**实验报告2**

WA2214014 杨跃浙 人工智能2班

**实验内容：**

**实验内容1-Python基本用法**

1. 编写程序，输入任意大的自然数，输出各位数字之和。

2. 编写程序，输入两个集合setA和setB，分别输出它们的交集、并集、差集setA–setB。

3. 编写程序，输入一个自然数，输出它的二进制、八进制、十六进制表示形式。

4. 编写程序，输入一个包含若干整数的列表，输出一个新列表，新列表中只包含 原列表中的偶数。

5. 编写程序，输入两个分别包含若干个整数的列表lstA和lstB，输出一 个字典，要求使用列表lstA中的元素作为键，列表lstB中的元素作为值， 并且最终字典中的元素数量取决于lstA和lstB中元素最少的列表的数量。

6. 编写程序，输入一个包含若干整数的列表，输出新列表，要求新列

表中的所有元素来自于输入的列表，并且降序排列。

7. 编写程序，输入一个包含若干整数的列表，输出列表中所有的整数

连乘的结果。

8. 编写程序，输入两个各包含2个整数的列表，分别表示城市中两个地点的坐标，输出两点之间的曼哈顿距离。

9. 编写程序，输入包含若干集合的列表，输出这些集合的并集。要求使用reduce()

函数和lambda表达式完成。

10. 编写程序，输入等比数列的首项、公比（不等于1且小于36的正整数）和一个 自然数n，输出这个等比数列前n项的和。关键步骤要求使用内置函数int()

11. 编写程序，输入一个字符串，输出其中出现次数最多的字符及其出现的次数。

要求使用字典。

**实验内容2-Python常用序列**

1. 编写程序，创建一个包含1-10之间的所有素数的列表。

2. 编写程序，创建一个包含1-100之间的所有偶数的列表。

3. 使用列表表示一个矩阵

4. 统计全班同学某门课程的及格率，使用列表表示成绩序列scores=[78,69,53,97,88,31,74,92]

