

# 暨南大学本科实验报告专用纸

课程名称 高级语言程序设计实验 成绩评定             
实验项目名称 一维数组的使用 指导教师 张鑫源  
实验项目编号 ⑦ 实验项目类型            实验地点 家中  
学生姓名 位雨昕 学号 2019051112  
学院 智能科学与工程学院 系            专业 信息安全  
实验时间 2020 年 4 月 15 日 上午~4 月 15 日 下午 温度     °C 湿度     

## (一) 实验目的

1. 进一步了解 Visual Studio 的使用以及 C 语言程序的结构；
2. 接触 C 语言中的部分常用函数、初步掌握其使用方法；
3. 熟练掌握条件语句、循环语句、一维数组的使用；
4. 锻炼个人的编程操作能力。

## (二) 实验内容和要求

内容：

有两个长度均为 15 的整型数组，分别是 a 和 b，其中，a 为用户输入得到，需要设计程序实现以下功能：

1. 通过 a“整体”赋值得到 b；
2. 得到 b 之后对其从大到小排序；
3. 用户输入整型数字 c，确定 c 在数组 b 中的位次并输出。

示例：

如果  $a[15]=\{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14\}$ ，那么首先通过 a 对 b 整体赋值得到  $b[15]=\{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14\}$ ，再对 b 中的元素按照从大到小排序后得到新的数组  $b[15]=\{14,13,12,11,10,9,8,7,6,5,4,3,2,1,0\}$ ，排序结束后用户输入数字 8，程序输出 8 在 b 中的位次，即 7。

要求:

1. 需要统计程序执行的时间, 单位为毫秒;
2. 用户输入的数字可能不在 b 中, 此时需要给出位次“不存在”。

### (三) 主要仪器设备

仪器: 计算机

实验环境: Visual Studio Community 2019

### (四) 源程序

```
#include<stdio.h>
#include <time.h>
#include <windows.h>

int main()
{
    long start1, stop1, start2, stop2, start3, stop3;
    long total;
    int a[15], b[15];
    int i, n, m, j, k, element, max;
    start1 = clock();
    printf("请任意输入15个不重复的正整数, 不同数字之间请用空格隔开: \n");
    stop1 = clock();
    for (i = 0; i < 15; i++)    //用于获取数组a
        scanf_s("%d", &a[i]);
    start2 = clock();
    for (i = 0; i < 15; i++)
        b[i] = a[i];    //将数组a的值赋给数组b
    printf("由大到小排序如下: \n");
    //以下用于数组b比较大小并排序, 使用选择排序法
    for (n = 0; n < 15; n++)    //控制某一轮的排序
    {
        max = n;    //max为最大数的下标 (先假设n为最大值下标)
        for (j = n + 1; j < 15; j++)    //找到该轮的最大值 (j~15中找最大)
        {
            if (b[max] < b[j])
                max = j;
        }
        if (max != n)
```

```

        {
            k = b[n];
            b[n] = b[max];
            b[max] = k;
        }

    }

    for (i = 0; i < 15; i++)    //将排序后的数组b输出
        printf("%-4d", b[i]);
    stop2 = clock();
    //以下用于查找数字并输出位次
    printf("\n请输入您要查找的数字: \n");
    start3 = clock();
    scanf_s("%d", &element);
    for (m = 0; m < 15; m++)
    {
        if (b[m] == element)
            break;
    }
    if (m < 15)    //此时循环在有效范围内跳出,说明查找到了该数字
        printf("%d为第%d个数字", element, m+1);
    else
        printf("不存在该数字! ");

    stop3 = clock();
    total = ((stop1 - start1) + (stop2 - start2) + (stop3 - start3));
    printf("\n本程序运行用时%ldms", total);

    return 0;
}

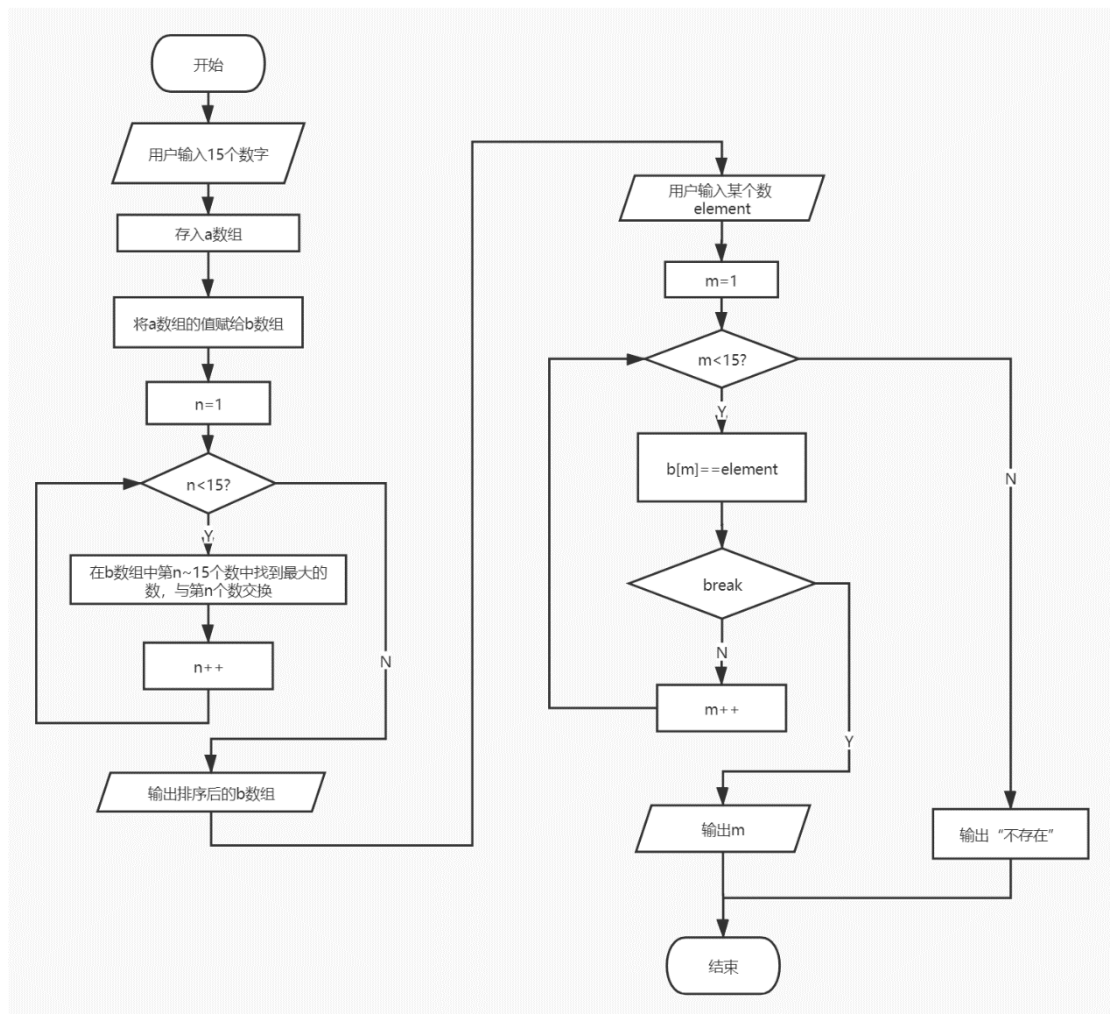
```

