# 第一章

一判断题：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| t | t | f | f | t | t | t | f | t | f |
| 11 | 12 | 13 |  |  |  |  |  |  |  |
| t | t | t |  |  |  |  |  |  |  |

二、

略

# 第二章

一判断

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| t | t | f | t | t | f | t | f | f | f |

二选择题

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| a | a | a | a | c | a | bd | a | b | c |
| 11 | 12 | 13 |  |  |  |  |  |  |  |
| b |  |  |  |  |  |  |  |  |  |

3填空题

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| \ | ; | # | 3.75 | 3.0 | 1.5 | 1.0 | 4 3 | 80 | 345 3 |
| 11 | 12 | 13 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

4略

# 第三章

一选择题

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| a | b | c | d | c | a | d |  |  |  |

二 读程序结果

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 1656  11 21 | 96 | 3.00 4.00 5.00 6.00 |

三 略

# 第四章

一选择题

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| b | c | a | a | b | d | b | ab | a | c |
| 11 | 12 | 13 | 14 | 15 |  |  |  |  |  |
| b | a | c | b | d |  |  |  |  |  |

二 读程序结果

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 5<10<15 | h  e  l  l  o  w  o  r  l  d | 直角三角形 |
| 4 | 5 | 6 |
| 1  12  123  1234 | 5 | \*#\*# |

三

**1、**

**month=int(int(intput())**

**(year%4==0 and year%100!=0) or year%400==0:**

**print(“28天”)**

**else:**

2、

sum=0

number=number if number>0 else -number

number%10

number//=10

3、

n<=3:

break

n==3:

n+=1

4、

Day=0

Total>0:

Total=total//2-2

5、

C=int(intput())

Score>=85:

D+=1

Print(d/c)

四、程序设计

略

## 第五章 列表 课后习题参考答案

1. **单选题：**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | C | 2 | D | 3 | D | 4 | B | 5 | C |
| 6 | A | 7 | B | 8 | C | 9 | C | 10 | A |
| 11 | D | 12 | C | 13 | D | 14 | B | 15 | D |

1. **程序阅读**

1. I would like to own a Benz car

I would like to own a GIANT bicycle

I would like to own a Honda motorcycle

2. ['Make', 'Rain', 'Jack', '秀秀', 'Brown', 'Alex']

3. 44

11

4. ('Rose', 17)

('Jone', 18)

('Make', 20)

('Tom', 21)

5. 91

23

59.375

1. **程序填空**
2. str=str.lower()

c.isalpha():

list.append(c)

list.sort()

1. a=range(0,100)

list1=sorted(list[:10])

list[10:]=list2

1. b=""

a=a+i

for j in str[1::2]:

c=a+b

1. sum=0

for j in range(3):

if i==j:

print(sum)

1. name!="Q":

info[0]==name:

worker.remove(info)

break

1. **程序设计**

1.

month = int(input("请输入月份："))

list=[[3,4,5,"春季"],[6,7,8,"夏季"],[9,10,11,"秋季"],[12,1,2,"冬季"]]

if month in range(1,13):

for i in range(len(list)):

if month in list[i]:

print(month, "月是", list[i][-1])

else:

print("输入的月份不对")

2.

list=[1,1]

for i in range(2,31):

lst.append(list[i-1]+list[i-2])

print(list)

3.

import random

List1 = [random.randint(0,100) for i in range (1000)]

list2= set(List1)

for x in list2:

print(x, ':', List1.count(x))

4.

word = 0

num = 0

other = 0

a = input("请输入一串字符: ")

for k in a:

if k.isalpha():

word += 1

elif k.isdigit():

num += 1

else:

other += 1

print('字母个数是：', word)

print('数字的个数是： ', num)

print('其他字符个数是： ', other)

5.

a = input("请输入一串字符: ")

b=[]

for k in range(len(a)):

if a[k].lower() in "abcdefghijklmnopqrstu":

b.append(chr(ord(a[k])+5))

elif a[k].lower() in "vwxyz":

b.append(chr(ord(a[k])-21))

else:

b.append(a[k])

print(''.join(b))

6.

a = input("请输入:")

b = reversed(list(a))

if list(a) == list(b):

print("是回文")

else:

print("不是回文")

7.

