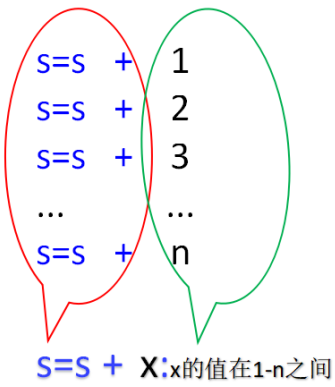
s=s+1

s=s+2

...

s=s+100

如何将上述计算过程转化为程序代码呢？编程也可以像数学上的提取公因数法，取出每个语句的公共部分，然后加上不同的部分来写代码。



上述的 100(n=100)条语句可以简化为一条语句 s=s+ x ,然后让 x 在[1,n]之间逐一枚举。要实现[1,n]之间逐一枚举，可以使用 for 语句和 range 对象实现。因此计算 s=1+2+3+...+100 的代码可以如下：

```
s = 0
for i in range(1,101):
    s += i
print(s)
```

2.数值累加模板

累加程序的代码可以用以下通用模板(写程序也可以用模板,简化写程序的过程)。

```
n = 101                 #累加元素的个数
s = 0                   #累加结果变量初始化为0
for i in range(1,n):    #枚举所有的数据范围
    s = s + X           #将待累加的每1个数据X累加到结果s中
print(s)                #输出结果s
```

在该模板中，程序将实现 n 个 X 的累加；X 是每一个需要累加的数据，可以用 python 表达式表示，累加程序关键是变量 X 如何表达。

3.python 方法实现累加

python 使用函数 sum()函数累加。该函数的原型如下 sum(iterable[,start])

参数说明：

iterable -- 可迭代对象，如：列表、元组、集合。

start -- 指定相加的参数，如果没有设置这个值，默认为 0。

返回值：计算结果。

4.实例讲解

① 编程实现计算 s=1+1/2+1/3+…1/n 的值(n<=100)

本问题是 n 个数相加的问题，可以应用累加模板，计算范围是[1,n]，变量 X = 1 / i。因此该问题可以写成如下代码：

```
n = 100
s = 0                       #累加结果变量初始化为0
for i in range(1,n + 1):    #枚举所有的数据范围
    X = 1 / i               #待累加的数据X
    s += X                  #将待累加的每1个数据X累加到结果s中
print(s)                    #输出结果s
```

【pythonic】利用函数 sum()计算累加,构造好迭代对象即可。

s = sum([1 / k for k in range(1,n + 1)]) ，其中 [1 / k for k in range(1,n + 1)]为列表生成式，该表达式生成列表[1,1/2,1/3,..,1/n]。

② 编程实现 s=1+3+5+…+n (n<=100)

本问题是 n 个数相加的问题，可以应用累加模板，计算范围是[1,n]，变量 x 是范围中的奇数。代码如下：

```
n = 100
s = 0                          #累加结果变量初始化为0
for i in range(1,n,2):         #枚举所有的数据范围
    s += i                     #将待累加的每1个数据X累加到结果s中
print(s)                       #输出结果s
```

【pythonic】先生成迭代对象，再利用函数 sum()求和

[k for k in range(1,n,2)]该列表生成式生成[1,3,5,…]范围内的奇数。代码如下： s=sum([k for k in range(1,n,2)])

练习

① 编程计算 s=1+(1+2)+(1+2+3)+…+(1+2+3…+n) (n<=10000)。

② 编程计算 s=1-2+3-4+…+n(n<=10000)。

③ 编程实现 100 以内能被 3 整除数的和。

3.2 字符串累加

字符串累加**是指**批量字符串前后连接起来，生成新的字符串。

1. 问题引入

① 输入 5，屏幕上显示"12345"。

本题的结果是将数据 1,2,3,4,5 前后拼接得到，这是字符串的累加。按 1,2,3,4,5 的先后顺序将数据累加。最先出现的 1 在最前面，最后出现的 5 在最后面。这就像排队时，先来的人排在前面，后来的人排在后面。我把它称为“排队累加”。代码如下：

```
s = ""
for i in range(1,6):
    s = s + str(i)
print(s)
```

