**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15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3 数组字符串与结构** | | | | | **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\*\*\*)

**实验A 数组字符串**

**一、实验目的及要求**

1．了解数组与数学矩阵的关系；

2．掌握数组的定义、引用与初始化；

3．理解二维数组数据的存储顺序；

4．理解并掌握冒泡排序法、选择排序法；

5．掌握字符数组与字符串的定义、引用；

6．了解字符串处理函数的用法。

**二、实验设备与平台**

1. 实验设备：计算机；

2. 平台：Windows 10操作系统，Clang 5.0.0。

**三、实验内容及步骤**

1．排序算法

排序算法有很多种，如冒泡排序、选择排序等。

1）冒泡排序法：相邻元素比较大小并发生交换，使最大值（最小值）“浮出”到数组尽头。注意比较冒泡排序和选择排序的差异

2．**实验3\_1：实现选择排序**

#include <cstdio>

#include <iostream>

#include <algorithm>

using namespace std;

#define M 10

int main()

{

int a[M];

printf("Please give me an array with 10 elements: \n");

for (int i = 0; i < M; i++) cin >> a[i];

printf("Unsorted：\n");

for (int i = 0; i < M; i++)

{

if (i) cout << " ";

cout << a[i];

}

cout << endl;

for (int i = 0; i < M - 1; i++)

{

int mn = i;

for (int j = i + 1; j < M; j++)

if (a[j] < a[mn]) mn = j;

swap(a[mn], a[i]);

}

printf("Sorted: \n");

for (int i = 0; i < M; i++)

{

if (i) cout << " ";

cout << a[i];

}

cout << endl;

}

程序调试要求：

1. 运行程序，任意输入10个整数，察看并分析程序执行结果；

10个整数按非降序输出

1. 运行程序，输入多于10个的整数，察看并分析程序执行结果；

10个整数按非降序输出

1. 运行程序，输入一组升序排列的有序整数，察看并分析程序执行结果；

10个整数按非降序输出

1. 运行程序，输入一组降序排列的有序整数，察看并分析程序执行结果。

10个整数按非降序输出

3．**实验3\_2：**有序数列的数据插入问题

这个问题的解决方案包括3个关键步骤：1）确定要插入数据的位置。这步操作使用的方法有很多种，如可以使用折半查找比较的方法，也可以使用顺序比较的方法；2）将自插入位置开始后的所有数据都向后移动一个位置，以便空出要插入数据的位置。若插入数据位置在原有所有数据之后，该步骤可省略，若插入数据位置在第一个数据之前，所有数据依次后移一个位置；3）将要插入的数据存储在该空位置上。

#include <iostream>

using namespace std;

#define M 10

int main()

{

int a[M + 1] = {10, 20, 30, 40, 50, 60, 70, 80, 90, 99};

cout << "Original array: \n";

for (int i = 0; i < M; i++)

{

if (i) cout << " ";

cout << a[i];

}

cout << endl;

cout << "Please give me an element,\n";

cout << "I will insert it into the correct location in the array. : )\n";

int tmp; cin >> tmp;

int pos;

for (pos = 0; tmp > a[pos] && pos < M; pos++); // 注意pos < M ！

for (int i = M; i > pos; i--) swap(a[i], a[i - 1]);

a[pos] = tmp;

cout << "New array: \n";

for (int i = 0 ; i < M + 1; i++)

{

if (i) cout << " ";

cout << a[i];

}

cout << endl;

}

程序调试要求：

1）运行程序，输入一个比所有原始数据都小的数，察看并分析程序执行结果；

2）运行程序，输入一个比所有原始数据都大的数，察看并分析程序执行结果；

3）运行程序，输入一个介于原始数据最小数和最大数之间的整数，察看并分析程序执行结果；

4）运行程序，输入原始数据中最小的数，察看并分析程序执行结果；

5）运行程序，输入原始数据中最大的数，察看并分析程序执行结果；

6）运行程序，输入原始数据中间的任何一个数，察看并分析程序执行结果。

无论如何输出，都有包括输入数据和原始数据在内的11个整数按非降序输出。

4．**实验3\_3：**不用字符串处理函数将两个字符串连接，即将字符串b连接到字符串a的后面，变成字符串c。

思考：如果将字符串S2连接到字符串S1的后面，且不需要第三个字符串变量作中间过度，如何实现？编程实现并测试。

#include <cstdio>

#include <iostream>

using namespace std;