height = eval(input('请输入身高(单位：米)：'))

weight = eval(input('请输入体重(单位：公斤)：'))

bmi = weight / pow(height, 2)

print("BMI 数值为：{:.2f}".format(bmi))

who, nat = "", ""

if bmi < 18.5:

who, nat = "偏瘦", "偏瘦"

elif 18.5 <= bmi < 24:

who, nat = "正常", "正常"

elif 24 <= bmi < 25:

who, nat = "正常", "偏胖"

elif 25 <= bmi < 28:

who, nat = "偏胖", "偏胖"

elif 28 <= bmi < 30:

who, nat = "偏胖", "肥胖"

else:

who, nat = "肥胖", "肥胖"

print("BMI 指标为:国际'{0}', 国内'{1}'".format(who, nat))

8.

name=[]

number=[]

a=('''

====通讯录管理系统====

1.增加姓名和手机

2.删除姓名

3.修改手机

4.查询所有用户

5.根据姓名查找手机号

6.退出

=====================

请选择：

''')

while True:

b=input(a)

if b not in ("1","2","3","4","5","6"):

input("输入有误请重新输入")

else:

if b==("1"):

name1=str(input("请输入姓名"))

if name1 in name:

print("已有此联系人请重新输入")

continue

else:

name.append(name1)

number1=str(input("请输入手机号"))

number.append(number1)

print("输入完成")

elif b==("2"):

name1=str(input("请输入要删除的联系人"))

c=name.index(name1)

name.remove(name1)

del number[c]

print("")

elif b==("3"):

name1=str(input("请输入要修改的联系人"))

c=name.index(name1)

d=str(input("要修改的手机号"))

number[c]=d

print("修改完成")

elif b==("4"):

for i in name:

print("所有用户有",i)

elif b==("5"):

name1=str(input("请输入您要查找的联系人"))

c=name.index(name1)

print("您要查找的手机号是",number[c])

elif b==("6"):

print("感谢使用")

break

else:

print("输入有误请重新输入")

## 第六章

一选择题

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |  |  |
| T | T | F | T | F | F | T |  |  |  |

二

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| Range(1001)  All\_nums.append  Num in sorted\_nums:  Num\_dict[num]=1 | Min=a[‘001’][‘age’  Max=i[‘age’]  Min=i[‘age’] |  |
|  |  |  |
|  |  |  |

三

1、

dic = {"ccg": "123456", "lzc": "abcdefg", "zy": "basdfsdf", "zcw": "1123123basdfsdf"}

for i in range(5):

ID = input('请输入你的用户名：')

if not dic.get(ID):

# 判断用户名是否有误

print('用户名输入错误，剩余输入次数：{}'.format(4 - i))

else:

a = input('用户名正确，请输入密码：')

if a == dic.get(ID):

print('恭喜，密码输入正确')

break

else:

print('密码输入错误，剩余输入次数：{}'.format(4 - i))

2

略

3略

4略

5、包含了密码生成和用户查找功能

以下密码生成

try:

import random # 导入随机数模块

number = int(input('请输入人数：')) # 输入人数

while number < 0:

number = int(input('输入错误，请输入人数：')) # 输入人数

dic = {} # 建一个空的字典

myKey = [] # 建一个空的列表用来存放键

myVaule = [] # 键一个空的列表来存到值

List = [] # 用一个空列表来存放学号用过的值

for i in range(number): # 循环人数的遍数

t = random.randint(1, number) # 产生一个随机学号

while t in List: # 检验这个学号是否使用过

t = random.randint(1, number) # 使用过重新产生

List.append(t) # 用过的学号添加到列表内

myKey.append(2020001000 + t) # 存放键的列表将学号存进去

passage = '' # 密码清空

for i in range(6): # 产生六位密码

if random.randint(1, 2) == 1: # 产生一个随机数觉得该位是数字还是英文单词

passage += chr(random.randint(48, 57)) # 产生一个随机数字

else:

passage += chr(random.randint(65, 90)) # 产生一个随机的大写英文字母

myVaule.append(passage) # 将值添加到存放值的列表

dic = dict(zip(myKey, myVaule)) # 压缩成一个字典

for key, vaule in dic.items(): # 循环字典取出键和值

print(key, vaule) # 输出键和值

except:

print('程序运行出错，请找管理员！')