运行结果如下：

```
12345
```

② 输入 5，屏幕是上显示”54321”。

本题也是将数据 1,2,3,4,5 拼接得到的结果，拼接的时候是最先出现的 1 在最后面，最后出现的 5 在最前面。这就像排队时，后来的人插队，排在了前面。我把它称为“插队累加”。代码如下：

```
s = ""
for i in range(1,6):
    s = str(i) + s
print(s)
```

运行结果如下：

```
54321
```

2. 字符串累加模板

“排队累加”中看出变量 i 从 1 递增到 5，程序每次将 i 累加到字符串 s 的后面。程序模板如下：

```
s = ""
for i in range(1,n):
    s = s + x
```

x 是待累加的字符串数据，s 是累加的结果。

“插队累加”中看出变量 i 从 1 递增到 5，程序每次将 i 累加到字符串 s 的前面。程序模板如下：

```
s = ""
for i in range(1,n):
    s = x + s
```

x 是待累加的字符串数据，s 是累加的结果。

3. 实例讲解

输入数据 5，在屏幕上打印如下的数字图形。

<table>
<tr><td>1
1 2
1 2 3
1 2 3 4
1 2 3 4 5</td><td>1
2 1
3 2 1
4 3 2 1
5 4 3 2 1</td></tr>
<tr><td>排队累加</td><td>插队累加</td></tr>
<tr><td colspan="2">从图上看出输出共 5 行，第 i 行打印的数据是第 i-1 行的内容加上 i，因此这是字符串累加问题。再观察发现左边的是字符</td></tr>
</table>

串的排队累加，右边的是插入累加，代码如下：	

s = ""
for i in range(1,6):
    s = str(i) + s
    print(s)
	s = ""
	for i in range(1,6):
	    s = s + str(i)
	    print(s)

练习

① 十进制数转 BCD 码。5421BCD 码是一种采用四位二进制数表示一位十进制是的编码，其各位的权依次为 5，4,2,1，并要求大于等于 5 的十进制数所对应的编码最高位为 1（如十进制数 7 所对应的“5421BCD”码为 1010）。转换时，将十进制数从右往左按位转换成对应的“5421BCD”码，然后依次连接。(程序运行如右图)

```
请输入十进制数:
753
101010000011
```

② .输入 1 个整数，输出如下的图形。

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

第四讲　秦九韶算法

秦九韶算法是中国南宋时期的数学家秦九韶提出的一种多项式简化算法。一般地，一元 n 次多项式的求值需要经过(n+1)*n/2 次乘法和 n 次加法，而秦九韶算法只需要 n 次乘法和 n 次加法。

