

实验要求：数组运算器

实验目的

掌握数组的基本操作，包括一维数组的定义和使用、循环结构和分支结构的应用、函数的定义与调用、数组作为函数参数的传递，以及数组处理技术（如排序、插入、删除、查找、统计和逆置）。

实验内容

必做部分（80%）

根据提供的代码框架，完成以下功能模块的实现，每个功能模块占 4%，共计 20 个功能模块：

1. 用指定范围的随机数填充数组
2. 从键盘输入填充数组
3. 整个数组填同一个值
4. 用等差序列填充数组
5. 插入新元素
6. 删除指定下标的元素
7. 删除指定区间的一组元素
8. 查找元素
9. 判断是否升序排列
10. 判断是否降序排列
11. 判断是否全部相等

12. 求最大值的下标
13. 求最小值的下标
14. 计算平均值
15. 在有序数组中插入新元素
16. 冒泡排序
17. 选择排序
18. 插入排序
19. 数组逆置
20. 二分查找

选做部分 (10%)

可以选择实现以下高级功能之一，每个功能模块占 5%：

1. 快速排序 (5%)
2. 归并排序 (5%)

交互界面 (10%)

交互界面需要清晰简洁：

1. 分级菜单 (5%)
2. 菜单提示与特殊情况处理 (5%)
 - 例如预期外的输入

实验要求

1. 实现必做部分的功能，确保每个功能都能正确运行。
2. 撰写实验报告，详细记录代码实现过程、测试结果及实验总结。

实验评分标准

实验检查 80%

- 必做部分：80%
- 交互界面：10%
- 选做部分：10%

实验报告 20%

提交要求

提交内容应包括：

- 完整的实验代码，具有良好的注释和代码风格。
- 详细的实验报告，描述实现过程、遇到的问题及解决方案。

提交实验源代码压缩包，命名为 PB2401xxxxxx_ 姓名
实验报告提交 pdf，不放在压缩包内！不放在压缩包内！

注意事项

推荐函数

```
1 void printarr(int *a, int n);           // 打印数组
2 void FillArray1(int a[], int n);        // 随机数填充
3 void FillArray2(int a[], int n);        // 键盘输入
4 void FillArray3(int a[], int n, int val); // 全部填同一
    值
5 void FillArray4(int a[], int n, int start, int diff); // 等差序列填
    充
6 void insert(int a[], int *n, int pos, int val); // 插入
7 void Delete(int a[], int *n, int id);     // 删除
8 void DeleteRange(int a[], int *n, int s, int t); // 删除区间
9 int Search(int a[], int n, int val);      // 查找
10 bool isAsc(int a[], int n);              // 判断升序
11 bool isDesc(int a[], int n);             // 判断降序
12 bool isEqual(int a[], int n);            // 判断相等
```

```

13 int maxIndex(int a[], int n);           // 最大值下标
14 int minIndex(int a[], int n);           // 最小值下标
15 float average(int a[], int n);           // 平均值
16 void OrderInsert(int a[], int *n, int val); // 有序插入
17 void BubbleSort(int a[], int n);         // 冒泡排序
18 void SelectSort(int a[], int n);         // 选择排序
19 void InsertSort(int a[], int n);         // 插入排序
20 void QuickSort(int a[], int left, int right); // 快速排序
21 void MergeSort(int a[], int left, int right); // 归并排序
22 void Merge(int a[], int left, int mid, int right); // 归并
23 void Reverse(int a[], int n);           // 逆置
24 int BiSearch(int a[], int n, int val);   // 二分查找
25 void menu();                             // 菜单显示

```

菜单示例

不一定按这个格式，但需要保证清晰简明，有可操作性。

```

26 void main() {
27     int sn,s1,s2,a[ArraySize];
28
29     for(;;) {
30         printf("0、退出\n");
31         printf("1、生成样本数据\n");
32         printf("2、显示数组\n");
33         //其它菜单
34         printf("请选择0-8:");
35         for(;;) {
36             scanf("%d",&sn);
37             if(sn<0 || sn>8) printf("\n\t输入错误，重选0-8: ");
38             else break;
39         }
40         switch(sn) {
41             case 1:
42                 printf("1、用指定范围的随机数填充数组\n");
43                 printf("2、从键盘输入\n");
44                 //其它菜单
45                 printf("请选择生成数据的方式 (1-4) :");
46                 for(;;) {
47                     scanf("%d",&s1);

```

```

48         if(sn<1 || sn>4) printf("\n\t输入错误, 重选1-4: ");
49         else break;
50     }
51     if(s1==1) FillArray1(a, ArraySize);
52     if(s1==2) FillArray2(a, ArraySize);
53     break;
54 case 2:
55     printf("样本数据如下: \n");
56     printarr(a, ArraySize);
57     break;
58 case 0:
59     printf("\n    再见!    \n");
60     return;
61 }
62 }
63 }

```