

第一次实验 9000000001~9000000002

第二次实验 9000000003~9000000006

第三次实验 9000000007

第四次实验 9000000008~9000000011

CS 试题

| | |
|-------------------|--|
| id | 9000000001 |
| title | 实验 1.1: 合并有序数组 |
| time_limit | 3000MS |
| memory_limit | 10000KB |
| filesize_limit | 0 |
| category | PF_Simple |
| difficulty | 10 |
| description | 给定两个按照升序排列的有序数组，请把它们合成一个升序数组并输出。 |
| input | 第一行为第一个有序数组的长度，正整数 n ， $n \leq 20$ ； 第二行为第一个有序数组的 n 个数字，用空格隔开； 第三行为第二个有序数组的长度，正整数 m ， $m \leq 20$ ； 第四行为第二个有序数组的 m 个数字，用空格隔开。 |
| output | 输出合并后的数组，每个数字占一行。 |
| sample_input | 3 1 3 7 5 2 4 6 8 10 |
| sample_output | 1 2 3 4 6 7 8 10 |
| hint | |
| source | INPOJ |
| solution_language | CS |
| solution | |

CS 试题

| | |
|----------------|--------------------|
| id | 9000000002 |
| title | 实验 1.2: 高精度计算 PI 值 |
| time_limit | 3000MS |
| memory_limit | 10000KB |
| filesize_limit | 0 |
| category | PF_Simple |
| difficulty | 10 |

| | |
|-------------------|---|
| description | 限制使用双向链表作存储结构，请根据用户输入的一个整数（该整数表示精确到小数点后的位数，可能要求精确到小数点后 500 位），高精度计算 PI 值。可以利用反三角函数幂级展开式来进行计算。 |
| input | 输入的一个正整数 n |
| output | 输出 PI 的值，精确到小数点后 n 位，最后输出一个回车。 |
| sample_input | 5 |
| sample_output | 3.14159 |
| hint | |
| source | INPOJ |
| solution_language | CS |
| solution | |

CS 试题

| | |
|-------------------|--|
| id | 9000000003 |
| title | 实验 2.1：稀疏矩阵转置 |
| time_limit | 3000MS |
| memory_limit | 10000KB |
| filesize_limit | 0 |
| category | PF_Simple |
| difficulty | 10 |
| description | 输出稀疏矩阵的转置矩阵。（行列均不大于 20） |
| input | 第一行输入两个正整数 n 和 m,分别表示矩阵的行数和列数， 然后输入矩阵三元组， 最后输入（0 0 0）表示结束输入。 |
| output | 转置后的矩阵。 |
| sample_input | 4 4 1 1 1 2 1 2 3 2 3 0 0 0 |
| sample_output | 1 1 1 1 2 2 2 3 3 |
| hint | |
| source | INPOJ |
| solution_language | CS |
| solution | |

CS 试题

| | |
|----------------|--------------------------|
| id | 9000000004 |
| title | 实验 2.2：稀疏矩阵加法,实现 $C=A+B$ |
| time_limit | 3000MS |
| memory_limit | 10000KB |
| filesize_limit | 0 |

| | |
|-------------------|---|
| category | PF_Simple |
| difficulty | 10 |
| description | 输入两个稀疏矩阵，输出它们相加的结果。 |
| input | 第一行输入四个正整数，分别是两个矩阵的行 m、列 n、第一个矩阵的非零元素的个数 t1 和第二个矩阵的非零元素的个数 t2。 接下来的 t1+t2 行是三元组，分别是第一个矩阵的数据和第二个矩阵的数据。三元组的第一个元素表示行号，第二个元素表示列号，第三个元素是该项的值。 |
| output | 输出相加后的矩阵三元组。 |
| sample_input | 3 4 3 2 1 1 1 1 3 1 2 2 2 1 2 1 2 2 3 |
| sample_output | 1 1 1 1 2 1 1 3 1 2 2 5 |
| hint | |
| source | INPOJ |
| solution_language | CS |
| solution | |