/\* 将src字符串复制到tar字符串后面，用length指明src字符串最大可容纳的字符的个数 \*/

char \* my\_strcat(char \* const tar, const char \* src, int length)

{

int i = 0, j = 0;

while (i < length && tar[i] != '\0') i++;

while (i < length && src[j] != '\0') tar[i++] = src[j++];

while (i <= length) tar[i++] = '\0';

return tar;

}

int main()

{

char str[53] = "This is ";

char str1[] = "Lightyears speaking.";

char str2[] = "\n";

char str3[] = "He likes computer games developing.";

my\_strcat(str, str1, sizeof(str) - 1);

my\_strcat(str, str2, sizeof(str) - 1);

my\_strcat(str, str3, sizeof(str) - 1);

printf("%s", str);

printf("\n(Max capacity of c-string: %zd)\n", sizeof(str) - 1);

}

**实验B结构体实验**

**一、实验目的**

1．掌握结构的概念；

2．学会通过结构成员来使用结构；

3．学会通过函数的参数使用结构；

4．学会结构变量成员值的输出。

**二、实验设备与平台**

1. 实验设备：计算机；

2. 平台：Windows 10操作系统，Clang 5.0.0，G++ 6.3.0。

**三、实验内容**

1.定义一个结构体变量（包括年、月、日），计算该日在本年中为第几天？（注意考虑闰年问题），要求写一个函数days，实现上面的计算。由主函数将年月日传递给days函数，计算后将日子传递回主函数输出。

#include <iostream>

using namespace std;

const int month[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

struct date

{

int yy, mm, dd;

};

date current();

int days(date);

int main()

{

date now = current();

cout << "Day index: " << days(now) << endl;

}

date current()

{

int year, month, day;

cout << "Please tell me a date : )\n";

cout << "For example: 2017 11 12\n";

while (!(cin >> year >> month >> day)) {

while (getchar() != '\n');

cout << "Can't reach you.... : (\n";

cout << "Please seperate year, month and day by a white-space\n";

cout << "For example: 2017 11 12\n";

cin.clear();

}

date tmp = {year, month, day};

return tmp;

}

int days(date d)

{

bool isleap = (d.yy % 100) ? (d.yy % 4 ? false : true) : (d.yy % 400 ? false : true);

int tot = d.dd;

for (int i = 1; i < d.mm; i++) tot += month[i];

if (isleap && d.mm > 2) tot++;

return tot;

}

2.编写一个程序，处理5个学生的信息，每个学生的信息包括：学号、姓名、三门成绩。要求从键盘输入学生的数据，并输出成绩表（包括每个学生的学号、姓名、三门成绩及平均分数），并输出平均分在前3名的学生的姓名及平均分。

#include <iostream>

#include <string>

#include <iomanip>

using namespace std;

const int CNT = 5;

struct Student

{

int serial, grade[3];

string name;

double getAverage()

{

double sum = grade[0] + grade[1] + grade[2];

return sum / 3;

}

} student[CNT];

void eatline();

bool getInfo(int);

void makeTable();

Student select(Student [], int, int);

int main()

{

cout << "-- Simple Students Information Management System --\n";

cout << "Please give me the information of students: \n";

cout << "For example: Xiaoming 1706300001 69 79 99\n";

for (int i = 0; i < CNT; getInfo(i) ? i++ : i);

makeTable();

}

// 读取用户的一行输入并丢弃

void eatline()

{

while (getchar() != '\n');

}

// 读取学生信息，并在读取信息出错时给出提示

bool getInfo(int idx)

