学号 E21714049 专业 计算机科学与技术 姓名 梅世祺
实验日期 2019-04-25 教师签字 成绩

实验报告

【实验名称】 数组与指针

【实验目的】

1. 熟练掌握数组的定义、初始化和数组元素的访问方法以及数组在内存中的组织形式。
2. 理解指针的概念，熟练掌握指针变量的定义、初始化和访问方法。
3. 理解指针与数组的关系，掌握通过指针操作数组元素的方法。
4. 掌握函数指针的使用方法。

【实验原理】

比较相邻的元素。如果第一个比第二个大，就交换他们两个；对每一对相邻元素做同样的工作，从开始第一对到结尾的最后一对；在这一点，最后的元素应该会是最大的数；针对所有的元素重复以上的步骤，除了最后一个；持续每次对越来越少的元素重复上面的步骤，直到没有任何一对数字需要比较。

【实验内容】

实验一 冒泡排序

题目：编写冒泡排序算法，实现对命令行读入的 10 个整数进行升序或降序排列，并打印排序后的结果。

要求： 1. 升序或降序指令由命令行读取；

2. 升序和降序各测试两组数据。

原理： 冒泡排序算法的基本思想是依次将数组中最大（小）的元素沉到数组末端。例如，使用冒泡排序对数组 arr 进行升序排序时，首先将数组中最大元素沉到 arr[n-1]处，接着将第二大元素沉到 arr[n-2]处，以此类推。

附加： 冒泡排序算法内部使用指针操作数组。

实验结果（含源码）：

实验一 冒泡排序

```
// bubble sort
// true: ascending
// false: descending
void bubbleSort(int *a, int len, bool type=true) {
    int temp;
    for (int i=0; i<len-1; i++) {
        for (int j=0; j<len-1-i; j++) {
            if ( (a[j] > a[j+1]) && type) {
                // exchange values of a[j+1] and a[j+1]
                temp = a[j]; a[j] = a[j+1]; a[j+1] = temp;
            }

            if( (a[j] < a[j+1]) && !type) {
                temp = a[j]; a[j] = a[j+1]; a[j+1] = temp;
            }
        }
    }
}
```

冒泡排序：外循环决定遍历的轮数，对于一个长度为 len 的数组，外循环需要进行 len-1 轮，内循环在每一轮中将还未比出大小的元素进行两两比较，每一次内循环都可以找到一个最大值或最小值，放在最后。
此外，还在交换时元素时增加了 type 参数，用于确定是升序排序还是降序排序。

```cpp
int main(int argc, char const *argv[])
{
    int a[10];
    bool type = true;

    // input
    cout<<"Please input the 10 values: ";
    for(int i=0; i<10; i++) {
        cin>>a[i];
    }
    cout<<"Please determine the sort type: (1 = ascending, 0 = descending): ";
    cin>>type;
    bubbleSort(a, 10, type);

    for(int i=0; i<10; i++) {
        cout<<a[i]<<" "<<endl;
    }
    return 0;
}
```

在主函数中，输入待排序的数据和排序类型（升序或降序）后，调用 bubbleSort 函数进行冒泡排序，最后输出排序后的数组以检查排序结果是否正确。

```
➜ ./build/bin
Please input the 10 values: 0 1 -1 111 999 -999 0 9 10 999
Please determine the sort type: (1 = ascending, 0 = descending): 1
-999
-1
0
0
1
9
10
111
999
999
➜ ./build/bin
Please input the 10 values: 0 0 1 1 2 2 999 -1 -9999 27
Please determine the sort type: (1 = ascending, 0 = descending): 0
999
27
2
2
1
1
0
0
-1
-9999
```

运行结果 1

```
➜ ./build/bin
Please input the 10 values: 0 9 -1 20 9 111 000 001 09 28
Please determine the sort type: (1 = ascending, 0 = descending): 1
-1
0
0
1
9
9
9
20
28
111
➜ ./build/bin
```

运行结果 2

```
➜ ./build/bin
Please input the 10 values: 0 0 -1 29 3393939 32320 0202 -2 32 10000
Please determine the sort type: (1 = ascending, 0 = descending): 0
3393939
32320
10000
202
32
29
0
0
-1
-2
```

运行结果 3

实验二 图形面积计算

题目： 使用函数指针实现三角形和矩形面积计算函数的统一封装 double area(double x, double y, double (*area_func)(double x, double y))。

要求： 1. 分别实现三角形和矩形面积计算函数；

2. 从命令行读入待计算图形类型，待计算图形参数（如对三角形而言是底和高，对矩形而言是长和宽）。

原理：分别定义三角形和矩形面积计算函数，并以函数指针的形式作

为 area 函数的第三个参数，计算图形面积的时候直接调用 area 函数即可。

实验结果（含源码）：

```
// calculate area
double area(double x, double y, double (*area_func)(double x, double y)) {
    return (*area_func)(x, y);
}

double triangleArea(double l, double h) {
    return (l * h)/2;
}

double squareArea(double a, double b) {
    return a*b;
}


int main(int argc, char const *argv[])
{
    double x, y;
    cin>>x>>y;
    cout<<area(x, y, triangleArea)<<endl<<area(x, y, squareArea)<<endl;
    return 0;
}
```

函数指针定义格式： func2((*func1)(...) ;)
函数指针传参 func1() {}; func2(func1);

```
-----------  构建用时:529 ms --------------
20 12
120
240
```

程序运行结果：第一行为三角形的面积，第二行为矩形的面积。