

实验10

1 (10分)

有一组关于学生成绩的信息，编写函数max，该函数返回值为分数最高的学生的信息（包括学号和分数）。再编写主函数对其进行调用并输出最高分者的信息。假设结构类型定义为：

```
struct student  
  
{  
  
    char *num;  
  
    int score;  
  
};
```


提示：num定义为指针类型，注意向其复制内容之前要先为其申请空间。

```
1  #include <iostream>  
2  #include <cstring>  
3  using namespace std;  
4  
5  // 定义结构体 student  
6  struct student {  
7      char *num; // 学号，使用指针  
8      int score; // 成绩  
9  };  
10  
11 // 定义max函数，返回分数最高的学生的信息  
12 student* max(student s[], int n) {  
13     student *maxStudent = &s[0]; // 假设第一个学生是分数最高的  
14     for (int i = 1; i < n; i++) {  
15         if (s[i].score > maxStudent->score) {  
16             maxStudent = &s[i]; // 更新最高分学生  
17         }  
18     }  
19     return maxStudent;  
20 }  
21  
22 int main() {  
23     // 定义学生数组，假设有5个学生  
24     student students[5];
```

```

25 // 给学生信息赋值
26 students[0].num = new char[10];
27 strcpy(students[0].num, "1001");
28 students[0].score = 85;
29
30 students[1].num = new char[10];
31 strcpy(students[1].num, "1002");
32 students[1].score = 92;
33
34 students[2].num = new char[10];
35 strcpy(students[2].num, "1003");
36 students[2].score = 78;
37
38 students[3].num = new char[10];
39 strcpy(students[3].num, "1004");
40 students[3].score = 95;
41
42 students[4].num = new char[10];
43 strcpy(students[4].num, "1005");
44 students[4].score = 88;
45
46 // 调用max函数, 获取分数最高的学生
47 student* topStudent = max(students, 5);
48
49 // 输出分数最高的学生信息
50 cout << "The student with the highest score is:" << endl;
51 cout << "Student ID: " << topStudent->num << ", Score: " << topStudent-
    >score << endl;
52
53 // 释放为num分配的内存
54 for (int i = 0; i < 5; i++) {
55     delete[] students[i].num;
56 }
57
58     system("pause");
59     return 0;
60 }

```

 D:\2024\课内\C++\第十章\10.1\Debug\10.1.exe

```

The student with the highest score is:
Student ID: 1004, Score: 95
请按任意键继续. . .

```

2 (10分)


编写程序，定义一个日期结构变量，计算某日期是本年度的第几天。提示：为简单起见，可定义一个存放12个月中每个月总天数的数组。

代码解析：

1. **结构体 Date**：定义了一个日期结构体，包含年份 `year`、月份 `month` 和日期 `day`。
2. **数组 daysInMonth**：存储了非闰年每个月的天数。注意，2月的天数为28天，假设输入的日期是非闰年。
3. **dayOfYear 函数**：这个函数计算某日期是本年度的第几天。它通过累加当前日期之前的所有月份的天数，再加上当前月的日期，得到结果。
4. **输入输出**：在 `main` 函数中输入一个日期，调用 `dayOfYear` 函数计算并输出该日期是本年度的第几天。

```
1 #include <iostream>
2 using namespace std;
3
4 // 定义日期结构体
5 struct Date {
6     int year; // 年
7     int month; // 月
8     int day; // 日
9 };
10
11 // 函数：计算某日期是本年度的第几天
12 int dayOfYear(Date d) {
13     // 每个月的天数（假设非闰年）
14     int daysInMonth[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
15
16     int dayOfYear = 0;
17
18     // 累加当前月份之前的所有月份的天数
19     for (int i = 0; i < d.month - 1; i++) {
20         dayOfYear += daysInMonth[i];
21     }
22
23     // 加上当前月的天数
24     dayOfYear += d.day;
25
26     return dayOfYear;
27 }
28
29 int main() {
```

```
30     Date d;
31
32     // 输入日期
33     cout << "Enter year: ";
34     cin >> d.year;
35     cout << "Enter month: ";
36     cin >> d.month;
37     cout << "Enter day: ";
38     cin >> d.day;
39
40     // 计算并输出结果
41     int result = dayOfYear(d);
42     cout << "The date is the " << result << "th day of the year." << endl;
43
44     system("pause");
45     return 0;
46 }
```

 D:\2024\课内\C++\第十章\10.2\Debug\10.2.exe

```
Enter year: 2004
Enter month: 05
Enter day: 08
The date is the 128th day of the year.
请按任意键继续. . .
```