# 以下产生的密码可以有数字大小写字母和特殊符号，验证输入的信息是否正确

try:

from random import randint # 导入随机数的整数模块

number = int(input('请输入人数：')) # 输入人数

while number < 0:

number = int(input('输入错误，请输入人数：')) # 输入人数

dic = {} # 建一个空的字典

List = [',','.','!','?'] # 用列表来存放密码产生的符号

for i in range(number): # 循环人数的遍数

xueHao = randint(1, number) # 产生一个随机学号

while xueHao in List: # 检验这个学号是否使用过

xueHao =randint(1, number) # 使用过重新产生

passage = '' # 密码清空

for j in range(6): # 产生六位密码

t=randint(1,4) # 产生应该随机数来表示该位密码的类型

if t == 1: # 产生一个随机数觉得该位是数字还是英文单词

passage += chr(randint(48, 57)) # 产生一个随机数字

elif t==2:

passage += chr(randint(65, 90)) # 产生一个随机的大写英文字母

elif t == 3 :

passage += chr(randint(97, 122)) # 产生一个随机的小写英文字母

else:

passage+=List[randint(0,len(List)-1)] # 产生一个列表内的随机符号

dic[chr(19968+i)+chr(20000+i)] = {'学号':2020001000 + xueHao,'密码':passage} # 将键和值存到字典内

for key, vaule in dic.items(): # 循环字典取出键和值

print('姓名：%s'%(key), '学号：%d'%(vaule['学号']),'密码：%s'%(vaule['密码']),sep=' ') # 输出键和值

print('\*'\*20+'查找个人信息'+'\*'\*20)

name=input('请输入你的姓名：') # 输入姓名

if name in dic.keys(): # 如果该姓名在字典中就运行

print('%s同学是该班的新生'%(name),'学号为：%d'%(dic[name]['学号']),'登录密码为：%s'%(dic[name]['密码'])) # 输出个人信息

else:

print('该同学不属于这个班级')

except:

print('程序运行出错，请找管理员！')

# 第七章习题参考答案：

一、 判断题

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |  |  |  |
| T | T | T | F | F | F | F |  |  |  |

二、单选题

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| D | D | C | c |

三、填空题

1. sqrt()

2. type()

3. len()

4. sum()

5. globals()

6. def

7. None

8. 6,10

9. 6

10. XYZ

11. 5.5。

12. global

13. 全局变量,局部变量

14. getrecursionlimit(),setrecursionlimit()

15. locals( ), globals()

三、思考题

1. 1 4 9

2. simple function

3. 4 0

4.求两个数的最大公约数。返回的结果为：3

5. quick

6. 略

7. 略

8. 略

# 第8章课后习题参考答案：

一判断

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| T | F | T | T | T | F | T | T | F | T |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |  |  |
| F | T | T | T | T | F | F | T |  |  |

二 读程序结果

|  |  |
| --- | --- |
| 1 | 2题目印刷格式有误，应为：  class Person:  def setName(self, name):  self.name = name  def greet(self):  print("Hello, I'm %s." % self.name)  person1 = Person()  person1.setName("Jhon")  person1.greet() |
| 你的名字是：张三,年龄是19 | Hello, I'm Jhon. |
| 3 |  |
| 4.7 + 4.5 =9.2  47 // 2 =23 |  |

三

1、

class funX():

a=0

b=0

c=0

def \_\_init\_\_(self,a,b,c):

self.a=a

self.b=b

self.c=c

def Value(self,x):

return self.a\*x\*\*2+self.b\*x+self.c

def Myflist(self,m,n):

answer=[]

for x in range(m,n+1):

answer.append(self.a\*x\*\*2+self.b\*x+self.c)

return answer

def Mroot(self):#求方程根，略，自己完成

pass

#以下测试数据自己设计，这里只是演示

myfun1=funX(2,3,5)

print(myfun1.Value(3.2))

print(myfun1.Myflist(2,65))

