

学习目标:

- 1 掌握3种数字类型的概念和使用方法;
- 2 了解3种数字类型在计算机中的表示方法;
- 3 运用Python的标准数学库进行数值计算;
- 4 掌握字符串类型的概念和使用方法;
- 5 掌握字符串类型的格式化操作方法和使用。



第三章 基本数据类型

3.1

数字类型

3.2

数字类型的操作

3.3

math库的使用

3.4

实例3 天天向上的力量

3.5

字符串类型及其操作

3.6

字符串类型的格式化

3.7

实例4 文本进度条

目

录

CONTENTS

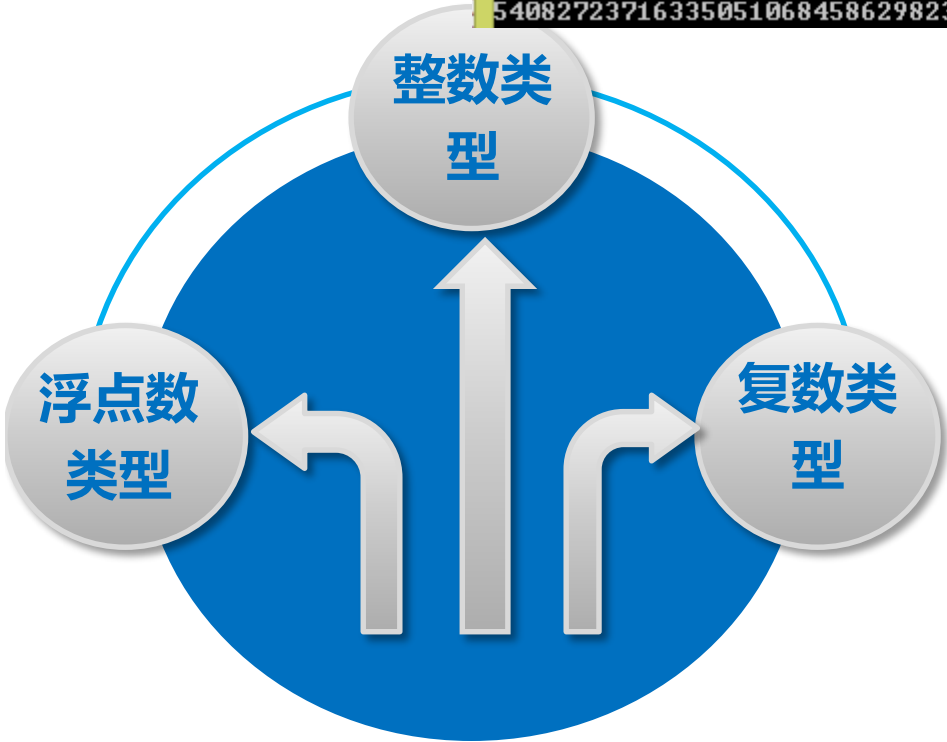
- ◆ 整数类型有4种进制：二、八、十、十六进制，二进制以0b开头，八进制以0o开头，十六进制以0x开头；
- ◆ 在Python中，一般认为整数类型没有取值范围，即 $[-\infty, \infty]$
- ◆ 函数pow(x,y)用于计算x的y次方

```
>>> pow(2,10)
1024
>>> pow(2,100)
1267650600228229401496703205376
>>> pow(2,pow(2,10))
17976931348623159077293051907890247336179769789423065727343008115773267580550096
31327084773224075360211201138798713933576587897688144166224928474306394741243777
67893424865485276302219601246094119453082952085005768838150682342462881473913110
540827237163350510684586298239947245938479716304835356329624224137216
```

Python中的浮点数采用两种表示方法：
十进制表示法和科学计数法

Python中的浮点数不受计算机系统的限制，sys.float_info详细列举了浮点数的各项参数：

- Max:最大值
- Min：最小值
- Max_10_exp:基数为10时最大幂值
- Min_10_exp:基数为10时最小幂值
- Radix:基数
- Max_exp:基数为2时最大幂值
- Min_exp:基数为2时最小幂值



复数可以看作二元有序对(a,b)，表示a+bj,a为实部，b为虚部。
如果z为复数，则z.real和z.imag分别为z的实部和虚部

内置的数值运算操作符：Python解释器提供9个基本的数值运算操作符，不需要引用标准或第三方函数库，又叫内置操作符

操作符	描述	操作符	描述
$x+y$	求和	$x\%y$	求余数
$x-y$	求差	$-x$	x 的负数
$x*y$	乘积	$+x$	x 本身
x/y	商	$x**y$	x 的 y 次幂
$x//y$	整数商		