1. 数学描述

把一个 n 次多项式改写如下所示：

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

$$= a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

$$= (a_n x^{n-1} + a_{n-1} x^{n-2} + \cdots + a_2 x + a_1) x + a_0$$

$$= ((a_n x^{n-2} + a_{n-1} x^{n-3} + \cdots a_3 x + a_2) x + a_1) x + a_0$$

$$\vdots$$

$$= (\ldots((a_n x + a_{n-1}) x + a_{n-2}) x + \cdots + a_1) x + a_0$$

求多项式的值时，首先计算最内层括号内一次多项式的值，

$ans_0 = a_n$

$ans_1 = ans0 * x + a_{n-1}$

然后由内向外逐层计算一次多项式的值，即

$ans_2 = ans_1 * x + a_{n-i}$

……

$ans_n = ans_{n-1} * x + a_0$

这样，求 n 次多项式 f(x)的值就转化为求 n 个一次多项式的值。

2. 算法描述

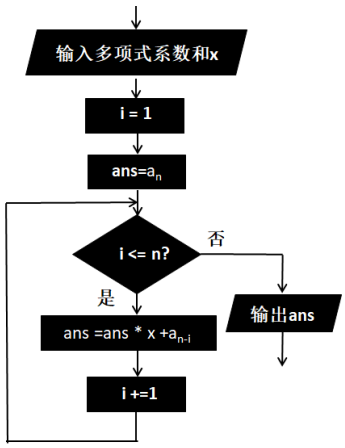
① 输入 f(x)的系数 a0, a1, …, an，输入 x 的值

② 答案 ans 初始化为 a_n

③ 对于 I =1, 2, 3…，n，循环执行 $ans=ans * x + a_{n-i}$

④ 输出 ans

3. 流程图描述



秦九韶算法流程图

4.程序代码

```
n = int(input("请输入多项式项数\n"))
xs = list(map(int,input().split()))
x = int(input("输入 x 的值"))
ans = xs[n]
for i in range(1,n + 1):
    ans = ans * x + xs[n - i]
print(ans)
```

xs 列表中存放多项式系数，多项式格式为：

f(x)=xs[n]*x**n+xs[n-1]*x**(n-1)+…+xs[1]*x+xs[0]

练习

① 输入一个正整数，计算各位数之和。

第五讲 对称数(回文)

对称数是指从左到右读和从右到左读是完全一样的数据。下面分数值类型和字符串类型两类进行描述。

1.对称数

1.1 对称数的判断思路及算法

判断某数是否为对称数，只需要生成该数的逆序数，然后判断这两者是否一致。生成逆序数的方法可以使用秦九韶算法。

如数据 x = 12321,生成它的逆序数的方法

1	2	3	2	1

逆序数结果保存在变量 rst 中，首先赋初值 0 即 rst = 0，接下来的步骤如下：

①取出倒数第 1 位 1，并计入结果变量 rst 中,rst = 10 * rst + 1

②取出倒数第 2 位 2，并计入结果变量 rst 中,rst = 10 * rst + 2

…

⑤取出倒数第 5 位 12，并计入结果变量 rst 中,rst = 10 * rst + 1

最后再判断 x 和 rst 是否相同，相同为对称数，否则不是。

1.2.对称数代码实现

```
x = int(input())
num = x
rst = 0
while x>0:
    rst = rst * 10 + x % 10
    x = x // 10
if num == rst:
    print("是对称数")
else:
    print("不是对称数")
```

1.3.生成对称数

从数字 1 开始连续递增生成指定长度（不超过 18 位）的对称数。如长度为 5，则生成对称数 12321，长度 6，则生成 123321。

思路：生成 n 位对称数，根据 n 的奇数偶数分别来生成对称数。步骤如下：

①先生成前半部分，从 1 开始循环到 n // 2 结束，利用秦九韶算法生成整数并存储在 rst；

②若 n 为奇数，则将中间数(n+1) // 2，添加到 rst 尾部；

③再生成后半部分，从 n//2 开始到 1 结束，利用秦九韶算法生成整数并存储在 rst；

1.4 生成对称数代码

方法 1：对称数分前后 2 部分分别生成

```
numlen = int(input())
rst = 0
for i in range(1,numlen // 2 + 1):
    rst = rst * 10 + i
if numlen % 2:
    rst = rst * 10 + (numlen + 1) // 2
for i in range(numlen // 2,0,-1):
    rst = rst * 10 + i
print(rst)
```

方法 2：1 个循环生成对称数(n 为正奇数)

