1. 请编写程序实现非递归版本的一维整数型数组的选择排序算法和冒泡排序算法。
2. 请编写程序对一个char型的二维数组A进行排序。排序之后的数组应该满足如下的要求：

(a)如果 i < j,那么第i行字符的ASCII码的和**不大于**第j行字符ASCII码的和。

(b)对于任意一行k，如果 i < j ，那么A[k][i] **<=** A[k][j]

举例： 如果

A = {

{‘D’,’C’},

{‘B’,’A’}

}

那么排序后A应该变为：

A = {

{‘A’,’B’},

{‘C’,’D’}

}

可以使用如下的数据来测试你的程序：

A = {

{‘S’,’D’,’U’,’Y’,’E’,’S’},

{‘C’,’H’,’I’,’N’,’A’,’!’},

{‘A’,’F’,’T’,’E’,’R’,’X’},

{‘h’,’e’,’r’,’o’,’a’,’b’},

{‘C’,’y’,’y’,’D’,’S’,’!’}

}

1. 八皇后问题。在一个8 x 8的国际象棋棋盘上放置8个皇后，使这些皇后之间不会相互攻击（在国际象棋中，皇后可以走直线和斜线）。请你编程输出所有的摆放方案。
2. 试着增强一下之前作业中的程序。输入为不确定个数的字符（字符数少于5个），输出为这些字符的图案表示。例如，输入为ET，那么输出应该为：

\*\*\*\*\*\* \*\*\*\*\*\*\*

\* \*

\* \*

\*\*\*\*\*\* \*

\* \*

\* \*

\*\*\*\*\*\* \*

1. 同一维数组相似，对于1个二维数组编译器也会为其开辟**一段连续**的内存空间用于数据存放。这个时候编译器有2种可能的数据存放方式：按行存放和按列存放。请你编写程序探索一下C语言使用的存放方式是哪一种（注：请不要在程序中使用指针类型变量）。

int A[2][3] = {{1,2,3},{4,5,6}}

对于数组A，如果按行存放，那么内存镜像如下（图中蓝色方块表示4个字节）：

3

6

1

2

5

4

如果按列存放，那么内存镜像如下：

6

2

5

1

4

3

1. 对于一个整数类型的二维数组A，请你补全如下函数实现对该二维数组任意元素A[m,n]的访问并验证其正确性（**注：在该函数中不允许使用[][]操作符！**）。

int two\_dim\_array\_access(int \*x, int i, int j, int m ，int n，int C)

{

}

该函数参数意义如下：

x： 数组元素A[i][j]的内存地址

m,n:需要访问的目标元素的下标

C： 该数组第2维的大小

如果你的代码正确，那么该函数的返回值仅取决于m和n。例如：

若int A[2][3] = {{1,2,3},{4,5,6}}，那么：

two\_dim\_array\_access(&A[0][0],0,0,1,2,3) 的返回值应该是6； two\_dim\_array\_access(&A[0][1],0,1,1,2,3) 的返回值也应该是6；

1. 现在有1个函数matrix\_multi用于实现矩阵A和矩阵B相乘的功能，并将结果放在矩阵C中。该函数声明如下：

void matrix\_multi(double A[3][4], double B[4][5], double C[3][5]);

请你编写该函数的函数体并测试其正确性。

1. 美化你的代码,为其添加适当的注释。

**实验要求**

1. 作业以mp4视频格式提交。请在视频中展示你的代码并编译运行它, 之后对结果做简要说明以证明你的程序是对的。

2.请在编写代码时保持良好的编码风格（易读、整洁，可以在必要的地方加入注释）。请在程序开头加上注释（参考<https://elixir.bootlin.com/linux/0.01/source/kernel/console.c>）。

3. 请独立完成作业，不允许抄袭。鼓励与同学讨论、上网搜索资料。