第一章

1. 选择题

1、C 2、D 3、C 4、C 5、B 6、D 7、B 8、B 9、D 10、C

**二、思考题**

略

第二章

一、选择题

1. A 2、C 3、B 4、D 5、B 6、B 7、A 8、A 9、A 10、B 11、C
2. B 13、C 14、C 15、B

**二、程序设计题**

1、

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

print('Hello Python!'.center(20))

print('\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*')

2、

N=int(input("请输入整数N："))

print(N\*\*3)

3、

x=input("请输入x的值：")

y=input("请输入y的值：")

x,y=y,x

print("x=",x,"y=",y)

第三章

一、选择题

1. B 2、C 3、B 4、C 5、B 6、C 7、A 8、C 9、D 10、D 11、A 12、D

13、D 14、A 15、BD

**二、程序设计题**

1、

x=int(input("请输入一个整数："))

print(bin(x))

print(oct(x))

print(hex(x))

2、

s=input("请输入一个字符串：")

print(s.center(10,'\*'))

print('{:}'.format(s[0:10]))

3、

s=input("请输入一个字符串：")

print(s.lower())

print(s.upper())

4、

s = " python程序设计 "

s1=s.strip(' ')

s2=s1.replace('p','P')

print(s2)

第四章

1. 选择题
2. A 2、A 3、C 4、B 5、D 6、D 7、A 8、A 9、B 10、B
3. 程序设计题

1、

count = 0 # 用于记录用户错误次数

while count < 3:

user = input("请输入您的账号：")

pwd = input("请输入您的密码：")

if user == 'admin' and pwd == '123': # 进行账号密码比对

print('登录成功')

break

else:

print("用户名或密码错误")

count += 1 # 初始变量值自增 1

if count == 3: # 如果错误次数达到 3 次，则提示并退出

print("输入错误次数过多，请稍后再试")

else:

print(f"您还有{3 - count}次机会") # 显示剩余次数

2、

count=0

n=int(input("请输入学生人数："))

for i in range(n):

score=int(input("请输入学生成绩："))

if score<60:

count+=1

continue

print(count/n)

3、

x=int(input("请输入第一个数："))

y=int(input("请输入第二个数："))

operator = input('请选择要执行的运算符：+、-、\*、/' + '\n')

if operator == "+":

print("计算结果为:", x + y)

elif operator == '-':

print("计算结果为:", x - y)

elif operator == '\*':

print("计算结果为:", x \* y)

elif operator == '/':

if y == 0:

print('被除数不能为 0!')

else:

print("计算结果为:", x / y)

4、

import random

print("\*\*\*\*\*\*\*\* 猜数字小游戏 \*\*\*\*\*\*\*\*")

print("猜数字游戏规则：输入一个 1-100 以内的数字,总共有3次机会！")

random\_num = random.randint(1, 100) # 生成一个1-100的随机数

for i in range(1,4):

number = input("请输入一个数字:")

if number.isdigit() is False:

print('请输入一个正确的数字')

elif int(number) < 0 or int(number) > 100:

print("请输入 1-100 范围的数字")

elif random\_num == int(number):

print("恭喜你用了%d 次猜对了" % i)

break

elif random\_num > int(number):

print("很遗憾，你猜小了")

else:

print("很遗憾，你猜大了")

if i == 3:

print("很遗憾，%d次机会已用尽，游戏结束,答案为%d" % (i, random\_num))

5、

for i in range(1,7):

for j in range(1,(7-i)):

print(' ',end='')

for k in range(1,2\*i):

print('\*',end='')

print()

6、

i=1

while i<6:

j=1

while j<=5-i:

print(' ',end='')

j+=1

j=1

while j<=2\*i-1:

if j==1 or i==5 or j==2\*i-1:

print('\*',end='')

else:

print(' ',end='')

j+=1

print()

i+=1

7、

for i in range(1,50):

for j in range(1,40):

if i+j==50 and 2\*i+4\*j==160:

print('鸡有%d只，兔有%d只' % (i,j))

8、

num = int(input("请输入一个四位数："))

single = int(num / 1000)

ten = int(num / 100 % 10)

hundred = int(num / 10 % 10)

ths = int(num % 10)

num1 = ths \* 1000 + hundred \* 100 + ten \* 10 + single

if num == num1:

print(num,"是回文数")

else:

print(num,"不是回文数")