- ◆ 3种数字类型之间存在扩展关系：整数 \rightarrow 浮点数 \rightarrow 复数
- ◆ 表中所列运算符与 $=$ 结合可构成复合赋值操作符，如 $x += y$ 等价于 $x=x+y$

内置的数值运算函数：Python解释器提供6个与数值运算有关的内置函数

函数	描述
<code>abs(x)</code>	x的绝对值
<code>divmod(x,y)</code>	输出为二元组, $(x//y, x\%y)$
<code>pow(x,y[,z])</code>	等同于 $(x**y)\%z$, z参数为可选参数
<code>round(x[,ndigitss])</code>	对x进行四舍五入, 保留ndigits位小数
<code>max(x1,x2,...,xn)</code>	最大值
<code>min(x1,x2,...,xn)</code>	最小值

内置的数字类型转换函数：Python的数字类型之间相互转换有两种方式，一是隐式转换，如5/2的结果为2.5（ $5//2=2$ ），二是通过类型转换函数转换。

函数	描述
int(x)	将x转换为整型数，x可以是浮点数或字符串
float(x)	将x转换为浮点数，x可以是整数或字符串
complex(re[,im])	生成一个复数，实部为re，虚部为im，re可以是整数、浮点数或字符串（ 如果第一个参数为字符串，则不需要指定第二个参数。 ），im可以是整数或浮点数但不能为字符串

复数不能直接转换为其它数字类型，可以通过`.real`和`.imag`将复数的实部或虚部分别转换。

Python数学计算的标准函数库math共提供了4个数学常数和44个函数

Python支持的库分为编辑器环境提供的默认函数库（称为**标准函数库**），另外一种是由第三方提供的**第三方库**，需要通过安装完成

- ◆ math只提供整数和浮点数的运算，不支持复数的运算；
- ◆ math提供4个常数、44个函数（包括16个数值表示函数、8个幂对数函数、16个三角对数函数和4个高等特殊函数）；
- ◆ 使用math库的函数，需要首先引入math库，引入方法有两种：
`import math`
`from math import <函数名>或 from math import *`

math库解析：数学常数

常数	数学表示	描述
math.pi	π	圆周率, 3.141 592 653 589 793
math.e	E	自然对数, 2.718 281 828 459 045
math.inf	∞	正无穷大, 负无穷大为-math.inf
math.nan		非浮点数标记

math库解析：数值表示函数

常数	描述	常数	描述
<code>math.fabs()</code>	绝对值	<code>math.ldexp(x,i)</code>	返回 $x*2^i$
<code>math.fmod(x,y)</code>	$x\%y$	<code>math.modf(x)</code>	返回x的小数部分和整数部分
<code>math.fsum(x,y,...)</code>	累加和	<code>math.trunc(x)</code>	返回x的整数部分
<code>math.ceil(x)</code>	取上整数	<code>math.copysign(x,y)</code>	用y的符号替换x的符号
<code>math.floor(x)</code>	取下整数	<code>math.isclose(a,b)</code>	比较a和b的相似性，返回True或False
<code>math.factorial(x)</code>	x的阶乘	<code>math.isfinite(x)</code>	当x无穷大时，返回True，否则返回False
<code>math.gcd(a,b)</code>	a,b的最大公约数	<code>math.isinf(x)</code>	当x为正数或负无穷大时，返回False，否则，返回True
<code>math.frexp(x)</code>	$x=m*2^e$ ，返回(m,e)	<code>math.isnan(x)</code>	当x是NaN，返回True，否则返回False

math库解析：幂对数函数

函数	数学表示	描述
<code>math.pow(x, y)</code>	x^y	x的y次幂
<code>math.exp(x)</code>	e^x	e的x次幂，e是自然对数
<code>math.expm1(x)</code>	$e^x - 1$	e的x次幂减1
<code>math.sqrt(x)</code>	\sqrt{x}	返回x的平方根
<code>math.log(x[, base])</code>	$\log_{base}x$	返回x的对数值，只输入x时，返回自然对数
<code>math.log1p(x)</code>	$\ln(1+x)$	返回1+x的自然对数值
<code>math.log2(x)</code>	$\log x$	返回x的2对数值
<code>math.log10(x)</code>	$\log_{10}x$	返回x的10对数值