## （五）实验步骤与调试

### 实验步骤：

1. 对实验进行基本构思，绘制流程图；
2. 启动 Visual Studio，创建新项目。将源程序写在新项目中；
3. 利用“本地 Windows 调试器”进行调试；
4. 进行多次调试并修正，直至得出理想结果。

实验思路：



调试：

1. 最初尝试运行程序时：

```
由大到小排序如下：
555 14 13 12 11 10
请输入您要查找的数字：
555
不存在该数字！
```

输出 555 的位次时运行出现错误。

检查代码后发现问题所在：

```
for (m = 1; m < 15; m++)
{
    if (b[m] == element)
        break;
}
```

错将下标的开头当作 1，而重新排序后 555 在数组 b 中的索引为 0，遍历的时候直接跳过了。将 m 初值赋为 0 后，解决了该问题。

```
由大到小排序如下：
555 14 13 12 11 10
请输入您要查找的数字：
555
555为第1个数字
本程序运行用时3555ms
```

2. 

```
11为第5个数字
本程序运行用时15974ms
```

```
12为第1个数字
本程序运行用时3045ms
```

统计程序执行时间时使用了 clock() 函数，在整个程序的头尾加入了 `start = clock();` `stop = clock();` 这两句代码，统计了整个程序的运行时间，所以包括了用户输入数值的时间。这样做无法精确地体现程序的运行效率。于是尝试在出现输入语句的前后加上了 start 与 stop 的计时语句：分别计算三段代码的运行时间。

运行结果：

```
12为第4个数字
本程序运行用时4ms
C:\Users\36687\Desktop

555为第1个数字
本程序运行用时6ms
C:\Users\36687\Desktop

14为第2个数字
本程序运行用时5ms
C:\Users\36687\Desktop
```

虽然运行结果基本一致，但仍然出现部分误差。

查阅后了解到，C 语言的程序运行时间并不精确，有过多的因素影响，存在一定误差。

## （六）实验结果与分析

```
请任意输入15个不重复的正整数,不同数字之间请用空格隔开:
0 1 2 3 14 5 6 8 7 9 10 11 12 13 555
由大到小排序如下:
555 14 13 12 11 10 9 8 7 6 5 3 2 1 0
请输入您要查找的数字:
555
555为第1个数字
本程序运行用时4ms
```

```
请任意输入15个不重复的正整数,不同数字之间请用空格隔开:  
0 1 2 3 14 5 6 8 7 9 10 11 12 13 555  
由大到小排序如下:  
555 14 13 12 11 10 9 8 7 6 5 3 2 1 0  
请输入您要查找的数字:  
666  
不存在该数字!  
本程序运行用时4ms
```

1. 程序可按实验要求运行;
2. 程序情况基本符合理想效果;
3. 调试过程中出现的问题均已解决。