# 第5章 字典和集合

1. **选择题**
2. C
3. ABCD
4. A
5. B
6. D
7. A
8. A
9. BCD
10. B
11. **程序设计题**

无

# 第6章 组合数据类型

1. **选择题**
2. B
3. C
4. D
5. B
6. A
7. A
8. C D
9. D
10. B
11. A
12. D
13. C
14. B
15. D
16. D
17. B
18. **程序设计题**

# 遍历字符串，使用字典记录各个字符及其出现的次数

str = 'skdaskerkjsalkj'

count = {}

for i in str:

if i in count.keys():

count[i] += 1

else:

count[i] = 1

print(count)

def Odd\_Even\_Numbers(num\_list):

"""

@param num\_list:

@return: list1,list2

"""

x = [] # 存偶数

y = [] # 存奇数

for i in num\_list:

if i % 2 == 0:

x.append(i)

else:

y.append(i)

return x, y

nums = input("请输入你要求解的数据，并用“,”隔开：")

nums\_list = []

if len(nums) < 1:

print("未输入任何数据！")

exit()

else:

nums\_list = nums.split(",")

# 将字符串列表转换成为整形列表

int\_nums\_list = []

for num in nums\_list:

int\_nums\_list.append(int(num))

evens\_list, odds\_list = Odd\_Even\_Numbers(int\_nums\_list)

# 非奇数的处理

sumEvens = 0

evens = []

if len(evens\_list) > 0:

for e in evens\_list:

if e > 10:

sumEvens += e

evens.append(e)

print("所有大于10的非奇数为：", evens)

print("所有大于10的非奇数之和为：", sumEvens)

else:

print("没有找到非奇数!")

# 奇数的处理

if len(odds\_list) > 0:

odds\_list = sorted(odds\_list, reverse=True)

# odds\_list.sort(reverse=True)

print("所有奇数从大到小排序为：", odds\_list)

print("所有奇数中最大的是：", max(odds\_list))

else:

print("没有找到奇数!")

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))

4.

# 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("gxxd"+"{:03d}".format(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7章 函数

1. **选择题**
2. C
3. ABC
4. D
5. C
6. A
7. B
8. D
9. **程序设计题**

1.

def evenSum():

sum = 0

for i in range(1,101):

if i % 2 == 0:

sum += i

return sum

print(evenSum())

2.

def multiply():

p = 1

for i in range(3,21):

p \*= i

return p

print(multiply())

3.

def count\_str(str):

dt={} # 声明字典dt，用来暂时保存所有的字母和其统计个数，格式为{字母:个数}

dt1={} # 声明字典dt1，用来保存所有统计个数为1的字母，格式为{字母:个数}

for i in str:

dt[i]=str.count(i,0)

for k,count in dt.items():

if count==1:

dt1.update({k:count})

return dt1

mystr=input("请输入需要统计的字符串：")

print(count\_str(mystr))

# 第8章

判断题

1.√

2.×

3.√

4.√

5.√

**程序设计题1**

import math

class Circle:

def \_\_init\_\_(self, r):

self.r = r

def get\_area(self):

return self.r \* self.r \* math.pi

def get\_cal(self):

return 2 \* self.r \* math.pi

if \_\_name\_\_ == '\_\_main\_\_':

radius = float(input("请输入圆的半径："))

if radius > 0:

r1 = Circle(radius)

print("该圆的周长是{:.3f}".format(r1.get\_cal()))

print("该圆的面积是{:.3f}".format(r1.get\_area()))

else:

print("半径不能小于0")

**程序设计题2**

class Person:

def \_\_init\_\_(self, name):

self.name = name

def buy(self, car):

print(self.name + '在' + carstore.name + '4S店购买了一辆' + car.name)

class CarStore:

def \_\_init\_\_(self, name):

self.name = name

class Car:

def \_\_init\_\_(self, name, cartstore):

self.name = name

self.carstore = carstore

person = Person('Joker')

carstore = CarStore('BMW')

car = Car('BMWx7', carstore)

person.buy(car)