math库解析：三角运算函数

函数	数学表示	描述
<code>math.degree(x)</code>		角度x的弧度值转角度值
<code>math.radians(x)</code>		角度x的角度值转弧度值
<code>math.hypot(x,y)</code>	$\sqrt{x^2 + y^2}$	返回(x,y)坐标到原点(0,0)的距离
<code>math.sin(x)</code>	$\sin x$	返回x的正弦函数值, x是弧度值
<code>math.cos(x)</code>	$\cos x$	返回x的余弦函数值, x是弧度值
<code>math.tan(x)</code>	$\tan x$	返回x的正切函数值, x是弧度值
<code>math.asin(x)</code>	$\arcsin x$	返回x的反正弦函数值, x是弧度值
<code>math.acos(x)</code>	$\arccos x$	返回x的反余弦函数值, x是弧度值
<code>math.atan(x)</code>	$\arctan x$	返回x的反正切函数值, x是弧度值
<code>math.atan2(y,x)</code>	$\arctan y/x$	返回y/x的反正切函数值, x是弧度值
<code>math.sinh(x)</code>	$\sinh x$	返回x的双曲正弦函数值
<code>math.cosh(x)</code>	$\cosh x$	返回x的双曲余弦函数值
<code>math.tanh(x)</code>	$\tanh x$	返回x的双曲正切函数值
<code>math.asinh(x)</code>	$\operatorname{arcsinh} x$	返回x的反双曲正弦函数值
<code>math.acosh(x)</code>	$\operatorname{arccosh} x$	返回x的反双曲余弦函数值
<code>math.atanh(x)</code>	$\operatorname{arctanh} x$	返回x的反双曲正切函数值

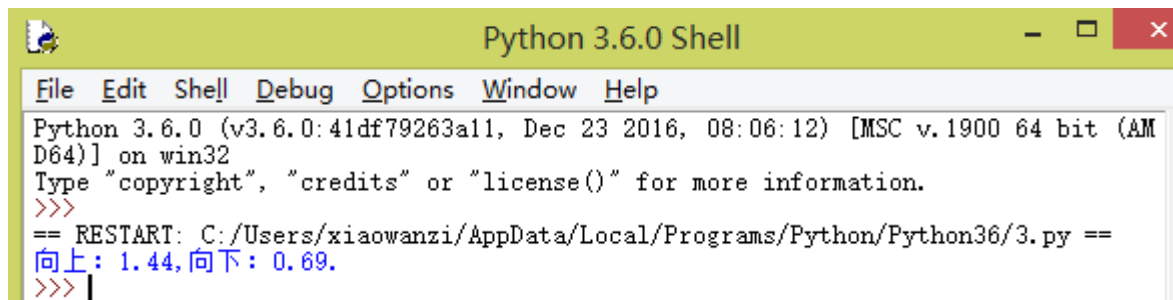
math库解析：高等特殊函数

函数	数学表示	描述
<code>math.erf(x)</code>	$\frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$	高斯误差函数，应用于概率论、统计学等领域
<code>math.erfc(x)</code>	$\frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt$	余补高斯误差函数， <code>math.erfc(x)=1 - math.erf(x)</code>
<code>math.gamma(x)</code>	$\int_0^{\infty} x^{t-1} e^{-x} dx$	伽玛（Gamma）函数，也叫欧拉第二积分函数
<code>math.lgamma(x)</code>	<code>ln(gamma(x))</code>	伽玛函数的自然对数

3.4 实例3 天天向上的力量

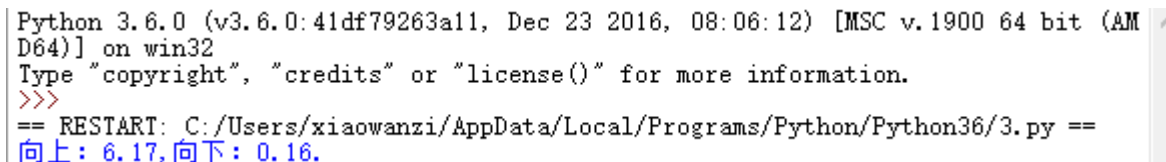
每年365天，以第一天的能力值为基数，假设为1.0，当好好学习时，每天的能力值提高0.1%，当不学习时，每天的能力值降低0.1%，每天努力学习和每天不学习时，一年后的能力值相差多少？

```
#e3.1daydayup365.py
import math
dayup=math.pow((1.0+0.001),365)
daydown=math.pow((1.0-0.001),365)
print("向上 : {:0.2f},向下 : {:0.2f}.".format(dayup,daydown))
```