CS 试题

| | |
|----------------|---|
| id | 9000000005 |
| title | 实验 2.3：稀疏矩阵加法，用十字链表实现 $C=A+B$ |
| time_limit | 3000MS |
| memory_limit | 10000KB |
| filesize_limit | 0 |
| category | PF_Simple |
| difficulty | 10 |
| description | 输入两个稀疏矩阵，输出它们相加的结果。 |
| input | 第一行输入四个正整数，分别是两个矩阵的行 m、列 n、第一个矩阵的非零元素的个数 t1 和第二个矩阵的非零元素的个数 t2。 接下来的 t1+t2 行是三元组，分别是第一个矩阵的数据和第二个矩阵的数据。三元组的第一个元素表示行号，第二个元素表示列号，第三个元素是该项的值。 |
| output | 输出相加后的矩阵三元组。 |
| sample_input | 3 4 3 2 1 1 1 1 3 1 2 2 2 1 2 1 2 2 3 |
| sample_output | 1 1 1 |

| | |
|-------------------|-------------------------|
| | 1 2 1 1 3 1 2 2 5 |
| hint | |
| source | INPOJ |
| test_input | |
| test_output | |
| solution_language | CS |
| solution | |

CS 试题

| | |
|-------------------|--|
| id | 9000000006 |
| title | 实验 2.4: 稀疏矩阵的乘法 |
| time_limit | 3000MS |
| memory_limit | 10000KB |
| filesize_limit | 0 |
| category | PF_Simple |
| difficulty | 10 |
| description | 计算两个稀疏矩阵的乘法 |
| input | 首先输入第一个矩阵的行数和列数，再输入该矩阵的三元组形式，以 0 0 0 结束。 然后输入第二个矩阵的行数和列数，再输入该矩阵的三元组形式，以 0 0 0 结束。 |
| output | 输出相加后的矩阵三元组。 |
| sample_input | 3 3 1 1 1 2 2 2 2 3 4 3 1 -4 0 0 0 3 3 1 3 -2 2 3 -5 3 1 8 3 2 -6 0 0 0 |
| sample_output | 1 3 -2 2 1 32 2 2 -24 2 3 -10 3 3 8 |
| hint | |
| source | INPOJ |
| solution_language | CS |
| solution | |

CS 试题

| | |
|-------------------|--|
| id | 9000000007 |
| title | 实验 3.1：哈夫曼编/译码器 |
| time_limit | 3000MS |
| memory_limit | 10000KB |
| filesize_limit | 0 |
| category | PF_Simple |
| difficulty | 10 |
| description | 写一个哈夫曼码的编/译码系统，要求能对要传输的报文进行编码和解码。构造哈夫曼树时，权值小的放左子树，权值大的放右子树，编码时右子树编码为 1，左子树编码为 0。 |
| input | 输入表示字符集大小为 n ($n \leq 100$) 的正整数，以及 n 个字符和 n 个权值（正整数，值越大表示该字符出现的概率越大）；输入串长小于或等于 100 的目标报文。 |
| output | 经过编码后的二进制码，占一行； 以及对应解码后的报文，占一行； 最后输出一个回车符 |
| sample_input | 5 a b c d e 12 40 15 8 25 bbbaddeccbbb |
| sample_output | 00011111110111010110110000 bbbaddeccbbb |
| hint | |
| source | INPOJ |
| solution_language | CS |
| solution | |

