

Python程序设计练习

开始时间 2024/03/13 14:08:00 结束时间 2024/03/31 23:08:00 答题时长 26460分钟
答卷类型 标准答案 总分 265

判断题

得分：暂无 总分： 11

1-1 从编译和解释的角度看，Python语言属于编译性语言()。(1分)

☐ T ☒ F

1-2 Python中，布尔型是一种特殊的整型，它只有True和False两种值，分别对应整数1和0。(1分)

☒ T ☐ F

1-3 在Python中的关系运算符可以连续使用。如：3<a<4。(1分)

☒ T ☐ F

1-4 在循环中continue语句的作用是退出循环的当前迭代。(1分)

☒ T ☐ F

1-5 在Python循环中，对于带有else子句的循环，如果因为执行了break语句而退出的话，会执行else子句的代码。(1分)

☐ T ☒ F

1-6 列表lst=[12, -5, -22, -10, -26, 35, 0, 49, 3, -21],lst[::-1]的结果是[12, -5, -22, -10, -26, 35, 0, 49, 3, -21]。(1分)

☒ T ☐ F

1-7 列表lst=[12, -5, -22, -10, -26, 35, 0, 49, 3, -21],lst[::-1]的结果是[-21, 3, 49, 0, 35, -26, -10, -22, -5, 12]。(1分)

☒ T ☐ F

1-8 语句 `print("输出结果是{:08.2f}".format(14.345))` 的输出是：输出结果是00014.34。(1分)

☐ T ☒ F

1-9 len(set([0,4,5,6,0,7,8]))的结果是7。(1分)

☐ T ☒ F

1-10 列表可以作为字典的键。(1分)

☐ T ☒ F

1-11 在Python中，try/except语句可以有一个可选的else子句，出现在所有except子句之后，若发生异常但没有匹配的except子句则执行else语句块。(1分)

☐ T ☒ F

单选题

得分：暂无 总分：24

2-1 Python指令a=eval(input())执行后，输入3+5。请问a的值是多少 ()

(1分)

- ☒ A. 8
- ☐ B. '8'
- ☐ C. '3'+5'
- ☐ D. '3+5'

2-2 有如下代码：

(1分)

```
i = sum = 0
while i <= 4:
    sum += i
    i = i+1

print(sum)
```

以上代码输出的结果是？

- ☐ A. 0
- ☒ B. 10
- ☐ C. 4
- ☐ D. 以上都不对。

2-3 使用_____逻辑运算符时，两个子表达式都必须为true，复合表达式才能为true。

(1分)

- ☐ A. or
- ☒ B. and
- ☐ C. not
- ☐ D. either or or and

2-4 在Python中，_____符号可以作为不等于运算符。

(1分)

- ☐ A. ==
- ☐ B. <>
- ☐ C. <=
- ☒ D. !=

2-5 Python过程化程序设计的三种基本程序控制结构是()。

(1分)

- ☒ A. 顺序结构、选择结构、循环结构

- ☐ B. 输入、处理、输出
- ☐ C. for、while、if
- ☐ D. 复合语句、基本语句、空语句

2-6 如果list1 = [1, 2, 3, 4, 5, 4, 3, 2, 1], 那么以下那个是list1[:-1]? (1分)

- ☐ A. 0
- ☐ B. [1, 2, 3, 4, 5, 4, 3, 2, 1]
- ☒ C. [1, 2, 3, 4, 5, 4, 3, 2]
- ☐ D. [0, 1, 2, 3, 4, 3, 2, 1, 0]

2-7 以下哪个方式可以创建一个空的集合? (1分)

- ☒ A. set()
- ☐ B. {}
- ☐ C. []
- ☐ D. ()

2-8 可以使用___运算符来确定一个键是否在字典中。 (1分)

- ☐ A. &
- ☒ B. in
- ☐ C. ^替换为正确项
- ☐ D. ?

2-9 对于集合s, 以下哪个操作是不存在的? (1分)

- ☐ A. len(s)
- ☒ B. s.append(1)
- ☐ C. max(s)
- ☐ D. s - {1}

2-10 list("abcd")的结果是? (1分)

- ☒ A. ['a', 'b', 'c', 'd']
- ☐ B. ['ab']

☐ C. ['cd']

☐ D. ['abcd']

2-11 若有列表numbers =[1, 2, 3, 4, 5]

(1分)

以下哪条语句不能正确输出列表值[1, 2, 3, 4, 5].

