

# 实验一 简单程序开发

## 实验目的

1. 理解并掌握程序的分支、循环结构
2. 学习数组的定义、初始化、赋值和使用的方法
3. 学习给函数传递数组的方法
4. 学习指针和引用的定义和使用方法
5. 学习用指针和引用给函数传递参数
6. 提高程序可读性
7. 进一步学习掌握查找与修改编译错误的方法
8. 初步学习调试方法

## 实验内容

### 1. 循环与分支结构

编写一个程序，循环从标准输入读入某雇员的工作时间（以小时计）和每小时的工资数，计算并输出他的工资。若雇员月工作小时超过40小时，则超过部分按原工资的1.5倍的加班工资来计算。若雇员月工作小时超过50小时，则超过部分按原工资的3 倍的加班工资来计算。

[测试数据]

输入：30 4

输出：120

输入：45 4.5

输出：213.75

输入：60 5

输出：425

输入：0 0

程序结束

[实现要求]

1. 分别用三种循环（for, while, do while）完成程序要求
2. 要求有输入提示和输出提示，如要输入雇员的工作时间和每小时的工资值时，可以提示：

“Please input employee's work time and wage\_per\_hour:”

输出时，提示：

“The employee's wage :”。

3. 循环在用户输入的工作时间为0时结束。
4. 为你的程序加上注释，使得其清晰可读。
5. 尝试利用调试程序来修改你程序的逻辑错误。

参考代码1：

```
//用while实现
#include <iostream>
using namespace std;
void main()
{ int workTime; //工作时间
float wagePerHour; //每小时工资
int overTime1 = 0; //超时过小时时间
int overTime2 = 0; //超时过小时时间
while(1) {
    cout << "Please input employee_work_time and wage_per_hour (0 to end):\n";
    //读入工作时间和每小时的工资
    cin >> workTime >> wagePerHour;
    //输入工作时间为表示程序结束
    if (workTime == 0) break;
    //根据工作时间不同判断超时性质
    if (workTime > 50)
    {
        overTime2 = workTime - 50;
        overTime1 = 10;
        workTime = 40; }
    else if (workTime > 40)
    {
        overTime2 = 0;
        overTime1 = workTime - 40;
        workTime = 40;
    }
    else
        overTime1 = overTime2 = 0;
    //输出工资值cout << "Wages : ";
    cout << workTime * wagePerHour + overTime1 * wagePerHour * 1.5 + overTime2 *
wagePerHour * 3 << endl;
}
}
```

### 参考代码2:

```
// 用do...while循环实现
```

### 参考代码3:

```
//用for循环实现
```

## 2. 数组排序

从键盘读入若干整数，将它们按由低到高排序输出。

[测试数据]:

程序先输出: Please input array number:

用户输入: 5

程序再输出: Please input all the integer:

用户输入: 300 700 600 450 500

程序输出: 300 450 500 600 700

[实现要求]:

1. 用一个数组存放各个整数;
2. 在主函数main()中实现数据的输入和输出操作, 并用一个函数实现对数组元素的排序操作。
3. 排序函数调用另一个函数swap()实现两个数组元素的交换。可以使用指针、引用两种方式实现函数参数的传递:

```
swap(int *pa, int *pb);
```

```
swap(int * a; int & b);
```

### 参考代码 1

```
//用数组实现
```

```
//从键盘读入若干整数，将它们按由低到高排序输出。
```

```
#include <iostream>
```

```
using namespace std;
```

```
void sort(int num, int iArray[]);
```

```
void swap(int *pa, int *pb);
```

```
void swap(int &a, int &b);
```

```
void main()
```

```
{
```

```
    int num,i;
```

```
    int iArray[10];
```

```
    //输入数组元素个数(如果输入值大于, 则要求重新输入)
```

```
    while(1) {
```

```
        cout << "\nPlease input array number(<10):";
```

```
        cin >> num;
```

```

        if (num <= 10) break;
    }
    //输入所有数组元素
    cout << "Please input all the integer:";
    for(i = 0; i < num; i++)
        cin >> iArray[i];
    //对数组元素排序
    sort(num, iArray);
    //输出排序后的内容
    cout << endl;
    for(i = 0; i < num; i++)
        cout << iArray[i] << " ";
    cout << endl;
}
//数组排序函数，采用选择排序法
void sort(int num, int iArray[])
{
}

//下面两个swap函数分别用指针和引用实现了函数参数的传递
void swap(int *pa, int *pb)
{
}
void swap(int &a, int &b)
{
}

```

## 参考代码 2

```

//用指针实现
//从键盘读入若干整数，将它们按由低到高排序输出。

```