

2017 年《程序设计导论》课程期中测试

2017—2018 学年第 1 学期

班级 _____ 姓名 _____ 学号 _____

注意事项

- 1. 本次测试的时间为 180 分钟；编程结果采用机器自动评测。
- 2. 共有 5 题，第 1、2 题，25 分；第 3 题，20 分；第 4、5 题，15 分。
- 3. 提交到在线评测系统中的程序均采用标准输入和标准输出（键盘输入和屏幕输出）。
- 4. 程序设计语言选用 C 或 C++。
- 5. 所有题目的时间限制均为 1s。

一、分段函数（25 分）

得分	评卷人

【问题描述】

已知有如下分段离散函数 f(x)，其中自变量 x 为[-1000, 1000]之间的实数。

$$f(x) = \begin{cases} |x - 1| - 2 & (|x| \leq 1); \\ \frac{1}{1 + x^2} & (|x| > 1); \end{cases}$$

请编写一个程序计算函数 f(f(x)) 的值。注意，当函数值有小数时四舍五入保留 2 位小数。

【输入格式】

输入一个实数 x。

【输出格式】

输出 f(f(x))函数值，当函数值有小数时四舍五入保留 2 位小数。

【输入样例 1】

0.5

【输入样例 2】

-3

【输出样例 1】

0.31

【输出样例 2】

-1.10

【注意】

浮点数计算过程中请使用 double 类型。

二、外侧元素求和（25 分）

得分	评卷人

【问题描述】

求一个 n×m 矩阵 [a_{ij}] 的所有靠外侧的元素值之和。外侧元素是指，矩阵四周外侧的元素，即首行和末行，及首列和末列的所有元素（注意每个元素只算一遍，不要重复计算）。取值范围：1≤n, m≤100；矩阵元素为整数 -99,999≤a_{ij}≤99,999。

【输入格式】

输入分为 n+1 行。

第一行为 n 和 m，n 为矩阵行数，m 为列数，之间由空格分隔；第二行至 n+1 行为矩阵的元素，按行排列矩阵的元素，数据之间由空格分隔，行以回车结束。

【输出格式】

输出一个整数，为求和结果。

【输入样例 1】

3 4
3 8 9 10
2 5 -3 5
7 0 -1 4

【输入样例 2】

1 1
100

【输出样例 2】

100

【输出样例 1】

47

三、共享单车停放（20 分）

得分	评卷人

【问题描述】

城市里共享单车越来越多，给人们带来方便的同时，单车的乱停放问题也给管理者与城市交通带来了一定困扰。共享单车定点停放势在必行，于是对共享单车 App 产生了新的功能需求：为用车人推荐一个目的地附近的停车区，保证用车人到达后有单车停车位且步行到目的地的距离最短。

现在，小明同学要从人民大学东门骑车到国家图书馆，为了给他推荐一个符合要求的停车区，手机 App 需得到如下表所示的信息：

国图阅览室 1000 米范围内可停车区的数量：	5				
停车区编号：	停车区 1	停车区 2	停车区 3	停车区 4	停车区 5
国图阅览室到各停车区的距离：	80	90	100	150	160
各停车区当前的剩余车位数：	20	5	10	0	15
停车区空余车位变化速度（个/每 10 分钟）：	-7	-3	-5	2	-5
从人大东门骑到停车区的预估时间（分钟）：	30	20	20	20	20

表中空余车位变化速度为正数时，表示被骑走的车比停入的车多，直到车位全空（不必在意全空时车位具体有多少）；反之空余车位变化速度为负数时，停入的车比骑走的车多，直到全满，剩余车位数为 0。这里有些数据（如车位变化速度和预估的骑行时间）来源于物联网和大数据技术的监测和分析。为了简化计算，假定剩余车位变化速度在小明骑行期间不变。

根据这些信息，手机 App 可引导小明直接骑到 4 号区所在的位置。
现根据 App 上述需求，当小明要骑车去别的地方时，为其推荐一个停车区的位置（编号，从 1 开始）。

【输入格式】

共 5 行，第 1 行，1 个整数 **n**，表示单车停放区数量。
之后的 4 行，**每行 n 个整数**。其含义如上表所示，分别为目的地点到各单车区的距离，各停车区的剩余车位数，空余车位变化速度，以及小明骑到停车区的预估时间。各行数字的次序按停车区一一对应，数字之间均为空格分隔。

