**实验内容4**

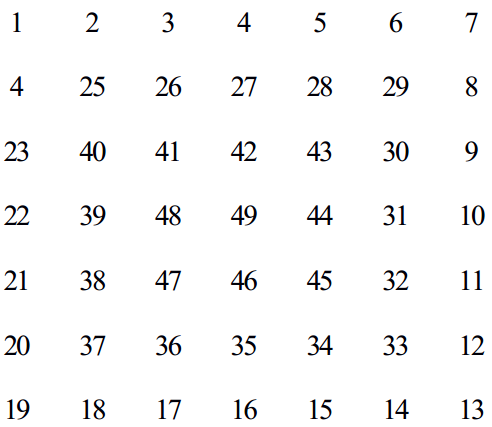
**指针与数组（4学时）**

* **基础类**

1. 从键盘输入10个学生的成绩存储在数组中，求成绩最高者的序号和成绩，以及平均成绩。
2. 对整型数组进行逆置，即：将数组中的所有元素镜像对调。
3. 编写程序，对从键盘输入6个字符串进行排序，并将排序后的结果输出。
4. 将2个按升序排列的数列，仍按升序合并存放到另一个数组中，要求每个数都一次到位，不得在新数组中重新排序。

* **提升类**

1. 对于任意的M×M方阵，编程按照下面的规律对其初始化并输出。如7×7的矩阵。



24

1. 学生的序号输入 n 名学生的成绩，按照分数由高到低的顺序输出学生的名次、序号和成绩，要求成绩相同的学生具有相同的名次。

输入： 学生人数和 n 名学生的成绩

输出： 按分数由高到低的顺序输出学生的名次、序号和成绩

输入： 5

22 42 92 82 92

输出： 1:3,92

1:5,92

3:4,82

4:2,42

4:1,22

* **综合类**

**题目**：特大整数的精确相加。特大整数用长整型也存不下，如果用双精度实型存储则会造成误差，可以用字符数组存储所有位，再按十进制由低到高逐位相加，同时考虑进位。

**算法分析**：

① 初始化：将两个特大整数输入两个字符数组，将两个字符数组的各元素右移，使最低位的元素位置对齐，高位补0，为了存储最高位的进位，位数多的数最高位前也应补一个0。

② 从最低位对应的数组元素开始将数字字符转换为整型数据相加，因为数字字符‘0’ 对应的ASCII值是48，则：整型数据1+2。相当于 ('1'-48)+('2'-48)，即'1'+'2'-96。

③ 将和整除以10，余数就是该位的结果，并转换为字符存入该位，商就是进位数。

④ 再对高一位对应的数组元素操作，将该位数字字符转换为整型相加，并与低位的进位数相加,将和整除以10，余数就是该位的结果，商就是本位的进位数。

⑤ 重复④直到最高位。

⑥ 如果最高位相加时进位数大于0则将此进位数转换为字符存入最高位上1位。

**测试数据1**：

123456789<回车>

23456789<回车>

146913578

**测试数据2**（最高位有进位）：

999999999<回车>

999999999<回车>

1999999998