5. 从控制台读入10个学生分数，并放入到一个列表中。

6. 假设读入的分数中可能有些是无效的分数(<0,>100)，请从分数列表中删除这些无效分数。

7. 从控制台读入一个字符串，把它转换为列表，然后输出其中每一个元音字母 { a, e, o, i, u }出现的次数。

8. 从一个分数列表中选出>=60的分数，构成一个及格成绩列表。

9. 编写程序，输入一个字符串，输出其中出现次数最多的字符及其出现的次数。

要求使用字典（结合课件PPT的例子）。

**实验原理：**

1. **主要代码**

**实验内容1-Python的基本用法**

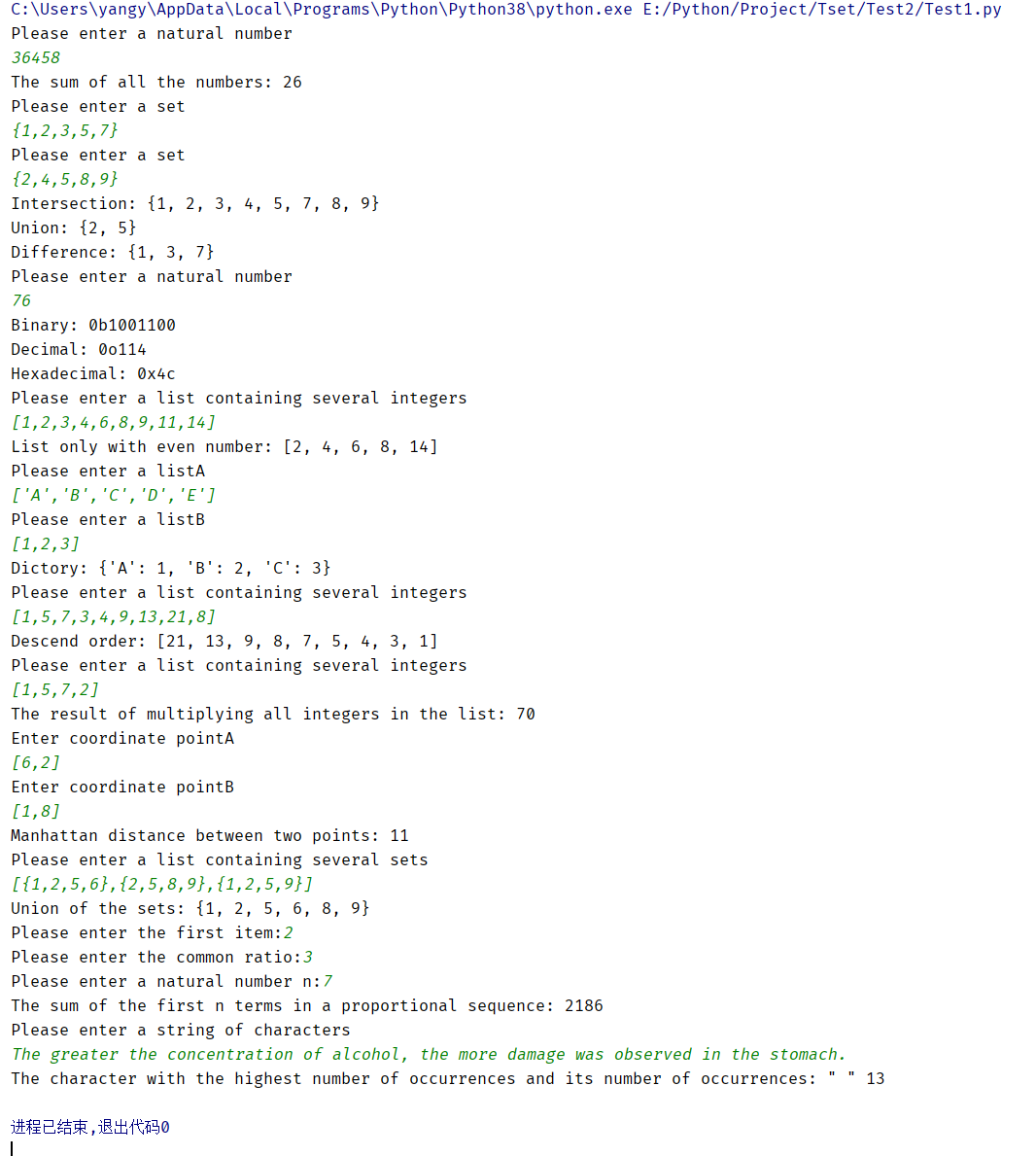
from functools import reduce  
def test1():  
 x=input("Please enter a natural number\n")  
 print("The sum of all the numbers:",sum(map(int,x)))  
def test2():  
 setA = eval(input("Please enter a set\n"))  
 setB = eval(input("Please enter a set\n"))  
 print("Intersection:", setA | setB)  
 print("Union:", setA & setB)  
 print("Difference:", setA - setB)  
def test3():  
 x = int(input("Please enter a natural number\n"))  
 print("Binary:", bin(x))  
 print("Decimal:", oct(x))  
 print("Hexadecimal:", hex(x))  
def test4():  
 lis=eval(input("Please enter a list containing several integers\n"))  
 lis=list(filter(lambda x: x % 2 == 0,lis))  
 print("List only with even number:",lis)  
def test5():  
 lisA=eval(input("Please enter a listA\n"))  
 lisB=eval(input("Please enter a listB\n"))  
 dictory = dict(zip(lisA,lisB))  
 print("Dictory:",dictory)  
def test6():  
 lis= eval(input("Please enter a list containing several integers\n"))  
 lis.sort(key=None, reverse=True)  
 print("Descend order:",lis)  
def test7():  
 lis=eval(input("Please enter a list containing several integers\n"))  
 result=reduce(lambda x,y:x\*y,lis)  
 print("The result of multiplying all integers in the list:",result)  
def test8():  
 pointA = eval(input("Enter coordinate pointA\n"))  
 pointB = eval(input("Enter coordinate pointB\n"))  
 distance = map(lambda x, y: abs(x - y), pointA, pointB)  
 print("Manhattan distance between two points:", sum(distance))  
def test9():  
 sets=eval(input("Please enter a list containing several sets\n"))  
 union\_set=reduce(lambda x,y:x.union(y),sets)  
 print("Union of the sets:",union\_set)  
def test10():  
 a1 = int(input("Please enter the first item:"))  
 q = int(input("Please enter the common ratio:"))  
 n = int(input("Please enter a natural number n:"))  
 sum = int(a1\* (1 - q \*\* n) / (1 - q))  
 print("The sum of the first n terms in a proportional sequence:",sum)  
def test11():  
 str = input("Please enter a string of characters\n")  
 dic= {}  
 for i in str:  
 dic[i] = str.count(i)  
 v = max(dic.values())  
 for key, value in dic.items():  
 if (value == v):  
 print("The character with the highest number of occurrences and its number of occurrences:",'"'+key+'"', v)  
if \_\_name\_\_ == '\_\_main\_\_':  
 test1()  
 test2()  
 test3()  
 test4()  
 test5()  
 test6()  
 test7()  
 test8()  
 test9()  
 test10()  
 test11()