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/xiaowanzi/AppData/Local/Programs/Python/Python36/3.py ==
向上 : 1.44, 向下 : 0.69.
>>> |
```

若改为每天提高或下降0.5%，结果为：



```
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 08:06:12) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/xiaowanzi/AppData/Local/Programs/Python/Python36/3.py ==
向上 : 6.17, 向下 : 0.16.
```

实验3.1：编写程序，分别计算 15° 、 30° 、 45° 、 60° 、 75° 、 90° 对应的弧度值。

实验3.2：一年365天，初始水平值为1.0，每工作一天水平增加N，不工作时水平不下降，一周连续工作5天，请编写程序运算结果并填写下表

N	0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009	0.01
年终值										

字符串类型的表示

字符串可以使用单引号、双引号、三引号表示，其中单引号、双引号可以表示单行的字符串，三引号可以表示多行的字符串，单引号表示的字符串中可以包含双引号字符，双引号表示的字符串中可以包含单引号字符，三引号表示的字符串可以包含单引号、双引号的字符

```
print('单引号表示可以使用"双引号"作为字符串的一部分')  
print("双引号表示可以使用'双引号'作为字符串的一部分")  
print("""三引号表示可以使用"双引号"  
'单引号'  
也可以换行""")
```

```
== RESTART: C:/Users/xiaowanzi/AppData/Local/Programs/Python/Python36/3.py ==  
单引号表示可以使用"双引号"作为字符串的一部分  
双引号表示可以使用'双引号'作为字符串的一部分  
三引号表示可以使用"双引号"  
'单引号'  
也可以换行
```

字符串类型的表示

print()函数可以原样输出字符串，input()函数可以接受从键盘输入的字符串，如：

```
>>> name=input("请输入名字：")
```

请输入名字：Python语言

```
>>> print(name)
```

Python语言

- 字符串的两种序号体系（**正向递增序号**和**反向递减序号**）均可以用于访问指定位置的字符；
- Python还提供了切片功能访问字符串的子字符串，采用[N:M]的形式；
- 由于字符串采用unicode标准编码存储，所以单个英文字符和中文汉字都算作一个字符。

如：

```
>>> name="Python语言程序设计"
```

```
>>> name[0]
```

P

```
>>> print(name[0],name[7],name[-1])
```

P 言 计

```
>>> print(name[2:-4])
```

thon语言

基本的字符串操作符

操作符	描述
x+y	连接两个字符串
x*n或n*x	复制n次字符串x
x in s	判断字符串x是否是字符串s的子字符串
str[i]	返回第i个字符，i可以是正向或反向的索引号
str[N:M]	切片，返回索引第N到M-1的子字符串

```
>>> "Python语言" + " 程序设计"
Python语言程序设计
>>> " Gool!" *3
Gool!Gool!Gool!
>>> ' Y' in 'Python'
False
```

基本的字符串操作符

微实例：

```
weekstr="星期一星期二星期三星期四星期五星期六星期日"
```

```
weekid=eval(input("请输入星期数字（1-7）："))
```

```
pos=(weekid-1)*3
```

```
print(weekstr[pos:pos+3])
```

```
请输入星期数字（1-7）：4
星期四
```

在print中，允许存在转义字符表示特殊的功能

字符	功能	字符	功能
\a	响铃	\r	回车，光标移到本行的开头
\b	向后退一格	\t	水平制表
\f	换页	\v	垂直制表
\n	换行，光标移到下行的开头	\0	NULL，发呆

内置的字符串处理函数

函数	描述
<code>len(x)</code>	返回字符串的长度，也可以返回其它组合类型的元素个数
<code>str(x)</code>	返回任意类型x所对应的字符串形式
<code>chr(x)</code>	返回unicode编码x对应的单字符
<code>ord(x)</code>	返回单字符表示的unicode编码
<code>hex(x)</code>	返回整数x对应的十六进制数的小写形式字符串
<code>oct(x)</code>	返回整数x对应八进制数的小写形式字符串