【输出格式】

1 行仅 1 个整数，表示推荐的停车区的编号。如果找不到合适停车区，输出 0。

【数据说明】

任意目的地周边停靠点最多不超过 50 个，且距离各不相同；更新剩余车位时，使用 round 库函数（math.h）对变化数进行四舍五入取整。

【输入样例】

5
80 90 100 150 160
20 5 10 0 15
-7 -3 -5 2 -5
30 20 20 20 20

【输出样例】

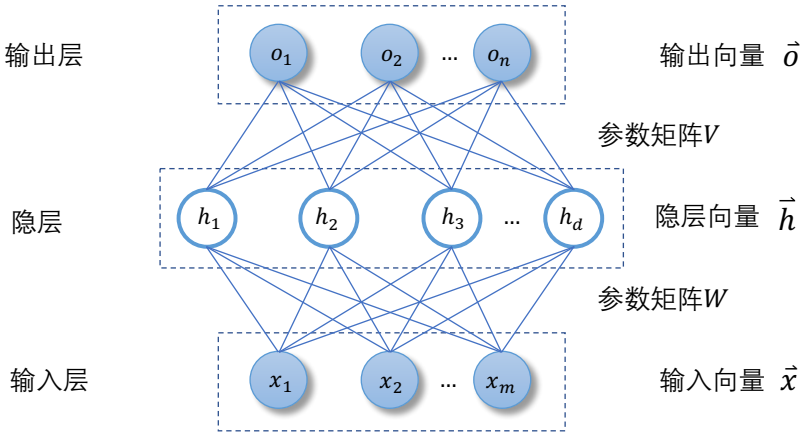
4

四、深度神经网络（15 分）

得分	评卷人

【问题描述】

美国谷歌公司开发的 Alpha Go 围棋程序接连战胜李世石、柯洁等人类顶尖棋手，其成功秘诀就在于使用了深度神经网络。现请你编写程序实现深度神经网络的一种最基本模型——多层感知器模型（Multi-Layer Perceptron, MLP）。



一个最基本的多层感知器模型（单隐层）如上图所示，它通过一系列非线性变换将输入层的列向量 $\vec{x} = (x_1, x_2, \dots, x_m)^T$ ，变换成隐层中的向量 $\vec{h} = (h_1, h_2, \dots, h_d)^T$ ，进而变换成输出层的列向量 $\vec{o} = (o_1, o_2, \dots, o_n)^T$ 。注：图中的虚线方框表示一个向量，内部的圆圈表示向量的数值，连线表示向量的变换。

具体来讲，将输入向量 \vec{x} 变换为隐向量 \vec{h} ，计算公式的矩阵表达形式为：

$$\vec{h} = s(W\vec{x})$$

其中 W 为 $d \times m$ 维的参数矩阵， $W\vec{x}$ 表示矩阵 W 与列向量 \vec{x} 的乘积。 s 表示向量的函数变换，该函数对向量中的每一个维度数值 a 进行计算变换，常用的一种变换是： $s(a) = 1/(1 + e^{-a})$ 。所有维度的计算结果就构成所需的向量 \vec{h} 。例如：向量 $(0,0)^T$ 经过 s 函数计算后的结果为 $(0.5, 0.5)^T$ 。

与此类似，将向量 \vec{h} 变换为输出层的列向量 \vec{o} ，计算公式的矩阵表达形式为：

$$\vec{o} = s(V\vec{h})$$

其中 V 为 $n \times d$ 维的参数矩阵， $V\vec{h}$ 表示矩阵 V 与列向量 \vec{h} 的乘积。 s 同样是向量上的函数变换，含意同上。
现给你提供输入向量 \vec{x} 、参数矩阵 W 和 V ，请你计算输出向量 \vec{o} 。注意，输出结果保留两位小数。

【输入格式】

第 1 行: 3 个整数, 分别表示输入向量 \vec{x} 的维度 m , 隐向量 \vec{h} 的维度 d , 以及输出向量 \vec{o} 的维度 n ; 第 2 行: 输入向量 \vec{x} , 共 m 个整数; 第 3 行: 参数矩阵 W 中的 $d \times m$ 个整数, 其次序与该矩阵中数字按行 (首尾相连) 次序一致; 第 4 行: 参数矩阵 V 中的 $n \times d$ 个整数, 其排列方式同前。数据之间都以空格分隔。