☐ A. print(numbers)

☐ B. print(numbers[:])

☐ C. print(numbers[0:len(numbers)])

☒ D. print(numbers[0:len(numbers)-1])

2-12

```
a = ['a', 'b', 'c', 'd', 'e']  
print(a[-3:-1])
```

(1分)

执行以上语句后输出:

☒ A. ['c', 'd']

☐ B. ['c', 'd', 'e']

☐ C. ['b', 'c', 'd']

☐ D. ['d', 'e']

2-13 lst = []

(1分)

使用以下 () 语句可以使得列表的值为: ['abcd']

☐ A. lst[0] = 'abcd'

☐ B. lst = list('abcd')

☐ C. lst.insert('abcd',0)

☒ D. lst.append('abcd')

2-14 a = [1,2,3]

(1分)

b = a[:]

b[0] = 4

print(a)

☐ A. [4, 2, 3]

☒ B. [1, 2, 3]

☐ C. [1]

☐ D. [4]

2-15 以下关于Python列表的描述中，错误的是：

(1分)

- ☒ A. 列表的长度和内容都可以改变，但元素类型必须相同
- ☐ B. 可以对列表进行成员关系操作、长度计算和分片
- ☐ C. 列表可以同时使用正向递增序号和反向递减序号进行索引
- ☐ D. D. 可以使用比较操作符（如>或<等）对列表进行比较

2-16 以下关于字符串类型的操作的描述，错误的是：

(1分)

- ☐ A. str.split(x,y)方法以x为分隔符将str分割y次
- ☐ B. 想把一个字符串str所有的字符都大写，用str.upper()
- ☒ C. 想获取字符串str的长度，用字符串处理函数 str.len()
- ☐ D. 设 x = 'aa'，则执行x*3的结果是'aaaaaa'

2-17 若有字符串 s = "abcdefg"，以下描述不正确的是：

(1分)

- ☐ A. 字符串的长度是7
- ☐ B. s[0]和s[-7]返回的是同一个字符
- ☐ C. s[x]和s[-(len(s)-x)]返回的是同一个字符
- ☒ D. 字符串s的结束标记是"\0"

2-18 以下关于函数定义的描述错误的是：

(1分)

- ☒ A. 函数必须有返回值
- ☐ B. 函数使用return给出返回值
- ☐ C. 函数定义中可以指定无限多个参数
- ☐ D. 函数定义中可以为某些参数指定默认值

2-19 以下关于函数作用的描述错误的是：

(1分)

- ☒ A. 提高代码执行速度
- ☐ B. 复用代码
- ☐ C. 增强代码的可读性
- ☐ D. 降低编程复杂度

2-20 导入模块的方式错误的是

(1分)

- ☐ A. import mo
- ☐ B. from mo import *
- ☒ C. import m from mo
- ☐ D. import mo as m

2-21 在Python中，关于全局变量和局部变量，以下选项中描述不正确的是。

(1分)

- ☐ A. 一个程序中的变量可以包含两类：全局变量和局部变量
- ☐ B. 全局变量一般没有缩进
- ☐ C. 全局变量在程序执行的全过程有效
- ☒ D. 全局变量不能和局部变量重名

2-22 当try语句中没有任何错误信息时，一定不会执行（ ）语句。

(1分)

- ☐ A. try
- ☐ B. else
- ☐ C. finally
- ☒ D. except

2-23 以下关于文件的描述中，正确的是（ ）。

(1分)

- ☐ A. 使用open()打开文件时，必须要用r或w指定打开方式，不能省略。
- ☒ B. 采用readlines()可以读入文件中的全部文本，返回一个列表。
- ☐ C. 文件打开后，可以用write()控制对文件内容的读写位置。
- ☐ D. 如果没有采用close()关闭文件，Python程序退出时文件将不会自动关闭。

2-24 下面不属于Python文件操作方法的是（ ）。

(1分)

- ☐ A. read()
- ☐ B. write()
- ☒ C. join()
- ☐ D. readline()

函数题

得分：暂无 总分：60

6-1 输出闰年 (10分)

编写程序，输出1900~2020年中所有的闰年，每行输出5个年份。要求定义一个函数isLeap()，该函数用来判断某年是否为闰年，是闰年则函数返回True，否则返回False。

