**实验三、输入输出与系列数据**

|  |  |  |  |
| --- | --- | --- | --- |
| **学号** | **姓名** | **年级专业班级** | **成绩** |
| 222018603193119 | 莫健行 | 18级计算机3班 |  |

**实验目的：**掌握python语言的输入输出和系列数据的使用

**实验要求：**独立完成，并上机实践

**实验内容：**

1、利用with语句读取指定的文本文件，并输出到控制台。

源代码：

with open("banana.txt",'r') as f:

content = f.read().split("\n")

for i in range(len(content)):

if i % 2 == 0:

print("")

for j in range(len(content[i])):

if j % 2 == 0:

print(content[i][j],end="")

运行结果（图）：



2、随机产生若干整数，并把这些整数存储到指定的文本文件。

源代码：

import random, time

def get\_rand(n):

random.seed(time.time())

r = []

while n > 0:

n -= 1

r.append(random.randint(-100, 100))

return r

if \_\_name\_\_ == "\_\_main\_\_":

with open("data.txt", 'w') as f:

r = get\_rand(100)

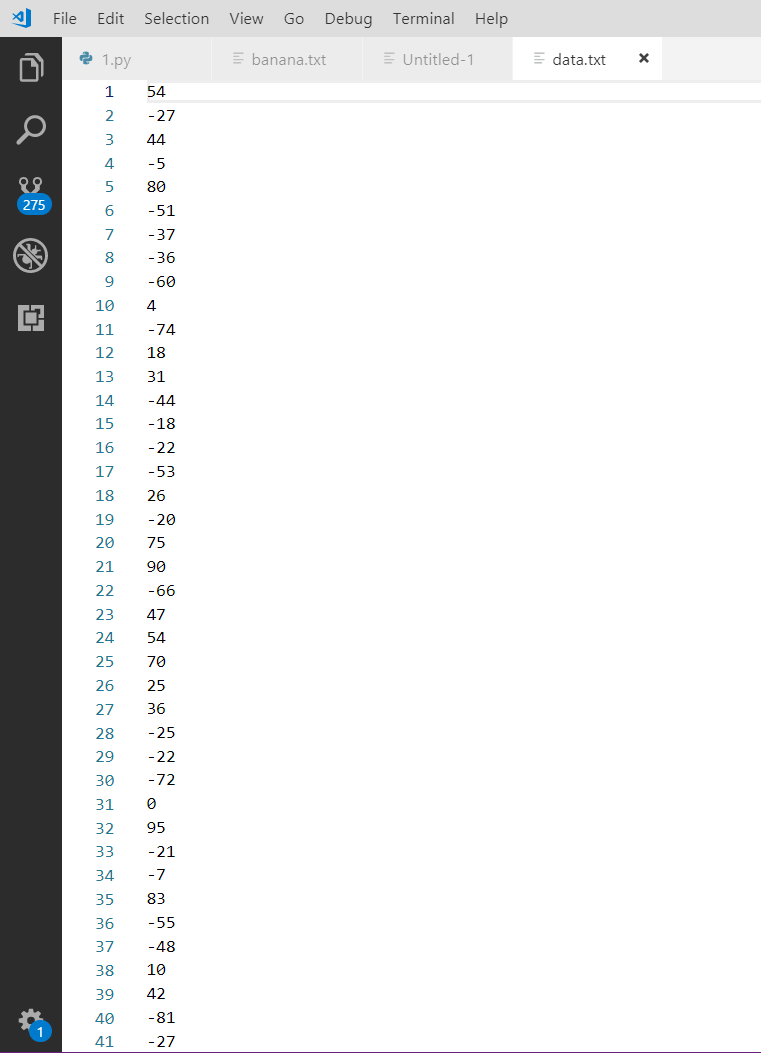
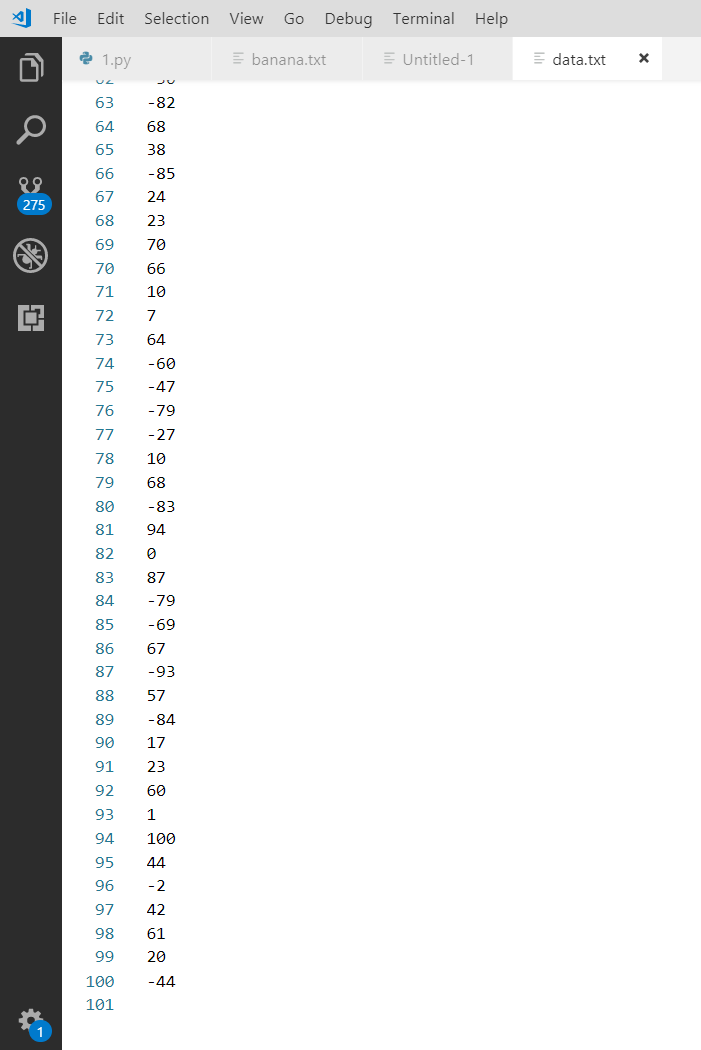
content = ""

for i in r:

content += "%d\n" % i

f.write(content)