分析：奇数个数的对称数的特点，如对称数 123454321，中间数 mid=5。

1	2	3	4	5	4	3	2	1

第 1 个数 x=1，与 mid 相差 y=4

第 2 个是 x=2，与 mid 相差 y=3

第 3 个是 x=3，与 mid 相差 y=2

…

第 7 个数 x=3，与 mid 相差 y=2

第 8 个数 x=2，与 mid 相差 y=3

第 9 个数 x=1，与 mid 相差 y=4

发现规律没有？

x 加 y(x 与 mid 的差值的绝对值)等于 1 个常量（mid 的值）。

代码如下：

```
rst = 0
mid = (numlen+1) // 2
for i in range(1,numlen + 1):
    rst = rst * 10 + mid - abs(mid - i)
print(rst)
```

2 对称字符串（或回文）

2.1 对称字符串的判断 2 个思路

思路 1：与对称数的判断方法一样，利用字符串累加方法生成原始字符串的逆序字符串，然后判断生成的字符串与原始字符串是否相同。（代码请同学们完成）

思路 2：假定字符串长度为 n，逐个判断第 1 个和倒数第 1 个是否相同，…，第 i 个与第 n-i-1 个是否相同，直到 n // 2 为止。若有一次取出的前后两个字符不相同，说明不是对称字符串，否则就是对称字符串。

2.2 思路 2 程序代码

```
strs = input()
nlen = len(strs)
flag = True
for i in range(nlen // 2):
    if strs[i] != strs[nlen - i -1]:
        flag = False
        break
if flag:
    print("是对称字符串")
else:
    print("不是对称字符串")
```

3 生成对称字符串

生成对称字符串与生成对称数类似，不同的数据类型发生改变，原来是数字，现在变成字符串。

3.1 生成以字符"a"开始长度 n(n<=52)的字符串。思路与生成对称数一样，有两种不同的方法。

思路 1 分前后两部分生成。

思路 2 利用字符串累加原理。（请自行完成代码）

思路 3 利用一个循环一次生成循环序列。（请自行完成代码）

思路 1 程序代码：

```
nlen = int(input())
rst = ""
firstch = "a"
for i in range(nlen // 2):
    rst = rst +chr(ord(firstch) + i)
if nlen % 2:
    rst = rst + chr(ord(firstch) + (nlen - 1) // 2)
for i in range(nlen // 2 - 1,-1,-1):
    rst = rst +chr(ord(firstch) + i)
print(rst)
```

练习

① 生成对称字符串的其它两个思路的程序实现。

② 编写程序生成左下图的菱形。

```
  *
 ***
*****
 ***
  *
```

第六讲　进制转换专题

计算机中所有数据都用二进制表示，而现实生活中更多的是十进制的数据。因此需要对不同的进制进行转换。

6.1　二进制数转十进制数

1. 二进制数转换为十进制数算法及程序实现

以二进制数”1011B”转换为十进制数为例讲解转换步骤。

方法 1：逐位按权值求和（从左往右）

1011B=1*2^3+0*2^2+1*2^1+1*2^0　①

思路：上述表达式①中可以看成是 4 个数字 x 相加，每个 x 有什么特征呢？x 是二进制数的某 1 位数字 * 相应的权值。该过程就是一个 4 个数据累加的过程，因此可以用累加语句模板来实现将二进制数转换为十进制数。

代码实现

```
binary = "1111"                 #二进制数
binarylen = len(binary)         #二进制数的长度
s=0
for i in range(binarylen):      #枚举二进制数的每一位
    a = int(binary[i])          #取出二进制数的每一位，转换为整型
    b = 2 ** (binarylen-i-1)    #计算权值
    x = a * b
    s += x                      #累加
print(s)
```

方法 2：逐位按权值求和（从右往左），这是取的方向不一样，结果是一致的。

1011B=1*2^0+1*2^1+0*2^2+1*2^3　②

思路同方法 1，只是数据的位置发生改变，代码如下：

```
binary = "1011"                         #二进制数
binarylen = len(binary)                 #二进制数的长度
s = 0
for i in range(binarylen):              #枚举二进制数的每一位
    a = int(binary[binarylen - i - 1]) #取出二进制数的每一位，转换为整型
    b = 2 ** i                          #计算权值
    x = a * b
    s += x                              #累加
print(s)
```

