**广州大学学生实验报告**

**开课学院及实验室：**计算机学院/电子信息楼418A **2017年11月5日**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学 院** | 计算机 | **年级/专业/班** | 软件171 | **姓名** | 谢金宏 | **学号** | 1706300001 |
| **实验课程名称** | 程序设计基础 | | | | | **成绩** |  |
| **实验项目名称** | **实验2函数** | | | | | **指导老师** | 张为 |

(\*\*\*（1）报告只能为文字和图片；（2）实验项目名称不能有任何改动；（3）必须填写正确的姓名和学号；否则会影响最后成绩的统计，后果自负\*\*\*)

**一、实验目的及要求**

主要叙述本节对应要掌握的主要概念和要解决的主要问题。

1．掌握函数的定义方法，及函数调用的形式；

2．掌握函数实参与形参的对应关系，“值传递”与“地址传递”的方式与区别；

3．掌握函数的应用方法，熟悉函数的嵌套调用和递归调用的设计方法；

4．了解全局变量、局部变量、动态变量及静态变量的概念和使用方法；

5．综合应用顺序结构、选择结构和循环结构的设计方法设计函数程序。

**二、实验设备与平台**

1. 实验设备：计算机；

2. 平台：平台：Windows 10操作系统，GCC 6.3.0。

**三、实验内容及步骤**

1. 分析下面程序,给出运行结果。本题要求理解和掌握函数的定义和调用

//**实验2\_1：**sy2\_1.cpp

#include <iostream>

using namespace std;

int f(int i)

{

i = i \* i - 1;

return (i);

}

int main()

{

    int i;

    for (i = 0; i < 5; i++)

cout << i << " " << f(i) << endl;

return 0;

}

程序运行结果：

0 -1

1 0

2 3

3 8

4 15

2. 下面程序为选择排序，要求按如下步骤完成本题实验，理解和掌握函数参数的传递理解函数的嵌套调用。

//**实验2\_2：**sy2\_2.cpp

#include <iostream>

using namespace std;

#define N 10

int max(int x[], int n)

{

    int i, j;

    j = 0;

    for (i = 1; i < n; i++)

        if(x[i] > x[j]) j = i;

    return j;

}

void sel\_sort(int x[], int n)

{

    int i, j;

    for(i = n; i > 1; i--) {

j = max(x, i);

int temp = x[j];

x[j] = x[i-1];

x[i-1] = temp;

}

}

int main()

{

    int a[N], i;

    for(i = 0; i < N; i++)

        cin >> a[i];

    sel\_sort(a, N);

    for(i = 0; i < N; i++)

        cout << a[i] << " ";

    cout << endl;