运行结果（图）：

3、随机产生20个整数元的列表list，（1）求列表的最大值，最小值，元素之和、平均值，奇数的平均值。（2）输出最接近平均值的元素。

源代码：

**方法一（原始实现）：**

# 只需要通过两次循环解决

import random, time

def get\_rand(n):

random.seed(time.time())

r = []

while n > 0:

n -= 1

r.append(random.randint(-100, 100))

return r

if \_\_name\_\_ == "\_\_main\_\_":

r = get\_rand(20)

print("Instaniate 20 random int:")

max = r[0]

min = r[0]

sum = 0

odds = 0

oddc = 0

a = 0.0

for i in range(20):

print("{} ".format(r[i]), end="")

if r[i] > max:

max = r[i]

if r[i] < min:

min = r[i]

sum += r[i]

if r[i] % 2 != 0:

odds += r[i]

oddc += 1

a = sum / 20.0

odds = float(odds) / float(oddc)

print("\nMax: %d\tMin: %d\t\nSum: %d\tAverage: %.2f\nOdd count: %d\t Odd Average: %.2f" % (max, min, sum, a, oddc, odds))

min = abs(r[0] - a);

index = 0

for i in range(20):

if abs(float(r[i])-a) < min:

min = abs(r[i] - a)

index = i

print("Most approach is the {}th element: {}".format(index, r[index]))

**方法二（More Pythonist）：**

if \_\_name\_\_ == "\_\_main\_\_":

r = get\_rand(20)

print("Instaniate 20 random int:")

max = r.max()

min = r.min()

sum = r.sum()

a = sum / 20.0

odd = [i for i in r if i % 2 != 0]

oddc = len(odd)

odds = sum(odd)

print("\nMax: %d\tMin: %d\t\nSum: %d\tAverage: %.2f\nOdd count: %d\t Odd Average: %.2f" % (max, min, sum, a, oddc, odds))

min = abs(r[0] - a);

index = 0

for i in range(20):

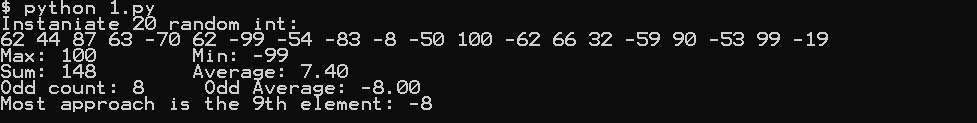
if abs(float(r[i])-a) < min:

min = abs(r[i] - a)

index = i

print("Most approach is the {}th element: {}".format(index, r[index]))

运行结果（图）：



4、系统里面有多个用户，用户的信息目前保存在列表里中，如下所示：

users = ['root','westos']

passwd = ['123','456']

编程模拟用户登陆(判断用户登陆是否成功）

1).判断用户是否存在

2).如果存在

1).判断用户密码是否正确

如果正确，登陆成功，退出循环

如果密码不正确，重新登陆，总共有三次机会登陆

3).如果用户不存在

重新登陆，总共有三次机会

源代码：

使用getpass模块，隐藏输入的密码字符

import getpass

users = ["root", "westos"]

passwd = ["123", "456"]

user\_dict = {}

for i in range(len(users)):

user\_dict[users[i]] = passwd[i]

print("Welcome to Bunbuntu System!")

try\_time = 0

while try\_time < 3:

user = input("User Login:")

if (not user\_dict.get(user)):

print("User %s not exist." % user)

continue

password = getpass.getpass("Password:")

if(password != user\_dict[user]):

print("Login Failed")

try\_time += 1

else:

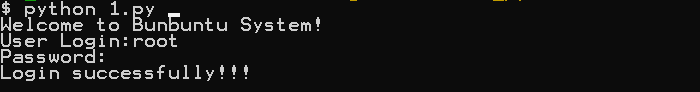
print("Login successfully!!!")

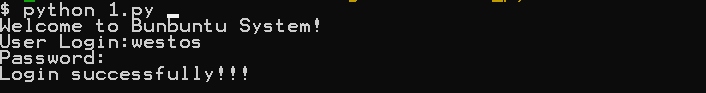
break

print("Failed to Login, max retrieve exceed")

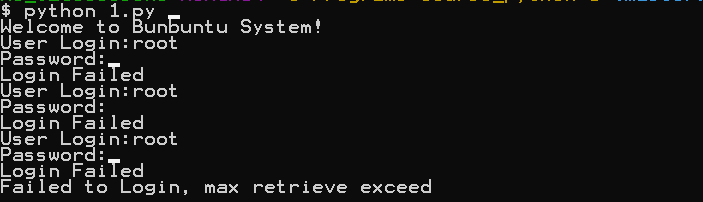
运行结果（图）：

1. 登录成功





1. 登录超过三次



1. 登录不超过三次