**实验内容2-Python常用序列**

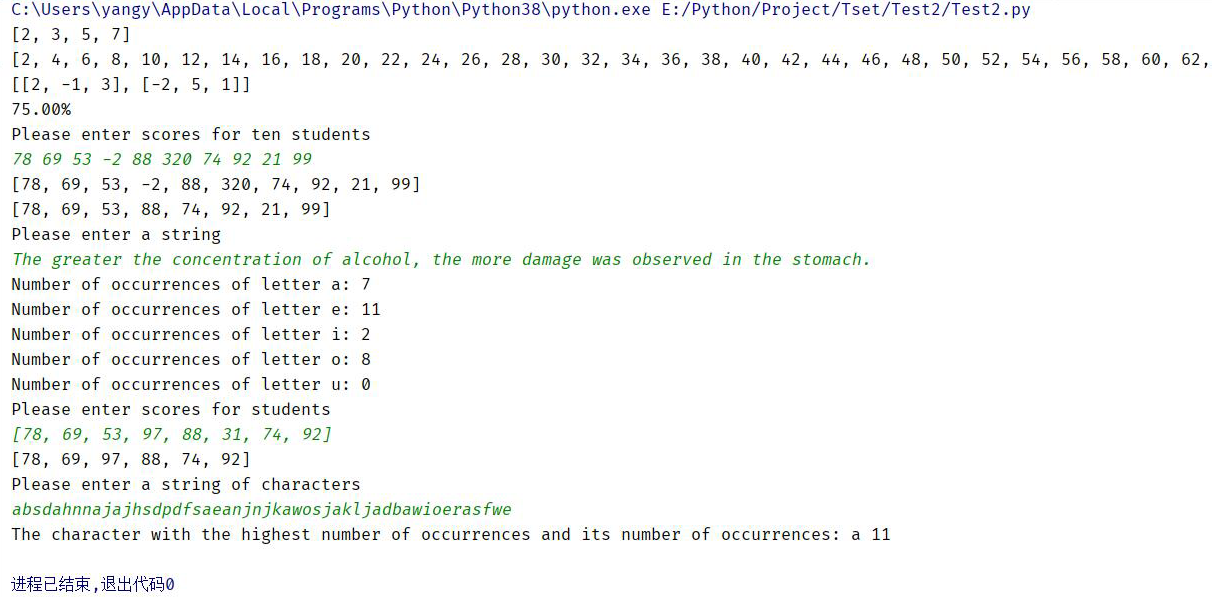
def is\_prime(n):  
 if n < 2:  
 return False  
 elif n == 2:  
 return True  
 else:  
 a = int(n \*\* (1 / 2) + 1)  
 for i in range(2, a + 1):  
 if n % i == 0:  
 return False  
 return True  
def test1():  
 lst=[i for i in range(1,11) if is\_prime(i)]  
 print(lst)  
def test2():  
 lst=[i for i in range(1,101) if i%2==0]  
 print(lst)  
def test3():  
 lst=[[2,-1,3],[-2,5,1]]  
 print(lst)  
def test4():  
 scores = [78, 69, 53, 97, 88, 31, 74, 92]  
 pa=sum(map(lambda score:score//60,scores))  
 print('{:.2%}'.format(pa/len(scores)))  
def test5():  
 global x  
 x=input("Please enter scores for ten students\n").split()  
 x=list(map(int,x))  
 print(x)  
def test6():  
 global x  
 x=list(filter(lambda score:0<=score<=100,x))  
 print(x)  
def test7():  
 str=input("Please enter a string\n")  
 str=list(str)  
 counta=str.count('a')+str.count('A')  
 counte=str.count('e')+str.count('E')  
 counti=str.count('i')+str.count('I')  
 counto=str.count('o')+str.count('O')  
 countu=str.count('u')+str.count('U')  
 print("Number of occurrences of letter a:",counta)  
 print("Number of occurrences of letter e:",counte)  
 print("Number of occurrences of letter i:",counti)  
 print("Number of occurrences of letter o:",counto)  
 print("Number of occurrences of letter u:",countu)  
def test8():  
 scores=eval(input("Please enter scores for students\n"))  
 scores=list(filter(lambda score:score>=60,scores))  
 print(scores)  
def test9():  
 str = input("Please enter a string of characters\n")  
 dic= {}  
 for i in str:  
 dic[i] = str.count(i)  
 v = max(dic.values())  
 for key, value in dic.items():  
 if (value == v):  
 print("The character with the highest number of occurrences and its number of occurrences:",key, v)  
if \_\_name\_\_ == '\_\_main\_\_':  
 test1()  
 test2()  
 test3()  
 test4()  
 test5()  
 test6()  
 test7()  
 test8()  
 test9()

1. **运行结果**

**实验内容1-Python的基本用法**



**实验内容2-Python常用序列**



**小结与讨论：**

**实验内容1-Python的基本用法**

题1中采用map函数处理。题3中0b，0o，0x，分别代表二进制，八进制，十进制。题4中采用filter和lambda表达式，简单解决这个问题。题5中采用zip函数解决长度不一致的问题。题7中引入reduce函数。

**实验内容2-Python常用序列**

题1 中编写is\_prime函数，而后采用列表推导式解决。题2中输出的列表没有显示完全，但是题目较简单，所以后续列表就算了。题3中直接用列表形式表示矩阵，当然也可以采取外部库numpy等更好的表示矩阵。题4中以百分号形式，保留两位有效数字。题5中引入global x，在题6中继续使用。题7中统计时应注意大小写。题8中引入filter函数。题9与实验1中题11一样。