```
>>>str(3.1415926)
```

```
'3.1415926'
```

```
>>>"1+1=2"+chr(10004)
```

```
'1+1=2✓'
```

```
>>>"金牛座♉的unicode值是：" + str(ord("♉"))
```

```
'金牛座♉的unicode值是:9801'
```

```
>>>hex(255),oct(-255)
```

```
('0xff', '-0o377')
```

内置的字符串处理函数

凯撒密码：古罗马凯撒大帝用于对情报加密的算法，每个字符的对应关系如下：

ABCDEFGHIJKLMNOPQRSTUVWXYZ

DEFGHIJKLMNOPQRSTUVWXYZABC

即加密算法为： $C=(P+3) \% 26$

解密算法为： $P=(C-3)\%26$

其加密过程可以通过如下代码实现：

```
plaincode=input("请输入明文:")
for p in plaincode:
    if (ord("a")<=ord(p)<=ord("z")):
        print(chr(ord("a")+(ord(p)-ord("a")+3)%26),end="")
    else:
        print(p,end=' ')
```

运行结果如下：

请输入明文:python is an excellent language.

sbwkrq lv dq hafhoohqw odqjxdjh.

内置的字符串处理方法

在面向对象的方式中，所有的数据类型都封装为一个类，具有类似<a>.()形式的处理函数，这种方法在面向对象中成为方法，

方法	描述	方法	描述
str.lower()	字符小写	str.split(sep=None,maxsplit=-1)	由str根据sep被分割的部分组成
str.upper()	字符大写	str.count(sub[,start[,end]])	返回str[start,end]中sub子串出现的次数
str.islower()	判断是否都是小写	str.replace(old,new[,count])	所有old被new替代，count给出替换次数
str.isprintable()	是否是可打印字符	str.center(width[,fillchar])	字符串居中
str.isnumeric()	是否是数字字符	str.endswith(suffix[,start[,end]])	str[start:end]是否以suffix结尾
str.isspace()	是否是空格	str.startswith(prefix[,start[,ens]])	str[start:end]]是否以prefix开头
str.strip([chars])	去除字符串的左右chars子字符串	str.format()	字符串str的排版格式
str.zfill(width)	生成width长度的串，不足的在左侧补0	str.join(iterable)	由组合数据类型iterable变量的每个元素组成新字符串，元素间用str分隔

- ## 几个例子如下：

22

format()方法的基本使用

在进行输出时，存在固定输出内容和可变内容的连接问题，如：“今天是2018-01-29”
其中下划线为可变内容，Python中可以使用.format()方法进行格式化处理。

format()方法的基本格式为：

<模板字符串>.format(<逗号分隔的参数>)

模板字符串由一系列槽组成，槽用大括号{}表示，用来控制可变字符串中嵌入值出现的位置，槽的顺序与参数的顺序对应。如：

“{}：计算机{}的CPU占用率为{}%。” .format(“2018-1-29” ,” Python” ,10)



“{1}：计算机{0}的CPU占用率为{2}%。” .format(“2018-1-29” ,” Python” ,10)



format()方法的格式控制

format()方法中模板字符串的槽除了包含参数序号，还可以包括格式控制信息：
{<参数序号>:<格式控制标记>}
其中**格式控制标记**用来控制参数显示的格式，格式如下：

:	<填充>	<对齐>	<宽度>	<,>	<.精度>	<类型>
引导符号	用于填充的单个字符	<左对齐 >右对齐 ^居中	槽设定输出宽度	数字的千位分隔符，适用于整数和浮点数	浮点数的小数部分的精度或字符串的最大输出长度	整数类型： b,c,d,o,x,X 浮点数： e,E,f%

```
>>> s="Python"
>>> "{0:30}".format(s)
'Python'
>>> "{0:*^30}".format(s)
'*****Python*****'
>>> "{0:-^20,}".format(1234567890)
'---1,234,567,890---
```

```
>>> "{0:.2f}".format(12345.6789)
'12345.68'
>>> "{0:H^20.3f}".format(12345.67890)
'HHHHH12345.679HHHHH'
>>> "{0:.4}".format("Python")
'Pyth'
```


入门

```
#4.1textprogressbar.py
import time
scale=10
print("-----执行开始-----")
for i in range(scale+1):
    a,b='**'*i,'..'*(scale-i)
    c=(i/scale)*100
    print(" {:^3.0f}%[{}->{}]".format(c,a,b))
    time.sleep(0.1)
print("-----执行结束-----")
```