函数接口定义：

```
1 | 在这里描述函数接口。例如：  
2 | def isLeap(y):
```

函数参数y表示年份，函数要返回True或False。

裁判测试程序样例：

```
1 | /* 请在这里填写答案 */  
2 | n=0  
3 | for year in range(1900,2021):  
4 |     if isLeap(year)==True:  
5 |         print(year,end=' ')  
6 |         n+=1  
7 |         if n%5==0:print()
```

输入样例：

在这里给出一组输入。例如：

输出样例：

在这里给出相应的输出。例如：

```
1904 1908 1912 1916 1920  
1924 1928 1932 1936 1940  
1944 1948 1952 1956 1960  
1964 1968 1972 1976 1980  
1984 1988 1992 1996 2000  
2004 2008 2012 2016 2020
```

6-2 计算Fibonacci数（10分）

实现一个计算Fibonacci数的简单函数，并利用其实现：输入一个正整数n（ $0 < n \leq 100000$ ），输出小于n的所有Fibonacci数。

所谓Fibonacci数列就是满足任一项数字是前两项的和（最开始两项均定义为1）的数列，

例如：1,1,2,3,5,8,13,21,...

函数fib(n)用列表返回[1, n]中的所有Fibonacci数。

函数接口定义：

```
1 | def fib(n)
```

其中m, n是用户传入的参数。

裁判测试程序样例：

```
1  /* 请在这里填写答案 */
2
3  n=int(input())
4  fiblist=fib(n)
5  print(fiblist)
```

输入样例：

20

输出样例：

[1, 1, 2, 3, 5, 8, 13]

6-3 判断素数函数 (10分)

写一个函数isPrime(n)用于判断一个数字n是不是素数，用户输入一个正整数，在一行内输出不大于该数的所有素数，各数后面用一个空格分隔。

函数接口定义：

```
1 def isPrime(n):
```

裁判测试程序样例：

```
1
2  /* 请在这里填写答案 */
3  num = int(input())          #接收用户输入并转成整数
4  for i in range(num+1):
5      if isPrime(i):
6          print(i,end=' ')    #在同一行内输出结果，不换行，中间用空格分隔
```

输入样例：

在这里给出一组输入。例如：

100

输出样例：

在这里给出相应的输出。例如：

2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

6-4 凯撒加密解密 (10分)

本题要求实现一个函数，实现对凯撒加密的密文进行解密。即输入为一个密文和加密参数，输出为解密后的原文。

函数接口定义：


```
def Decrypt( encryptedText, numToMove);
```

`encryptedText` 为等待解密的文本，`numToMove` 是凯撒加密的参数，即移动位数。

裁判测试程序样例：

```
1  在这里给出函数被调用进行测试的例子。例如：
2  text = input('请输入密文：')
3  num = eval(input("请输入加密的移动位数："))
4  orgText = Decrypt(text, num)
5  print(orgText)
6  /* 请在这里填写答案 */
```

输入样例：

在这里给出一组输入。例如：

```
mjqqt btwqi
5
```

输出样例：

在这里给出相应的输出。例如：

```
hello world
```

6-5 缩写词 (10分)