}

1) 阅读程序，理解选择排序的思想。

2) 上机编辑，调试程序。

3) 用两组数据，给出程序运行结果。

>> 1 2 3 4 5 6 7 8 9 10

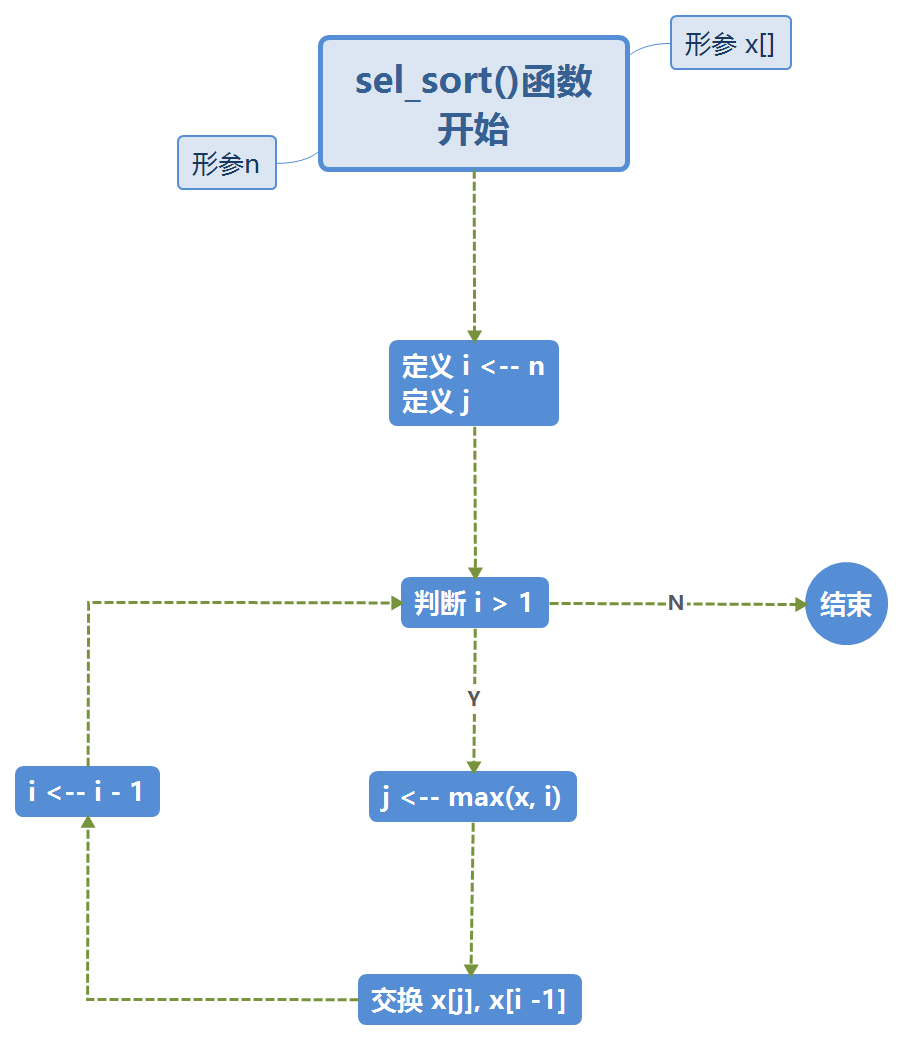
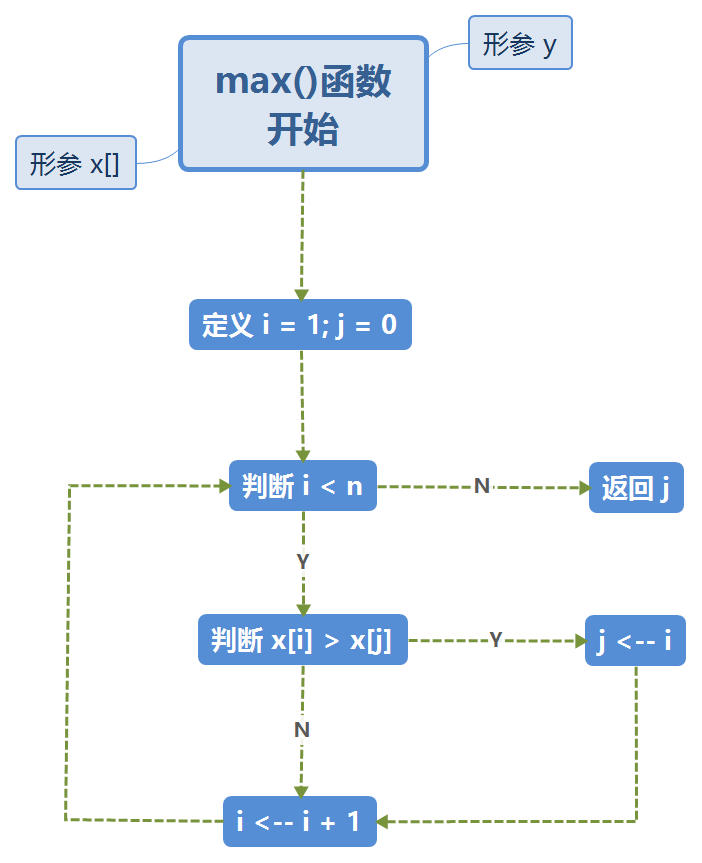
1. 2 3 4 5 6 7 8 9 10