**程序设计题3**

**def Rank(sum):**

**# 对成绩的等级进行判断**

**if 90 <= sum <= 100:**

**return "A"**

**elif 80 <= sum < 90:**

**return "B"**

**elif 70 <= sum < 80:**

**return "C"**

**elif 60 <= sum < 70:**

**return "D"**

**elif sum < 60:**

**return "E"**

**class Student:**

**def \_\_init\_\_(self, name, sno, yuwen, shuxue, yingyu):**

**# 初始化学生对象**

**self.name = name**

**self.sno = sno**

**self.yuwen = yuwen**

**self.shuxue = shuxue**

**self.yingyu = yingyu**

**def Rankyuwen(self):**

**# 语文等级**

**return Rank(self.yuwen)**

**def Rankshuxue(self):**

**# 数学等级**

**return Rank(self.shuxue)**

**def Rankyingyu(self):**

**# 英语等级**

**return Rank(self.yingyu)**

**def All\_News(self):**

**# 输出学生的所有信息**

**print("姓名：{}，学号{}，语文成绩{}等级{}，数学成绩{}等级{}，英语成绩{}等级{}".format(self.name, self.sno, self.yuwen, self.Rankyuwen(),**

**self.shuxue, self.Rankshuxue(), self.yingyu,**

**self.Rankyingyu()))**

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

**list = []**

**ls = []**

**i = 1**

**# 输入10个学生个信息，进行处理，应再加入学号不能相同的判断，可使用list列表中存放的学号进行判断处理**

**while i <= 3:**

**name = input("第{}名学生名字：".format(i))**

**sno = int(input("第{}学生学号:".format(i)))**

**dir = {name: sno}**

**list.append(dir)**

**yuwen = float(input("第{}名学生的语文成绩：".format(i)))**

**shuxue = float(input("第{}名学生的数学成绩：".format(i)))**

**yingyu = float(input("第{}名学生的英语成绩：".format(i)))**

**ls.append(Student(name, sno, yuwen, shuxue, yingyu))**

**i += 1**

**print(list)**

**print("所有学生信息：{}".format(list))**

**for i in range(0, 3):**

**ls[i].All\_News()**

**程序设计题4**

**class Role:**

**def \_\_init\_\_(self, name, HP, MP, status, skill):**

**'''**

**定义角色的名字，血量，蓝量，状态，以及技能和技能伤害值**

**'''**

**self.name = name**

**self.HP = HP**

**self.MP = MP**

**self.status = status**

**self.skill = skill**

**self.damage = (MP / HP) \* 10**

**def Skill(self, ra, rb):**

**'''**

**角色A对角色B发起攻击，并进行蓝量的判断。**

**'''**

**if ra.MP > 0:**

**rb.HP -= rb.damage**

**ra.MP -= (self.HP / self.MP) \* 10**

**if ra.MP < 0:**

**ra.MP = 0**

**print("角色{}使用技能{}对{},造成伤害{}".format(r1.name, r1.skill, r2.name, r1.damage))**

**else:**

**print("没蓝了。")**

**def Roles(self):**

**print("角色{}，剩余HP{}，剩余MP{}".format(self.name, self.HP, self.MP))**

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

**'''**

**创建两个角色**

**'''**

**r1 = Role("孙悟空", 100, 100, "健康", "火眼金睛")**

**r2 = Role("蜘蛛精", 100, 100, "亚健康", "天罗地网")**

**while True:**

**'''**

**直到某一方血量为0是跳出循环**

**'''**

**chioce = (int(input("请输入1/2，1则{}发动技能，2则{}发动技能:".format(r1.name, r2.name))))**

**if chioce == 1:**

**r1.Skill(r1, r2)**

**r1.Roles()**

**r2.Roles()**

**elif chioce == 2:**

**r2.Skill(r2, r1)**

**r2.Roles()**

**r1.Roles()**

**else:**

**print("输入无效")**

**if r1.HP == 0:**

**print("哎哟，%s居然死啦！" % r1.name)**

**break**

**if r2.HP == 0:**

**print("诛杀此妖，干得漂亮，可以继续西天取经了！")**

**break**

# 第9章

**1.D**

**2.A**

**3.D**

**4.D**

**5.B**

**6.A**

**7.B**

**8.C**

**9.D**

**10.A**

# 第10章

**1.C**

**2.D**

**3.C**

**4.C**

**5.D**

**6.B**

**7.C**

**8.B**

**9.D**

**10.A**

**11.B**

**12.B**

**13.A**

**14.C**

**15.B**

**16.A**

# 第11章

1. **选择题**

1.D

2.D

3.D

4.B

5.D

1. **判断题**

1.x

2.√

3.√

4.√

5.√

6.√

7.√

8.x

9.√

10.x

1. **程序设计题**

**程序设计题1**

import tkinter as tk

window = tk.Tk()

window.title('喜欢的人')

window.geometry('400x350')

var1 = tk.StringVar() # 创建变量，用var1用来接收鼠标点击具体选项的内容

def print\_selection():

if check1.get() == 1:

value1 = '黄蓉'

else:

value1 = ''

if check2.get() == 2:

value2 = '华筝'

else:

value2 = ''

if check3.get() == 3:

value3 = '郭靖'

else:

value3 = ''

if check4.get() == 4:

value4 = '杨康'

else:

value4 = ''

value = value1 + value2 + value3 + value4

if value == '':

var1.set("我没有喜欢的人")

else:

var1.set("我喜欢" + value)