```
-----执行开始-----
0 %[->.....]
10 %[*->.....]
20 %[***->.....]
30 %[*****->.....]
40 %[*****->.....]
50 %[*****->.....]
60 %[*****->.....]
70 %[*****->.....]
80 %[*****->....]
90 %[*****->..]
100%[*****->]
-----执行结束-----
```

单行动态刷新

#4.2textprogressbar.py

import time

for i in range(101):

 print("\r{:2}%".format(i),end="")

 time.sleep(0.05)

IDLE中的输出结果:

```
0% 1% 2% 3% 4% 5% 6% 7% 8% 9% 10% 11% 12% 13% 14% 15% 16% 17% 18% 19%
20% 21% 22% 23% 24% 25% 26% 27% 28% 29% 30% 31% 32% 33% 34% 35% 36% 37% 38% 39%
40% 41% 42% 43% 44% 45% 46% 47% 48% 49% 50% 51% 52% 53% 54% 55% 56% 57% 58% 59%
60% 61% 62% 63% 64% 65% 66% 67% 68% 69% 70% 71% 72% 73% 74% 75% 76% 77% 78% 79%
80% 81% 82% 83% 84% 85% 86% 87% 88% 89% 90% 91% 92% 93% 94% 95% 96% 97% 98% 99%
100%
```

3.7 实例4 文本进度条

带刷新的文本进度条

#4.3textprogressbar.py

```
import time
```

```
scale=50
```

```
print("-----执行开始-----".center(scale//2,'-'))
```

```
t=time.clock()
```

```
for i in range(scale+1):
```

```
    a='*'*i
```

```
    b='.'*(scale-i)
```

```
    c=(i/scale)*100
```

```
    t=time.clock()
```

```
    print("\r{:^3.0f}%[{}->{}]{:.2f}s".format(c,a,b,-t),end=' ')
```

```
    time.sleep(0.05)
```

```
print("\n" + "执行结束".center(scale//2,'-'))
```

```
-----执行开始-----
0 %[->.....]0.00s  2 %[*->.....]0.08s  4 %[*->.....]0.16s  6 %[*->.....]0.24s  8 %[*->.....]0.32s 10 %[*->.....]0.40s 12 %[*->.....]0.48s 14 %[*->.....]0.56s 16 %[*->.....]0.64s 18 %[*->.....]0.72s 20 %[*->.....]0.80s 22 %[*->.....]0.88s 24 %[*->.....]0.96s 26 %[*->.....]1.04s 28 %[*->.....]1.12s 30 %[*->.....]1.20s 32 %[*->.....]1.28s 34 %[*->.....]1.36s 36 %[*->.....]1.44s 38 %[*->.....]1.52s 40 %[*->.....]1.60s 42 %[*->.....]1.68s 44 %[*->.....]1.76s 46 %[*->.....]1.84s 48 %[*->.....]1.92s 50 %[*->.....]2.00s
-----执行结束-----
```

```
C:\compare_homework\venv\Scripts\python.exe C:/compare_homework/1.py
```

```
-----执行开始-----
```

```
100%[*****->]64.02s
```

```
-----执行结束-----
```

```
Process finished with exit code 0
```

实验3.3：设计一个替换加密算法，替换方案如下：

a	b	c	d	e	f	g	h	i	j	k	l	m
z	y	x	w	v	u	t	s	r	q	p	o	n
N	o	p	q	r	s	t	u	v	w	x	y	z
m	l	k	j	i	h	g	f	e	d	c	b	a

如：明文为：study hard and make progress every day

密文为：hgfwb sziw zmw nzpv kiltivhh vevib wzb

实验3.4（大圆距离）：假设 (x_1, y_1) 和 (x_2, y_2) 为两点的经度和纬度，两点之间的大圆距离可以用如下公式计算：

$$D = \text{radius} * \arccos(\sin(x_1) * \sin(x_2) + \cos(x_1) * \cos(x_2) * \cos(y_1 - y_2))$$

编写程序，提示用户输入两点的经度和纬度（**负值表示东经或北纬**），显示两点的球面距离，地球半径**radius=6371.01km**，

例如：39.55,-116.25

41.5,87.37

The distance between the two points is:10691.79183231593km