1. 一维数组名
   1. 除了两种特殊情况外，都是指向数组第一个元素的指针
      1. 特殊情况1 sizeof 统计数组长度
      2. 特殊情况2 对数组名取地址，数组指针，步长整个数组长度
   2. 数组名是指针常量，指针的指向不可以修改的，而指针指向的值可以改
   3. 传参数时候，int arr[] 可读性更高
   4. 数组索引下标可以为负数
2. 数组指针的定义方式
   1. 先定义出数组类型，再通过类型定义数组指针变量
      1. typedef int(ARRARY\_TYPE)[5];//ARRARY\_TYPE 代表存放5个int类型元素的数组 的数组类型
   2. 先定义数组指针类型，再通过类型定义数组指针变量
      1. typedef int(\*ARRARY\_TYPE)[5];
   3. 直接定义数组指针变量
      1. int(\* p )[5] = &arr;
3. 二维数组名
   1. 二维数组名 除了两种特殊情况外，是指向第一个一维数组的 数组指针
   2. 两种特殊情况
      1. sizeof 统计二维数组大小
      2. 对数组名称取地址 int(\*p)[3][3] = &arr
   3. 二维数组做函数参数
      1. //void printArray(int (\*array)[3], int row, int col)
      2. //void printArray(int array[][3], int row ,int col)
      3. void printArray(int array[3][3], int row ,int col) 可读性比较高
   4. 数组指针 和 指针数组？
      1. 数组指针： 指向数组的指针
      2. 指针数组： 由指针组成数组
4. 指针数组排序
   1. 选择排序
      1. 例如从小到大
      2. 开始认定最小值下标为i，从j = i+1的位置起找真实最小值下标，如果计算的真实最小值下标与i不等，互换元素
   2. 利用选择排序实现 指针数组 从大到小排序
      1. 字符串对比
      2. if ( *strcmp*(pArr[max],pArr[j]) == -1)
5. 结构体基本概念
   1. 加typedef 可以给结构体起别名
   2. 不加typedef ，可以直接创建一个结构体变量
   3. 结构体声明 可以是匿名
   4. 在栈上创建和在堆区创建结构体
   5. 在栈上和堆区创建结构体变量数组
6. 结构体深浅拷贝
   1. 系统提供的赋值操作是 浅拷贝 – 简单值拷贝，逐字节拷贝
   2. 如果结构体中有属性 创建在堆区，就会出现问题，在释放期间，一段内存重复释放，一段内存泄露
   3. 解决方案：自己手动去做赋值操作，提供深拷贝
7. 结构体嵌套一级指针练习
   1. 在堆区创建一个 结构体指针数组
      1. malloc(sizeof(struct Person \*) \*3 )
   2. 在堆区创建出结构体变量
      1. malloc(sizeof(struct Person))
   3. 在堆区创建出具体姓名
      1. malloc(sizeof(char )\*64)；
   4. 打印数据
   5. 释放数组