【输出格式】

输出一行, 为计算结果 \vec{o} 向量, 每个维度数值保留两位小数, 之间以空格分隔

【数据规模说明】

- 1. 题目中的向量维度不超过 16、矩阵行与列的维度均不超过 16;
 - 2. 向量和矩阵中的数字均在 int 类型表数范围内。
- 【提示信息】
- 1. 数值计算请使用 double 数据类型;
 - 2. 指数函数请使用 math.h 头文件中的 double exp(double x) 函数。
 - 3. 矩阵乘以列向量的公式: 以前面的 $W\vec{x}$ 为例, 结果为 d 维的列向量, 可以表示为 $(\sum_{i=1}^m w_{1i}x_i, \sum_{i=1}^m w_{2i}x_i, \dots, \sum_{i=1}^m w_{di}x_i)^T$

【输入样例 1】

1 1 1
1
2
-2

【输出样例 1】

0.15

【输入样例 2】

3 4 2
0 0 1
-1 1 0 -1 2 0 0 -2 2 1 0 1
1 -2 1 -2 0 -2 2 2

【输出样例 2】

0.25 0.90

五、班会时间 (15 分)

得分	评卷人

【问题描述】

由于班上同学所选课程和上课时段各不相同。为了找到一个合适的时间开班会, 班长小 A 收集了班上所有同学的课表。
请你编写程序, 帮助小 A 处理所有同学的课表, 并得到开班会的合适时间段,

时间段按照中国人民大学的课表划分 (每个时间段的表示法如下表所示)。要求程序根据输入的所有课表, 统计出每个时间段有课的学生人数, 并将时间段按照上课人数从小到大排序, 如果某两个时间段的上课人数一样, 则按照时间段的先后顺序排序 (时间在前的排在前面), 输出有课人数前 k ($1 \leq k \leq 49$) 少的时间段。

【输入格式】

第一行包含 2 个正整数 n 和 k , 分别表示班级的人数 n 和题目中要求的 k 。
接下来的 n 行, 每行表示一个学生的课表, 具体格式如下:
2017101234 10 2.1 1.2 4.3 2.4 2.5 5.1 5.2 1.6 4.2 4.4

节次	时间	周一	周二	周三	周四	周五	周六	周日
1	8:00-9:30	1.1	2.1	3.1	4.1	5.1	6.1	7.1
2	10:00-11:30	1.2	2.2	3.2	4.2	5.2	6.2	7.2
3	12:00-13:30	1.3	2.3	3.3	4.3	5.3	6.3	7.3
4	14:00-15:30	1.4	2.4	3.4	4.4	5.4	6.4	7.4
5	16:00-17:30	1.5	2.5	3.5	4.5	5.5	6.5	7.5
6	18:00-19:30	1.6	2.6	3.6	4.6	5.6	6.6	7.6
7	19:40-21:10	1.7	2.7	3.7	4.7	5.7	6.7	7.7

其中第 1 个十位整数 m 表示这个学生的学号, 第 2 个正整数 p 表示这个学生有 p 个时间段有课, 之后跟着 p 个小数表示有课的时间段, 例如 2.1 表示周二第 1 节有课。每 2 个数字之间用一个空格隔开。

【输出格式】

输出 k 行, 每行包含一个小数和一个整数, 分别表示时间段和该时间段有课的学生人数, 按照有课人数从小到大的顺序输出前 k 个, 如果某两个时间段的上课人数一样, 则时间在前面的先输出。

【输入样例】

3 5
2017101000 10 2.1 1.2 4.3 2.4 2.5 5.1 5.2 1.6 4.2 4.4
2017101001 9 2.1 2.2 2.5 3.2 1.1 1.3 4.3 4.4 5.2
2017101002 10 1.4 1.5 2.1 2.2 3.4 3.4 4.1 4.6 5.4 5.5

【输出样例】

1.7 0
2.3 0
2.6 0
2.7 0
3.1 0

【数据规模说明】

40% 的数据, $k=1$;
100% 的数据, $1 \leq n \leq 1000, 1 \leq k \leq 49, 1 \leq p \leq 49$, 输入数据一个学生在同一个时间段至多只有 1 门课。