# 复选框

check1 = tk.IntVar()

check2 = tk.IntVar()

check3 = tk.IntVar()

check4 = tk.IntVar()

c1 = tk.Checkbutton(window, text="黄蓉", variable=check1, onvalue=1, offvalue=0, height=1, width=10)

c2 = tk.Checkbutton(window, text="华筝", variable=check2, onvalue=2, offvalue=0, height=1, width=10)

c3 = tk.Checkbutton(window, text="郭靖", variable=check3, onvalue=3, offvalue=0, height=1, width=10)

c4 = tk.Checkbutton(window, text="杨康", variable=check4, onvalue=4, offvalue=0, height=1, width=10)

c1.place(relx=0.3, rely=0.05)

c2.place(relx=0.3, rely=0.20)

c3.place(relx=0.3, rely=0.35)

c4.place(relx=0.3, rely=0.50)

l = tk.Label(window, bg='pink', fg='blue', font=('Arial', 12), width=40, textvariable=var1)

l.place(relx=0.035, rely=0.65)

b1 = tk.Button(window, text='喜欢的明星', width=20, height=1, command=print\_selection)

b1.place(relx=0.3, rely=0.8)

window.mainloop()

**程序设计题2**

# 程序设计题2：简单计算器程序

from tkinter import \*

root = Tk()

root.title('计算器程序')

frame = Frame(root)

frame.pack(padx=10, pady=10)

v1 = StringVar()

v2 = StringVar()

v3 = StringVar()

def test(content):

return content.isdigit()

testCMD = frame.register(test)

# focusout指定在当前输入框失去焦点时，代表输入完，会去调用test校验<tab>键可测试

e1 = Entry(frame, width=10, textvariable=v1).grid(row=0, column=0) # width的单位是字符数

Label(frame, text='+').grid(row=0, column=1)

e2 = Entry(frame, width=10, textvariable=v2).grid(row=0, column=2)

Label(frame, text='=').grid(row=0, column=3)

e3 = Entry(frame, width=10, textvariable=v3, state='readonly').grid(row=0, column=4)

def calc():

result = int(v1.get()) + int(v2.get())

v3.set(str(result))

Button(frame, text='计算结果', command=calc).grid(row=1, column=2, pady=5)

mainloop()

**程序设计题3**

from tkinter import \*

win = Tk() # 窗口

win.geometry("400x250") # #窗口位置400后面是字母x

menu = Menu(win) # 创建主菜单

menu\_main1 = Menu(menu) # 创建主菜单1

menu\_main1.add\_command(label='登陆系统') # 一个子菜单

menu\_main1.add\_command(label='修改密码') # 一个子菜单

menu\_main1.add\_command(label='注销账号') # 一个子菜单

menu\_main1.add\_command(label='退出') # 一个子菜单

menu\_main2 = Menu(menu) # 创建主菜单2

menu\_main2.add\_command(label='教学评价') # 一个子菜单

menu\_main2.add\_command(label='作业') # 一个子菜单

menu\_main2.add\_command(label='成绩录入') # 一个子菜单

menu\_main3 = Menu(menu) # 创建主菜单3

menu\_main3.add\_command(label='教学反馈') # 一个子菜单

menu\_main3.add\_command(label='师生互联') # 一个子菜单

menu\_main4 = Menu(menu) # 创建主菜单4

menu\_main4.add\_command(label='帮助') # 一个子菜单

menu.add\_cascade(label='系统', menu=menu\_main1) # 添加主菜单1到主菜单上

menu.add\_cascade(label='教学', menu=menu\_main2) # 添加主菜单2到主菜单上

menu.add\_cascade(label='反馈', menu=menu\_main3) # 添加主菜单3到主菜单上

menu.add\_cascade(label='帮助', menu=menu\_main4) # 添加主菜单4到主菜单上

win.config(menu=menu) # 设置主菜单到界面

win.mainloop()