{

printf("Student #%d: ", idx + 1);

int &serial = student[idx].serial;

string &name = student[idx].name;

int \* grade = student[idx].grade;

if (!(cin >> name >> serial >> grade[0] >> grade[1] >> grade[2])) {

cout << "Can't reach you. I understand only input like 'Xie 1706300001 69 79 99'\n";

cin.clear();

eatline();

return false;

}

if (!(student[idx].serial <= 9999999999 && student[idx].serial >= 1000000000))

{

cout << "Invalid Serial: " << student[idx].serial << endl;

cout << "Serial ranges from 1000000000 to 9999999990. e.g. 1706300001" << endl;

eatline();

return false;

}

eatline();

return true;

}

// 返回n个学生中平均成绩排名第k的学生

Student select(Student stu[], int n, int k)

{

double privot = stu[0].getAverage();

int left = 1, right = n - 1;

while (true)

{

while (stu[left].getAverage() > privot && left + 1 < n) left++;

while (stu[right].getAverage() < privot && right - 1 >= 0) right--;

if (left < right) swap(stu[left], stu[right]);

else break;

}

int nleft = right;

if (nleft == k - 1) return stu[0];

else if (nleft < k - 1) return select(stu + nleft + 1, n - nleft - 1, k - nleft - 1);

else return select(stu + 1, nleft, k);

}

/\*\*

\* 选择问题算法潜在编程问题警告

\*

\* L78，L79是否取得等号无关紧要，重点是应该保证对数组的访问不会越界。

\* 越界通常会导致奇特的Stack Overflow问题

\*\*/

void makeTable()

{

string header = "| NAME";

string slash = "| -";

int length = header.size();

for (int i = 0; i < CNT; i++) length = max(length, (int)student[i].name.size());

header.resize(length + 2, ' '); slash.resize(length + 2, '-');

for (int i = 0; i < CNT; i++) student[i].name.resize(length, ' ');

header += " | SERIAL | GRADE1 | GRADE2 | GRADE3 | AVERAGE |\n";

slash += " | ---------- | ------ | ------ | ------ | ------- |\n";

system("cls");

cout << "CHART:\n";

cout << header << slash;

for (int i = 0; i < CNT; i++)

{

cout << "| " << student[i].name << " | ";

cout << student[i].serial << " | ";

cout << setw(6) << student[i].grade[0] << " | " << setw(6) << student[i].grade[1] << " | ";

cout << setw(6) << student[i].grade[2] << " | ";

cout.setf(ios\_base::fixed);

cout << setw(7) << setprecision(2) << student[i].getAverage() << " |\n";

}

cout << endl;

cout << "TOP STUDENTS: \n";

cout << "1. " << select(student, CNT, 1).name << " (" << select(student, CNT, 1).getAverage() << ")\n";

cout << "2. " << select(student, CNT, 2).name << " (" << select(student, CNT, 2).getAverage() << ")\n";

cout << "3. " << select(student, CNT, 3).name << " (" << select(student, CNT, 3).getAverage() << ")\n";

}

**四、实验指导**

1．某一年份是否为闰年的条件是，或者该年份能被4整除但却不能被100整除，或者该年份能被400整除。

2．闰年对天数影响是使2月的天数从28天增为29天。

3．注意结构体变量与其成员之间的“.”或“->”运算符的使用。

4．结构体作为函数参数时宜用指针形式，并注意实际参数的调用形式。

**五、实验结果**

1．调试闰年与非闰年在同一日期的运行结果, 观察并且写出输出结果。

例如输入：2004/05/31

输出：Day index: 151

输入：2005/05/31

输出：Day index: 133

2. 调试处理5个学生信息的程序，输出平均分在前3名的学生的姓名及平均分。

3. 现补充对部分源程序的算法说明如下：

1）**日期计算程序** 要求从输入中读取年月日的信息，然后输出对应的日期是对应的年份中的第几天。注意输入时使用斜杠作为分隔符，并注意到对闰年的判断。

2）**学生信息管理** 常见作业题型，算法上没有难点。只是需要使用结构体来实现。