CS 试题

| | |
|----------------|---|
| id | 9000000008 |
| title | 实验 4.1：求赋权图中一个结点到所有结点的最短路径的长度 |
| time_limit | 3000MS |
| memory_limit | 10000KB |
| filesize_limit | 0 |
| category | PF_Simple |
| difficulty | 10 |
| description | 给一个赋权图（无向图），求 0 号结点到其余所有结点的最短路径的长度。 |
| input | 先输入一个小于等于 100 的正整数 n ，然后输入赋权图的邻接矩阵（10000 表示无穷大,并且任意一条简单路径的长度都小于 10000） |
| output | 按结点编号的顺序输出 0 号结点所有结点的最短路径的长度。 |
| sample_input | 6 0 1 4 10000 10000 10000 1 0 2 7 5 10000 4 2 0 10000 1 10000 10000 7 10000 0 3 2 10000 5 1 3 0 6 10000 10000 10000 2 6 0 |

| | |
|-------------------|----------------------------|
| sample_output | 0 1 3 7 4 9 |
| hint | |
| source | INPOJ |
| solution_language | CS |
| solution | |

CS 试题

| | |
|-------------------|--|
| id | 9000000009 |
| title | 实验 4.2：用迪杰斯特拉算法求赋权图中的最短路径 |
| time_limit | 3000MS |
| memory_limit | 10000KB |
| filesize_limit | 0 |
| category | PF_Simple |
| difficulty | 10 |
| description | 用迪杰斯特拉算法求一点到其余所有结点的最短路径。 |
| input | 先输入一个小于 100 的正整数 n，然后输入图的邻接矩阵（10000 表示无穷大，即两点之间没有边），最后输入两个 0 到 n-1 的整数表示两个点。 |
| output | 先用迪杰斯特拉算法求给定的第一个点到其余所有结点的最短路径。 然后再输出给定的两个点之间的最短路径（按顺序输出最短路径上的每一个点，每个数据占一行）。 |
| sample_input | 4 0 2 10 10000 2 0 7 3 10 7 0 6 10000 3 6 0 0 2 |
| sample_output | 0 1 2 |
| hint | |
| source | INPOJ |
| solution_language | CS |
| solution | |

CS 试题

| | |
|----------------|---------------------------------|
| id | 9000000010 |
| title | 实验 4.3：用弗洛伊德算法求赋权图的两点间的最短路径的长度。 |
| time_limit | 3000MS |
| memory_limit | 10000KB |
| filesize_limit | 0 |

| | |
|-------------------|---|
| category | PF_Simple |
| difficulty | 10 |
| description | 用弗洛伊德算法求任意两点间的最短路径的长度 |
| input | 先输入一个小于 100 的正整数 n ，然后输入图的邻接矩阵（10000 表示无穷大，即两点之间没有边），之后再输入一个小于 100 的正整数 m ，最后的 m 行每行输入两个不同的 0 到 $n-1$ 之间的整数表示两个点。 |
| output | 用弗洛伊德算法求任意两点间的最短路径的长度，并输出这些两个点之间的最短路径的长度。 |
| sample_input | 4 0 2 10 10000 2 0 7 3 10 7 0 6 10000 3 6 0 2 0 2 3 0 |
| sample_output | 9 5 |
| hint | |
| source | INPOJ |
| solution_language | CS |
| solution | |

CS 试题

| | |
|----------------|---|
| id | 9000000011 |
| title | 实验 4.4：用弗洛伊德算法求赋权图中任意两点间的最短路径。 |
| time_limit | 3000MS |
| memory_limit | 10000KB |
| filesize_limit | 0 |
| category | PF_Simple |
| difficulty | 10 |
| description | 用弗洛伊德算法求任意两点间的最短路径，并输出指定的 m 对结点间的最短路径。 |
| input | 先输入一个小于 100 的正整数 n ，然后输入图的邻接矩阵（10000 表示无穷大，即两点之间没有边），之后再输入一个小于 100 的正整数 m ，最后的 m 行每行输入两个不同的 0 到 $n-1$ 之间的整数表示两个点。 |
| output | 用弗洛伊德算法求任意两点间的最短路径，并输出这些两个点之间的最短路径。 |
| sample_input | 4 0 2 10 10000 2 0 7 3 10 7 0 6 10000 3 6 0 2 0 2 3 0 |

| | |
|-------------------|----------------------------|
| sample_output | 0 1 2 3 1 0 |
| hint | |
| source | INPOJ |
| solution_language | CS |
| solution | |