2、

class Employee():

def \_\_init\_\_(self,id,name,salary,age,telephone):

self.ID=id

self.Name=name

self.\_\_Salary=salary

self.Age=age

self.Telephone=telephone

self.realSalary=self.\_\_Salary

@property

def Salary(self):#获得工资

return self.\_\_Salary

@Salary.setter

def setSalary(self,money):#修改工资

self.\_\_Salary=money

self.realSalary=self.\_\_Salary

def waterElctCharge(self,water,elct):

self.realSalary=self.realSalary-water-elct

return water+elct

def HouseAccumulat(self,accumulation):

self.realSalary=self.realSalary-accumulation

return accumulation

#以下程序为测试功能，请大家自己完善

employee1=Employee(1001,'张三',50000,40,1377705249)#示例，员工工资，请用循环生成，ID递增方法获得，其它从键盘输入

print(employee1.ID,employee1.Name,employee1.Salary,employee1.Age,employee1.Telephone)

employee1.setSalary=30000#可以用这个方法改工资

print(employee1.Salary)

WaterElct=employee1.waterElctCharge(200,300)#扣水电费后实发工资

print("水电费：",WaterElct)

print("工资：",employee1.Salary,"实发工资",employee1.realSalary)

#employee1.Salary=20000 不能修改，只能通过方法来完成

3、

class Motor:

number=0

def Work1(self):

print("打开发动机，发动机开始运行..请挂挡.")

def work(self,inputNumber):

self.number=inputNumber

if self.number>=1:

print("出发喽！")

if self.number==1:

print("我们慢慢玩过去！")

#其它自行模拟

class Chassis:

def Work2(self):

print("根据路况开始调节底盘...")

# 其它自行模拟

class Seat:

passengeNumber=0

def Work3(self):

print("驾驶员调整座椅...")

def passengeSet(self,number):

self.passengeNumber=number

if self.passengeNumber<=0:

print("人员设置数量有问题")

def passengeGetUP(self,number):

if number>self.passengeNumber:

print("超员，不能开车！")

else:

print("乘客人数{}符合规定，可以出发！".format(number))

class Shell:

pass

#其它自行模拟

class Vehicle(Motor,Seat,Chassis):

def Run(self):

print("汽车开始运动...")

# 其它自行模拟，可以增加改变档位功能

v = Vehicle()

v.passengeSet(5)#设置乘客人数5人

v.passengeGetUP(4)#上车人数

v.Work2()

v.Work1()

v.Work3()

mySpeed=int(input("输入档位："))

v.work(mySpeed)

v.Run()

4略

# 第九章课后习题参考答案：

1. 判断题

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| F | F | F | T | T | T | T |

1. 填空题
2. open
3. with
4. exists()
5. isfile()
6. isdir()
7. splitext()\_\_\_\_。
8. 程序设计

略

# 第十章课后习题参考答案：

1. 选择题

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |  |  |
| B | D | A | B | D |  |  |

1. 判断题

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| F | T | T | T | T | T | T |

三、程序设计题

略

# 第十一章课后习题参考答案：

略（自行完成）

# 第十二章课后习题参考答案：

一、程序填空

1、print(frame.loc['青菜','价格'])

2、print(frame2.loc['猪肉','价格'])

3、pd.read\_excel

df2=df.loc[:,['面积','单价'，,'姓名','销售人员']]

二、程序设计题

1、（调试该程序实现读写文件需安装模块xlrd和xlwt等模块）

import pandas as pd

sExcelFile = "fcxs.xlsx"

df = pd.read\_excel(sExcelFile, sheet\_name='Sheet11')

pd.set\_option('display.max\_colwidth', 1000)#单列数据宽度，以字符个数计算，超过时用省略号表示

pd.set\_option('display.width', 1000)#数据显示区域的总宽度，以总字符数计算

pd.set\_option('display.max\_columns', 1000)#最大显示列数，超过该值用省略号代替，为None时显示所有列

# #中文标题时对齐数据这两个参数的默认设置都是False

pd.set\_option('display.unicode.ambiguous\_as\_wide', True)#中文标题时对齐数据

pd.set\_option('display.unicode.east\_asian\_width', True)#中文标题时对齐数据

values={'契税':0.15}

df.fillna(value=values,inplace=True,axis=0)

values={'契税':0.15,'契税总额':df.面积\*df.单价\*df.契税,'房价总额':df.面积\*df.单价}

df.fillna(value=values,inplace=True,axis=0)

print(df)

df['契税'] = df['面积'].apply(lambda x:0.35 if x>=150 else 0.25)

print(df)

df.to\_excel(r'DataTest1.xls',sheet\_name='Sheet1')

print(df.describe().round(2))

print(max(df['单价']))

2略

3略