上述代码中，如果熟练了就没有必要变量 a,b 分步写，可以直接写一个表达式。这样写为了让学生初学时更好理解。

方法 3：运用秦九韶算法，该算法是中国南宋时期的数学家秦九韶提出的一种多项式简化算法。通俗的说十进制数 x，在其尾数加一位数 y，得到新的数 z，则 z=10*x+y；同理对于任何二进制数 x，其尾数加一位数 y，得到新的数 z，则 z=2*x+y。

将二进制数“1011B”转换为十进制数的分步过程如下表

1011B	二进制数每 1 位	初始值 s=0
第 1 次取的数	1	s=2*s+1
第 2 次取的数	0	s=2*s+0
第 3 次取的数	1	s=2*s+1
第 4 次取的数	1	s=2*s+1

经过以上 4 个步骤，s 的值就是该二进制数所对应的十进制数。怎么转换为代码呢，通过观察同样是累加问题。应用累加模板代码得到如下代码：

```
binary = "1011"                    #二进制数
binarylen = len(binary)            #二进制数的长度
s = 0
for i in range(binarylen):         #枚举二进制数的每一位
    x = int(binary[i])             #逐位取出二进制数
    s = 2 * s + x                  #累加
print(s)
```

以上是二进制数转换为十进制数的 3 种方法，1，2 两种方法其实是一样的，只是顺序不一样而已，第 3 种方法比较简单明了。其它进制转换为十进制数原理是一样的，只需要将相应的权值基数 2 改掉即可，如十六进制转为十进制数，将程序中 2 改成 16 即可。

2. pythonic

Python 实现上述问题，有 pythonic 的方式实现。现介绍 2 种方法：

①列表生成式

```
42 s=sum([int(x)*2**(len(bina)-k-1) for k,x in enumerate(bina)])
43 print(s)
```

②利用 lambda 表达式和 reduce()函数

```
7 from functools import reduce
8 data=[0,1,0,1,1]
9 a=reduce(lambda x,y:2*x+y,data)
```

③除此之外，python 利用 int()方法，将一个数字字符串转换十进制数。

函数 int()原型

class int(x, base=10)

参数

x -- 字符串或数字。

base -- 进制数，默认十进制。

返回值

返回整型数据

函数 int()一般用于将数值字符串转换为数字（十进制数），但是如果加上 base 参数，就可以将不同进制数转换为十进制数。如上述的二进制数”1011B”转换可以用下面的代码。

```
38 s=int('1011',2)
39 print(s)
```

练习

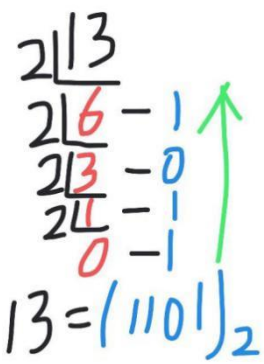
① 编程实现十六进制数转换为十进制数。

② **[浙 2018.4 选考 14]**某种编码以 4 位二进制码为一组，每组前两位表示方向，后两位表示距离。编写一个程序,将编码翻译成方向和距离，距离值为每组编码后两位二进制码转换为十进制数的值。（如右图）

```
输入编码字符串：
00000101
东 0
南 1
```

6.2 十进制数转二进制数

1.十进制数转二进制数的方法：辗转相除取余法

十进制数转换为二进制数的步骤 ①13 除以 2，得到商 6，余数 1； ②6 除以 2，得到商 3，余数 0； ③3 除以 2，得到商 1，余数 1； ④1 除以 2，得到商 0，余数 1； ⑤商为 0，结束。 依次将步骤④③②①得到的余数前后连接起来，结果是$(1101)_2$。 注意结果是余数的倒序排列。可以应用字符串的插队累加模板。	

