

## 实验题

### 一、基本要求：

- 1) 程序必须可运行（编译无错，可运行），否则无成绩
- 2) 利用测试文件（input file）测试，测试文件格式说明见每道题目的要求。生成相应的结果文件（output file）。
- 3) 测试包括自测和统测，自测是检查时自行编制测试数据，然后运行程序进行测试，自测通过可以得一半的分。统测是上交源代码后统一测试，统测通过得另一半的分。
- 4) 杜绝抄袭，如发现抄袭，取消成绩并严肃批评。
- 5) 提交：

仅提交源代码和自测数据文件，不写实验报告

时间：每次实验课后两天内，即：第一次实验 11 月 1 日 23:59 之前，第二次实验 11 月 15 日 23:59 之前，逾期没有成绩。

方式：电子档，在硬盘上建一个以自己学号命名的目录，下面再建名为 1、2、3、4 的子目录，将自己 1-4 道题的源程序和数据文件分别拷到相应目录下，压缩打包，压缩文件命名为：学号.zip 学委统一收齐后，与截止时间之前发：1097412466@qq.com

注：因命名错误可导致检查失败，后果自负。

### 二、题目

共 4 道，第一次实验（10.30）做 1-2 题，第二次实验（11.13）做 3-4 题。

#### 1. 最近点对问题的算法实现

##### 1) 测试文件：

Input file: in.dat

格式：第一行为一个整数，表示测试用例的组数，其后跟相应组数的测试用例  
每组测试用例包括：

首行：一个整数，表示本组测试用例包含的点数，其后跟相应点数的行

其后：每行两个整数，表示该点的 x、y 坐标

Output file: out.dat

格式：每行输出一个测试用例的答案，即本组测试用例中相距最近的两个点，用点的坐标表示：四个整数。前两个整数表示第一个点，后两个整数表示第二个点。若有多对相距最近的点，依次罗列。

##### 2) 评分：2

自测通过：1 分

统测通过：1 分

抄袭：-2 分

#### 2. 利用分治法设计一个计算两个 n 位的大整数相乘的算法，要求计算时间低于 $O(n^2)$ 。

大整数（big integer）：位数很多的整数，普通的计算机不能直接处理，如：

9834975972130802345791023498570345

对大整数的算术运算，显然常规程序语言是无法直接表示的。编程实现大整数的加、减、乘运算，需考虑操作数为 0、负数、任意位等各种情况

##### 1) 测试文件：

Input file in.dat

格式：第一行为一个整数，表示测试用例的组数，其后跟相应组数的测试用例  
每个测试用例一行，包含 3 个整数（长整数数字串），前两个是待测试的

操作数，第 3 个整数表示操作类型（1：加法，2：减法，3：乘法）。

Output file: out.dat

格式：每组测试用例输出一行，最后不要加空行

2) 评分：2

自测通过：1 分

统测通过：1 分

抄袭：-2 分

### 3. 15.5-1 最优二分查找树

#### 4. 每对结点之间的最短路径(Floyd-Warshall 算法)

补充 ALL-PATHS 算法，增加前驱矩阵(Chp.25.2)，使得在求出结点间的最短路径长度矩阵 A 后，能够推导出每对结点间的最短路径。

1) 测试文件：

Input file: in.dat

格式：第一行为一个整数，表示测试用例的组数，其后跟相应组数的测试用例  
每组测试用例包括：

首行：一个整数，表示本组测试用例包含的结点数  $n$ ，其后跟  $n$  行

其后：每行  $n$  个整数，表示结点间邻接关系及边的长度（邻接成本矩阵）  
边的长度  $< 100$ ，100 即表示结点间没有边。

Output file: out.dat

格式：第一行为一个整数，表示测试用例的组数，其后跟相应组数的测试用例  
每组测试用例输出包括：

首行：一个整数，表示本组测试用例包含的结点数  $n$ ，其后跟  $n+n^2$  行

其后：开始的  $n$  行，每行  $n$  个整数，表示结点间最短路径的长度（A 矩阵）。路径的程度  $< 32767$ ，32767 即表示结点间没有可达的路径。

其后  $n^2$  行，顺次输出结点对  $(1,1)$ 、 $(1,2)$ 、...、 $(1,n)$ ， $(2,1)$ 、 $(2,2)$ 、...、 $(2,n)$ 、...、 $(n,1)$ 、...、 $(n,n)$  之间的最短路径结点序列，结点间用空格隔开。 $(i,i)$  输出  $i$ ，若  $(i,j)$  之间没有路径，输出 NULL。

2) 评分：2

自测通过：1 分

统测通过：1 分

抄袭：-2 分