缩写词是由一个短语中每个单词的第一个字母组成，均为大写。例如，CPU是短语“central processing unit”的缩写。

函数接口定义：

```
acronym(phrase);
phrase是短语参数，返回短语的缩写词
```

裁判测试程序样例：

```
/* 请在这里填写答案 */
```

```
phrase=input()
print(acronym(phrase))
```

输入样例：

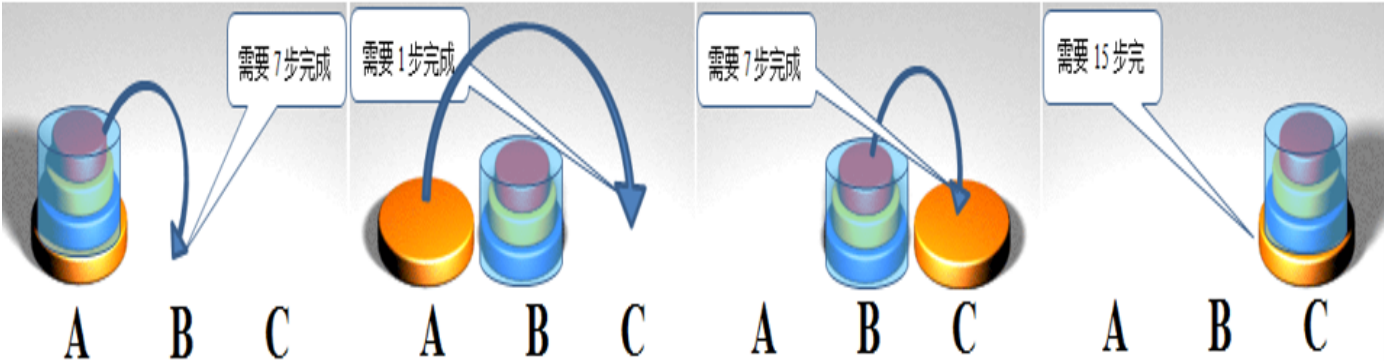
```
central processing unit
```

输出样例：

```
CPU
```

6-6 汉诺塔问题 (10分)

汉诺塔问题求解：有三根相邻的柱子，假设标号分别为A、B、C，其中A柱子从下到上按金字塔状依次叠放了N个不同大小的圆盘，现要把A柱子上的所有圆盘一次一个地移动到C柱子上，移动的过程中可以借助B柱子做中转，并且每根柱子上的圆盘必须始终保持上小下大的叠放顺序。编写一个函数，输出移动轨迹（提示：用递归函数）



函数接口定义：

- 1 函数接口：
- 2 hanoi(n,a,b,c),print移动轨迹并统计移动次数
- 3 n表示要移动的圆盘个数，a,b,c为三根柱子名称，表示将盘子从a移动到c,b为中转柱。

裁判测试程序样例：

- 1 测试样例：
- 2 n=int(input())
- 3 step=0
- 4 hanoi(n,'A','B','C')
- 5 print('共移动次数:',step)

输入样例1：

输入n (1<=n<=5)：

1

输出样例1：

在这里给出相应的输出。例如：

A -> C
共移动次数：1

输入样例2：

输入n (1<=n<=5)：

3

输出样例2：

在这里给出相应的输出。例如：

```
A -> C
A -> B
C -> B
A -> C
B -> A
B -> C
A -> C
共移动次数：7
```

编程题

得分：暂无 总分：170

7-1 输入学生姓名，输出问候信息。（10分）

输入学生姓名name，输出问候信息"Hello,name"。

输入格式：

在一行中给出学生姓名 name。

输出格式：

在一行中输出：Hello,name!

输入样例：

在这里给出一组输入。例如：

Christ

输出样例：

在这里给出相应的输出。例如：

Hello,Christ!

7-2 成绩是否及格（10分）

输入成绩值 $x \in [0, 100]$ ，如果 $x \geq 60$ ，则输出pass，否则输出fail。

输入格式：

直接输入 $[0, 100]$ 之间的1个整数，没有其它任何附加字符。

输出格式：

直接输出你“pass”或“fail”，没有其它任何附加字符。

输入样例：

78

输出样例：

pass

输入样例:

```
46
```

输出样例:

```
fail
```

7-3 五级制成绩-zzuli (10分)

给定一个百分制成绩, 请根据百分制成绩输出其对应的等级。转换关系如下:
90分及以上为'A', 80~89为'B', 70~79为'C', 60~69为'D', 60分以下为'E'。

输入格式:

一个百分制成绩 (0~100的整数)。

输出格式:

输出对应的等级。

输入样例:

```
100
```

输出样例:

```
A
```

7-4 月份与天数 (10分)

输入公历的年份以及月份, 输出这个月的天数。按照公历的历法, 每年的1、3、5、7、8、10、12月份有31天, 4、6、9、11月份有30天, 平年的2月份有28天, 闰年的2月份有29天

输入格式:

输入的数据格式是固定的YYYYMM的形式, 即: 年份占4个数位, 月份占2个数位, 且没有不合理的输入, 如2017!@#\$%13^&*之类的形式。

输出格式:

要求输出月份的天数

输入样例:

```
201802
```

输出样例:

```
28
```

7-5 学号 (10分)

ZWU每年招收的本科生人数大约6000左右，每位新生入学时，都被分配了不同的学号。学号由十位数字组成，前四位表示入学年份即年级，中间两位数字01表示学生的类别为本科生（注:ZWU也招收专升本、第二学位以及硕士研究生，会使用其他编码区分学生类别），最后四位数字代表录取序号。