2. 程序实现

思路：除 2 取余法，并利用字符串的插队累加得到最终结果。

```
num = int(input())
rst = ""
while num>0:
    r = num % 2
    num = num // 2
    rst =str(r) + rst
print(rst)
```

3. 十进制数转十六进制数

思路：与十进制数转二进制数一样，只需要将权值 2 改为 16；同时处理好余数大于 9 的数字转换为字母"ABCDEF"的问题。代码如下：

```
def dToh(num):
    rst = ""
    HTABLES = "0123456789ABCDEF"    #十六进制数元素
    while num > 0:
        r = num % 16                #取16的余数
        num = num // 16
        rst = HTABLES[r] + rst      #余数插队累加
    return rst
```

上述代码中取得的余数转换成十六进制数应用了一个对照表；也可以利用 chr()函数将大于 9 的数值转换为十六进制元素"ABCDEF"，代码如下：

```
def dToh2(num):
    rst = ""
    while num > 0:
        r = num % 16                #取16的余数
        num = num // 16
        if r > 9:
            rst = chr(r + 55) + rst      #余数插队累加
        else:
            rst = chr(r + 48) + rst
    return rst
```

练习

① [浙 17.4 学考 14]奇偶校验是一种校验数据传输正确性的方法。其中奇校验方法：统计二进制数据的数位中“1”的个数，若个数为奇数，则校验位值为 0，否则校验位值为 1。小李编写了一个计算奇校验位值程序，功能如下：输入 1～255 十进制待校验数，输出该数的二进制数及校验位值。

② 编写程序求一个负整数的反码。（提示：原码即将一个整数转换为二进制数，正数的反码等于原码，负数的反码等于原码逐位取反）

③ 求一个十进制负数(范围-127~-1)的 8 位二进制补码。

6.3 二进制与十六进制相互转换

1 十六进制数转换为二进制数

思路：将十六进制数中的每一位转换为四位二进制数并累加；本问题难点是将十六进制中的元素 A-Z 转化为相应的十进制数 10-15。代码如下：

```
def htob(hexnum):
    rst = ""
    for i in hexnum:                    #枚举十六进制数中的每个元素
        if i >='A' and i <='F':         #A-F字母的转化为10-15
            t = ord(i) - 55
        elif i>='0' and i<='9':         #数字的转化为0-9
            t = int(i)
        else:
            print('error')
            break
        rst += dtob(t)
    return rst
def dtob(num):                          #十进制数转二进制数
    rst =""
    for i in range(4):
        rst = str(num % 2) + rst
        num = num // 2
    return rst
```

2 二进制数转换为十六进制数

思路：将二进制数从右往左每隔 4 位分割，并将之转换为一位十六进制数，最后逐位按字符串插队累加。

```
def btoh(binary):
    rst = ""
    hextables = "0123456789ABCDEF"
    i = len(binary) - 1
    while i >= 0:
        if i>=3:
            t = binary[i-3:i+1]
        else:
            t = binary[:i + 1]
        h = btod(t)
        print(h,hextables[h])
        rst = hextables[h] + rst
        i -= 4
    return rst
def btod(binary):
    r = 0
    for i in binary:
        r = r * 2 + int(i)
    return r
```

程序将四位二进制数转换为十进制数自定义了一个函数 btod 来实现，也可以使用 int()函数来实现该问题。本实例是从右往左逐位取出并转换为十六进制数，也可以从左往右取出，并将之转换为十六进制数；代码请自行实现。

第七讲　字符串专题

7.1 字符串一般操作

字符串的一般操作有字符串的插入，删除，修改，查找等操作。

1. 查找

在字符串 source 中查找指定的关键字 key，若存在，则返回第 1 次出现该 key 的位置，否则返回-1。

思路：从字符串第 1 个位置开始，取出关键帧 key 长度的子串，并与关键字 key 比较，若相同则返回该位置，否则继续，直到字符串结束。

程序代码实现