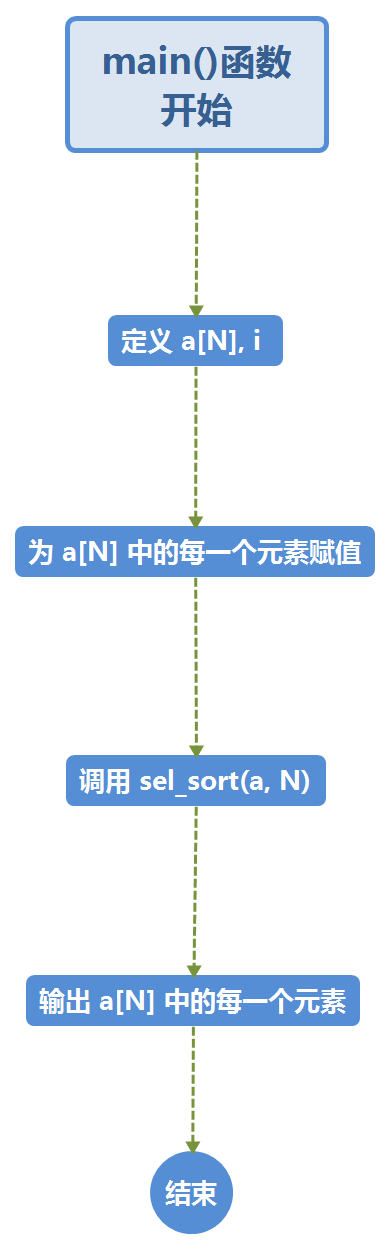
>> 998 29 31 21 31 1 3 13 21 33 123

1 3 13 21 21 29 31 31 33 998

4) 用单步执行，观察程序执行的流程。

5) 画出流程图描述程序的实现。





6) 问题：什么是函数的嵌套调用？

在一个函数中调用另一个函数。

比如main()函数中调用sel\_sort()函数，sel\_sort()函数中调用max()函数等等都是函数的调用。

3. 函数的递归调用

题目：采用递归方式实现二分查找

{问题分析}二分查找算法可以描述为：对于一个已经从小到大排序的数据序列，用给定数据key与查找区间中央位置的数据比较，如果相等则表明查找成功；否则，如果key比中央位置的数据小，则在前半个区间用同样的方法继续查找；否则在后半个区间用同样的方法继续查找。因此，这是一个递归的过程，当查找区间的长度为0时，说明查找不成功。

要求完成本题的填空并上机调试通过，理解递归调用的执行过程。

//**实验2\_3：**sy2\_3.cpp

#include <iostream>

using namespace std;

#define NUM 10

int search(int value[], int key, int low, int high);

int main()

{

    int value[NUM], result, key;

    int i;

    cout << "请输入含有十个元素的非降序序列:" << endl;

    for ( i = 0; i < NUM; i++)

        cin >> value[i]; /\* 输入有序数列\*/

    cout << "请输入其中一个元素值:" << endl;

    cin >> key; /\* 输入待查找的数值\*/

    result = search(value, key, 0, NUM - 1); /\*调用二分查找函数进行查找\*/

    if (result != -1)

        cout << "第" << result << "个元素是：" << key << endl;

    else

        cout << "不能找到" << key <<endl;

}

int search(int value[], int key, int low, int high)

{

    int mid;

    if (low > high) return -1; /\*查找区间为空\*/

    mid = (high + low) / 2; /\*求中间位置\*/

    if (value[mid] == key) return mid; /\*得到查找的数据位置\*/

    if (key < value[mid]) return search(value, key, low, mid - 1);

  else return search(value, key, mid + 1, high);

}

4. 全局变量与局部变量及变量的存储类型

1) 全局变量与局部变量

题目：一维数组内放10个学生成绩，写函数求平均分、最高分和最低分。要求掌握全局变量与局部变量的使用,并回答问题。

#include <iostream>

#include <ctime>

using namespace std;

const int n = 10;

int score[n];

double cal\_average();

int get\_highest();

int get\_lowest();

int main()

{

    srand(time(0));

    for (int i = 0; i < n; i++) {

        score[i] = rand();

        i ? (cout << ' ' << score[i]) : (cout << score[i]);

    }

    cout << endl;

    cout << "Average: " << cal\_average() << endl;

    cout << "Highest: " << get\_highest() << "\nLowest: " << get\_lowest() << endl;

}

double cal\_average()

{

    int sum = 0;

    for (int i = 0; i < n; i++) sum += score[i];

    return (double)sum / n;

}

int get\_highest()

{

    int h = score[0];

    for (int i = 1; i < n; i++) if (score[i] > h) h = score[i];

    return h;

}

int get\_lowest()

{

    int l = score[0];

    for (int i = 1; i < n; i++) if (score[i] < l) l = score[i];

    return l;

