

北京林业大学

2025学年—2026学年第1学期 程序设计基础 实验报告书

专业：计算机类 班级：计算机类 25-2

姓名：玄尚 学号：251002213

实验地点：学研 N01 任课教师：王春玲

实验题目：实验 2 数组

实验环境：Visual Studio Code (GCC)

一、实验目的

- 能够进行一维数组的定义和初始化，以及数组元素的输入和输出；
- 熟悉数组处理的常用算法的设计方法以及对数组进行插入和删除等；
- 具备简单数据查找能力

二、实验内容

编写程序，实现如下任务：从键盘输入若干个整数，输入整数的个数小于100，其值在-100~100 范围内，用-1 作为输入结束的标志。统计每个整数的个数 并从大到小排序，输出排序后的结果。例如：

```
请输入数字 (-100~100) , -1作为输入结束, 请输入小于100个数:  
1 1 1 1 2 2 2 3 3 3 3 5 5 5 5 -3 -4 -5 -1  
数字 频率  
1 4  
2 3  
3 5  
5 5  
-3 1  
-4 1  
-5 1
```

排序后的结果：

```
数字 频率  
3 5  
5 5  
1 4  
2 3  
-3 1  
-4 1  
-5 1  
Press any key to continue
```

三、实验步骤及结果

The screenshot shows a terminal window with a dark background. At the top, there are three colored window control buttons (red, yellow, green). Below them is the C source code for a program that reads a list of numbers, calculates their frequencies, and then sorts them. The code uses arrays `numbers` and `frequency` to store the data. It includes sections for input, frequency calculation, output, and sorting.

```
1 #include <stdio.h>
2 int main(){
3     printf("Please input numbers(-100~100 , -1 = end)\n");
4     int numbers[100];
5     int frequency[205] = {0};
6     int index = 0;
7     while(scanf("%d",&numbers[index]) != EOF){
8         if(numbers[index] == -1){
9             break;
10        }
11        index++;
12    }
13    for(int i = 0;i < index;i++){
14        int frequency_index = numbers[i] + 100;
15        frequency[frequency_index]++;
16    }
17    printf("Numbers\tFrequency\n");
18    for(int i = 0;i < 200;i++){
19        if(frequency[i] > 0){
20            int num = i - 100;
21            printf("%d\t%d\n",num,frequency[i]);
22        }
23    }
24    printf("After sorting:\n");
25    printf("\n");
26    printf("Numbers\tFrequency\n");
27    for(int i = 0;i < 201;i++){
28        int max = i;
29        for (int j = i; j < 201; j++){
30            if (frequency[j] > frequency[max] || (frequency[j] == frequency[max] && j > max)){
31                max = j;
32            }
33        }
34        if(frequency[max]!=0){
35            printf("%d\t%d\n", max - 100, frequency[max]);
36        }
37        frequency[max] = 0;
38    }
39    return 0;
40 }
```

测试用例: 1 1 1 1 1 2 3 4 2 3 4 99 101 -2 -2 -3 2 3 4 1 5 1 3 2 99 -1 99

测试结果:

```
Numbers Frequency
-3      1
-2      2
1       7
2       4
3       4
4       3
5       1
99     2
After sorting:

Numbers Frequency
1       7
3       4
2       4
4       3
99     2
-2      2
5       1
-3      1
```

四、实验分析

在实验的过程中，我遇到了数字频率在排序后会莫名冒出不明数字 100 并进行输出的问题，通过调试发现，问题发生的原因是在进行频率排序时数组越界的问题，解决方法就是简单擴大数组的容量。

本次实验中的关键收获在于对数组边界问题的深刻理解。通过排查数字频率排序中异常出现的“100”，我认识到数组越界访问会引发不可预知的数据混乱。这一问题警示我们，在涉及数组操作时，必须严谨计算索引范围，并预留足够的安全容量。同时，调试过程中定位具体异常点的经验，也进一步强化了我系统化排查问题的能力。