例如：2023012023表示2023年入学的本科生，录取序号为2023

要求：输入正整数 $N(N \leq 6000)$ ，输出2023级本科生的学号（前 N 个）

输入格式:

输入正整数 $N(N \leq 6000)$

输出格式:

输出2023级本科生的学号（前 N 个），每个学号占一行

输入样例:

3

输出样例:

```
2023010001
2023010002
2023010003
```

7-6 打印倒直角三角形图案 (10分)

打印一个由 m 行字符 * 组成的倒置的直角三角形图案。

输入格式:

输入一个整数，表示行数 m ，其中 m 满足 $2 \leq m \leq 10$ 。

输出格式:

由字符 * 组成的 m 行倒立直角三角形图案。

输入样例:

5

输出样例:

```
*****
*****
***
**
*
```

7-7 jmu_python_最大公约数&最小公倍数 (10分)

本题要求从键盘输入两个整数（以逗号间隔），编程求出这两个数的最大公约数和最小公倍数

提示：求最大公约数可用辗转相除法，最小公倍数用两数的积除以最大公约数

输入格式:

在一行中输入两个整数，以逗号间隔

输出格式:

输出"GCD:a, LCM:b", 其中a为求出的最大公约数，b为求出的最小公倍数

注意：在逗号后面有个空格

输入样例:

```
12,14
```

输出样例:

```
GCD:2, LCM:84
```

7-8 百钱百鸡 (10分)

百鸡问题是北魏数学家张丘建在其著作《张丘建算经》中提出的一个世界著名的不定方程问题：“今有鸡翁一，值钱五；鸡母一，值钱三；鸡雏三，值钱一。凡百钱买鸡百只，问鸡翁母雏各几何。”

百钱百鸡问题的白话版：100元钱买100只鸡，公鸡5元1只，母鸡3元1只，小鸡1元3只。问公鸡、母鸡、小鸡各多少只（某种鸡可以为0只）？

百钱百鸡的结果如输出样例所示。

现在把100改为n，即n元钱买n只鸡，各种鸡价格不变，结果又如何呢？

测试数据保证至少存在一组解。

输入格式:

测试数据有多组，处理到文件尾。每组测试输入一个整数n（ $100 \leq n \leq 1000$ ）。

输出格式:

对于每组测试，按公鸡、母鸡、小鸡的数量（按公鸡数从小到大的顺序）逐行输出各种买法（每行数据之间空一个空格）。

输入样例:

```
100
```

输出样例:

```
0 25 75
4 18 78
```

```
8 11 81
12 4 84
```

7-9 查找学号 (10分)

ZWU每年招收的本科生人数大约6000左右，每位新生入学时，都被分配了不同的学号。学号由十位数字组成，前四位表示入学年份即年级，中间两位数字01表示学生的类别为本科生（注:ZWU也招收专升本、交换生以及硕士研究生，会使用其他编码区分学生类别），最后四位数字代表录取序号。

例如：2022015021表示2022年入学的本科生，录取序号为5021

输入一组数据（整数，不多于20个），查找指定的数据项是否存在。如果存在，输出第一次出现的位置；如果不存在，输出No

输入格式:

输入共有3行。

第1行是正整数 n ($n \leq 100$),

第2行是 n 个学号（int范围内的整数），学号之间用空格分隔。

第3行是1个学号（int范围内的整数），表示要查找的学号。

输出格式:

如果查找的学号存在，输出第一次出现的位置；如果不存在，输出No

输入样例1:

```
3
2022015021 2022015098 2022011999
2022015021
```

输出样例1:

```
1
```

输入样例2:

```
5
2022015021 2022015098 2022011999 2022010337 2022012911
2022010199
```

输出样例2:

```
No
```

输入样例3:

```
5
2022015021 2022015098 2022011999 2022010337 2022015098
2022015098
```

输出样例3:

```
2
```

7-10 成绩统计 (10分)

现有某班级Python考试成绩，请你帮老师做一下成绩统计，找出最高分、最低分，算出平均分，统计出不及格人数。

输入格式:

输入在第一行给出班级人数n，第二行为n个学生的成绩（0~100之间的整数），中间用空格分隔。

输出格式:

输出共四行，分别为最高分、最低分、平均分和不及格人数，其中平均分保留1位小数。

输入样例:

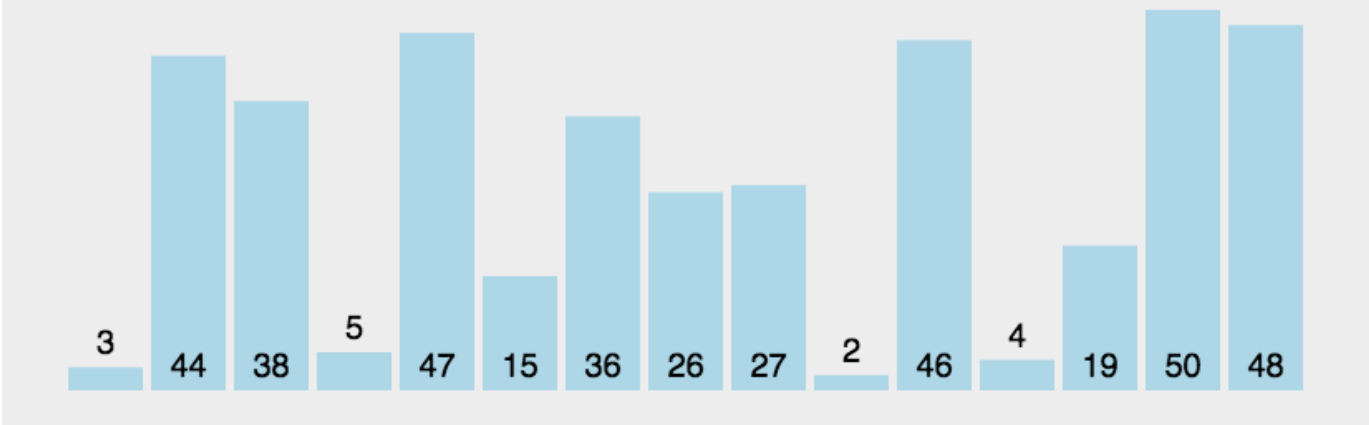
```
10
80 95 75 85 90 55 66 88 100 58
```

输出样例:

```
Max: 100
Min: 55
Ave: 79.2
Fail: 2
```

7-11 冒泡排序 (10分)

冒泡排序，将一个列表中的两个元素进行比较，并将最小的元素交换到顶部。两个元素中较小的会冒到顶部，而较大的会沉到底部，该过程将被重复执行，直到所有元素都被排序。



输入格式:

输入在第1行中给出N（ $1 < N \leq 100$ ），在第2行中给出N个待排序的整数，数字间以空格分隔，并保证数字没有重复的出现。

输出格式:

给出冒泡排序每一遍后的中间结果数列，数字间以空格分隔，但末尾不得有多余空格。**注意：当排序完成时应立即停止。**

输入样例1:

```
7
4 5 7 6 3 2 1
```


输出样例1:

```
4 5 6 3 2 1 7
4 5 3 2 1 6 7
4 3 2 1 5 6 7
3 2 1 4 5 6 7
2 1 3 4 5 6 7
1 2 3 4 5 6 7
```

输入样例2:

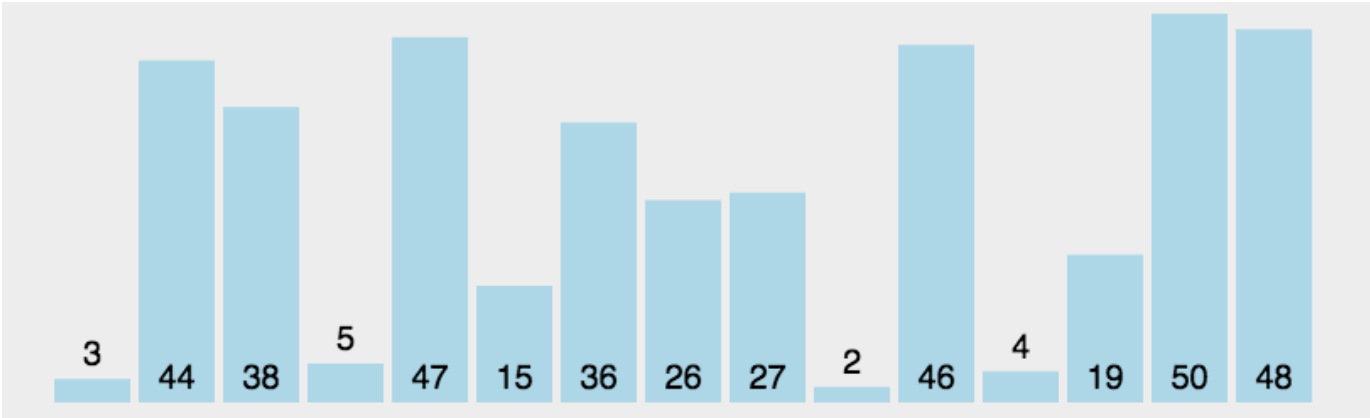
```
6
1 2 3 6 5 4
```

输出样例2:

```
1 2 3 5 4 6
1 2 3 4 5 6
```

7-12 选择排序（10分）

选择排序，从头至尾扫描序列,找出最小的一个元素，和第一个元素交换,接着从剩下的元素中继续这种选择和交换方式，最终得到一个有序序列。



输入格式:

输入在第1行中给出N（ $1 < N \leq 100$ ），在第2行中给出N个待排序的整数，数字间以空格分隔，并保证数字没有重复的出现。

输出格式:

给出选择排序每一遍后的中间结果数列，数字间以空格分隔，但末尾不得有多余空格。**注意：当排序完成时应立即停止。**

输入样例1:

```
7
4 5 7 6 3 2 1
```

输出样例1:

```
1 5 7 6 3 2 4
1 2 7 6 3 5 4
1 2 3 6 7 5 4
```

```
1 2 3 4 7 5 6
1 2 3 4 5 7 6
1 2 3 4 5 6 7
```

输入样例2:

```
5
1 2 3 5 4
```

输出样例2:

```
1 2 3 4 5
```

7-13 杨辉三角 (20分)

还记得中学时候学过的杨辉三角吗？具体的定义这里不再描述，你可以参考以下的图形：

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```

输入格式:

输入数据包含多个测试实例，每个测试实例的输入只包含一个正整数 n ($1 \leq n \leq 30$)，表示将要输出的杨辉三角的层数。

输出格式:

对应于每一个输入，请输出相应层数的杨辉三角，每一层的整数之间用一个空格隔开，每一个杨辉三角后面加一个空行。

输入样例:

在这里给出一组输入。例如：

```
2
3
```

输出样例:

在这里给出相应的输出。例如：

```
1
1 1

1
1 1
1 2 1
```

7-14 判断回文串 (10分)

若一个串正向看和反向看等价，则称做回文串。例如：t, abba, xyzyx均是回文串。
给出一个长度不超过60的字符串，判断是否是回文串。

输入格式:

首先输入一个正整数T，表示测试数据的组数，然后是T组测试数据。每行输入一个长度不超过60的字符串（串中不包含空格）。

输出格式:

对于每组测试数据，判断是否是回文串，若是输出 **Yes**，否则输出 **No**。

输入样例:

```
2
abba
abc
```

输出样例:

```
Yes
No
```

7-15 统计字符串中不同种类的字符个数（10分）

本题目要求读入一个字符串，统计字符串中字母、数字、空格、其它字符的个数。

输入格式:

输入一行由任意字符组成的字符串。

输出格式:

统计字符串中字母、数字、空格、其它字符的个数。

输入样例:

在这里给出一组输入。例如：

```
2a and Am3,MNak888!..
```

输出样例:

在这里给出相应的输出。例如：

```
letters=10,digits=5,spaces=2,others=4
```

7-16 字典合并（10分）

输入用字符串表示两个字典，输出合并后的字典。字典的键用一个字母或数字表示。注意：1和'1'是不同的关键字！

输入格式:

在第一行中输入第一个字典字符串；

在第二行中输入第二个字典字符串。

输出格式:

在一行中输出合并的字典，输出按字典序。

"1" 的 ASCII 码为 49，大于 1，排序时 1 在前，"1" 在后。其它的字符同理。

输入样例1:

在这里给出一组输入。例如：

```
{1:3,2:5}  
{1:5,3:7}
```

输出样例1:

在这里给出相应的输出。例如：

```
{1:8,2:5,3:7}
```

输入样例2:

在这里给出一组输入。例如：

```
{"1":3,1:4}  
{"a":5,"1":6}
```

输出样例2:

在这里给出相应的输出。例如：

```
{1:4,"1":9,"a":5}
```

鸣谢用户 卧龙几两钱 补充数据！