}

1. 用两组数据，给出运行结果。

<< 647 25400 9707 5614 10817 23072 30256 2156 8737 6327

Average: 12273.3

Highest: 30256

Lowest: 647

<< 656 24877 30531 12268 8993 2084 6825 26142 25504 20651

Average: 15853.1

Highest: 30531

Lowest: 656

1. 回答外部变量与内部变量的不同。

作用域不同。

全局变量（外部变量）具有文件作用域。

局部变量（内部变量）的作用域可视为在包含它的最短左右花括号之间。

1. 思考：本题如不采用外部变量定义形式，同样采用主函数调用子函数方式如何实现？

在代码的基础上修改函数的原型、定义和调用形式即可。

double cal\_average(int \*);

int get\_highest(int \*);

int get\_lowest(int \*);

double cal\_average(int \* score)

{

    int sum = 0;

    for (int i = 0; i < n; i++) sum += score[i];

    return (double)sum / n;

}

int get\_highest(int \* score)

{

    int h = score[0];

    for (int i = 1; i < n; i++) if (score[i] > h) h = score[i];

    return h;

}

int get\_lowest(int \* score)

{

    int l = score[0];

    for (int i = 1; i < n; i++) if (score[i] < l) l = score[i];

    return l;

}

int main()

{

    srand(time(0));

    int score[n];

    for (int i = 0; i < n; i++) {

        score[i] = rand();

        i ? (cout << ' ' << score[i]) : (cout << score[i]);

    }

    cout << endl;

    cout << "Average: " << cal\_average(score) << endl;

    cout << "Highest: " << get\_highest(score) << "\nLowest: " << get\_lowest(score) << endl;

}

2) 变量的存储类型

验证：自动变量与静态变量的特性，理解变量的存储方式。

题目：根据以下程序，写出程序运行结果，并指出自动变量与静态变量的区别。

//**实验2\_4：**sy2\_4.cpp

#include <iostream>

using namespace std;

void auto\_static()

{

    int var\_auto = 0;

    static int var\_static=0;

    cout << "var\_auto=" << var\_auto << endl;

    cout << "var\_static=" << var\_static << endl;

    ++var\_auto;

    ++var\_static;

}

int main()

{

    int i;

    for (i=0; i<5; i++)

        auto\_static();

}

var\_auto = 0

var\_static = 0

var\_auto = 0

var\_static = 1

var\_auto = 0

var\_static = 2

var\_auto = 0

var\_static = 3

var\_auto = 0

var\_static = 4

自动变量生存于栈，在函数中定义的自动变量在每次函数调用时会被重新初始化。

静态变量生存于全局空间，只在程序启动时被初始化一次。

实验2-5. 函数的编程应用

编写程序，计算下面公式并输出结果。

C

1. 编写一个函数计算n!

long long factorial(long long n)

{

    if (n == 0 || n == 1) return 1;

    else return factorial(n - 1) \* n;

}

1. 编写主函数，由键盘输入n和m，调用（1）中的函数完成计算。

int main()

{

    int n, m;

    while (cout << "Input n, m:", cin >> n >> m)

    {

        if (n < 0 || m < 0 || m > n) {

            cout << "Impossible" << endl;

            continue;

        }

        long long rst = factorial(n) / (factorial(n - m) \* factorial(m));

        cout << rst << endl;

    }

    cout << "Job Done : )" << endl;

}

1. 输入n和m要给出提示，并检查n和m的合理性，不合理的输入应输出错误信息，并不再进行计算。

如上

1. 运行程序，输出计算, , 

Input n, m: << 8 5

56

Input n, m: << 5 2

10

Input n, m: << 7 0

1

Input n, m: << ^Z

Job Done : )

本次实验完成后，请填写实验完成情况汇总表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 题号 | 完成情况 | 知识点掌握情况 | 难度系数 | 测试评价 |
| 1 | 完成 | 函数的定义与调用 | 易 | 了解，掌握 |
| 2 | 完成 | 函数参数的传递及函数的嵌套调用 | 中 | 掌握 |
| 3 | 完成 | 函数的递归调用 | 难 | 理解 |
| 4 | 完成 | 全局变量与局部变量及变量的存储类型 | 易 | 了解，掌握 |
| 5 | 完成 | 函数的编程应用 | 中 | 应用 |