实验12

1 (10分)

编写程序,从键盘输入一串字符,要求将该串字符的倒序串先写入到文件f1.txt中,然后再将原字符串的内容接到该文件的末尾。例如,假设从键盘输入的字符串为"How do you do?",则文件f1.txt的内容为:

?od uoy od woHHow do you do?

```
1 #include <stdio.h>
2 #include <string.h>
3 #include <iostream>
4
5 int main() {
      char str[100], reversedStr[100];
      FILE *fp;
7
8
      // 从键盘输入字符串
9
      printf("请输入一串字符:");
10
11
      fgets(str, sizeof(str), stdin);
      str[strcspn(str, "\n")] = '\0'; // 去掉换行符
12
13
14
      // 获取字符串的倒序
      int len = strlen(str);
15
      for (int i = 0; i < len; i++) {
16
          reversedStr[i] = str[len - i - 1];
17
18
      }
      reversedStr[len] = '\0'; // 确保倒序字符串正确结束
19
20
21
      // 打开文件f1.txt进行写入操作
22
      fp = fopen("f1.txt", "w");
      if (fp == NULL) {
23
          printf("无法打开文件!\n");
24
          return 1;
25
26
      }
27
      // 将倒序字符串写入文件
28
29
      fputs(reversedStr, fp);
30
      // 将原字符串追加到文件末尾
31
      fputs(str, fp);
32
33
```

程序说明:

- 1. 输入字符串:通过 fgets() 函数从键盘输入字符串。
- 2. **倒序字符串**:通过遍历字符串并将字符倒序存储在 reversedStr 数组中。
- 3. **文件操作**:使用 fopen("f1.txt", "w") 打开文件进行写入,写入倒序字符串后,再使用 fputs() 追加原字符串内容。
- 4. **关闭文件**:使用 fclose() 关闭文件。
- D:\2024\课内\C++\第十二章\12.1\Debug\12.1.exe

```
青输入一串字符: dugshi
操作成功! 文件f1.txt已更新。
请按任意键继续. . .
```

2 (10分)

用记事本建立一个文本文件f2.txt,在该文件中任意存放一组整数。编写程序统计该文件中正整数、负整数和零的个数。(提示:用fscanf函数读取文件中的数据)

```
1 #include <stdio.h>
2 #include <iostream>
3
4 int main() {
       FILE *fp;
5
       int number;
6
       int positiveCount = 0, negativeCount = 0, zeroCount = 0;
7
8
       // 打开文件f2.txt进行读取
       fp = fopen("f2.txt", "r");
10
       if (fp == NULL) {
11
           printf("无法打开文件f2.txt! \n");
12
```

```
13
           return 1;
       }
14
15
       // 使用fscanf逐个读取文件中的整数
16
       while (fscanf(fp, "%d", &number) == 1) {
17
           if (number > 0) {
18
               positiveCount++; // 统计正整数
19
           } else if (number < 0) {</pre>
20
21
               negativeCount++; // 统计负整数
           } else {
22
                             // 统计零
23
              zeroCount++;
           }
24
       }
25
26
       // 输出统计结果
27
28
       printf("正整数的个数: %d\n", positiveCount);
       printf("负整数的个数: %d\n", negativeCount);
29
30
       printf("零的个数: %d\n", zeroCount);
31
       // 关闭文件
32
33
       fclose(fp);
34
           system("pause");
35
       return 0;
36
37 }
```

程序说明:

- 1. **打开文件**:使用 fopen("f2.txt", "r") 打开文件进行读取。如果文件无法打开,程序会输出错误信息并退出。
- 2. **读取整数**:通过 fscanf(fp, "%d", &number) 循环读取文件中的每个整数。 fscanf() 函数返回读取的项目个数,如果成功读取到一个整数,它返回 1。
- 3. 分类统计:根据整数的大小(正数、负数、零),分别增加对应的计数变量。
- 4. 输出结果: 最后打印出文件中正整数、负整数和零的个数。

3 (10分)

将从键盘输入的N个学生的学号和成绩存入到文件student.dat中。再从文件中读取学生的信息,求出最高分、最低分和总分。N可通过符号常量自行定义大小。

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define N 5 // 定义学生的数量
5
6 // 定义学生结构体
7 struct Student {
      char id[20]; // 学号
      float score; // 成绩
9
10 };
11
12 int main() {
      FILE *fp;
13
14
      struct Student students[N];
      float maxScore, minScore, totalScore = 0.0;
15
16
      // 从键盘输入N个学生的学号和成绩
17
      printf("请输入 %d 个学生的学号和成绩: \n", N);
18
      for (int i = 0; i < N; i++) {
19
          printf("请输入第 %d 个学生的学号: ", i + 1);
20
          scanf("%s", students[i].id);
21
22
          printf("请输入第 %d 个学生的成绩: ", i + 1);
          scanf("%f", &students[i].score);
23
24
      }
25
      // 将学生信息写入文件 student.dat
26
      fp = fopen("student.dat", "wb"); // 二进制写入模式
27
      if (fp == NULL) {
28
          printf("无法创建文件 student.dat! \n");
29
30
          return 1;
      }
31
32
      fwrite(students, sizeof(struct Student), N, fp);
      fclose(fp);
33
34
      // 从文件中读取学生信息
35
      fp = fopen("student.dat", "rb"); // 二进制读取模式
36
37
      if (fp == NULL) {
          printf("无法打开文件 student.dat! \n");
38
          return 1;
39
      }
40
41
```

```
fread(students, sizeof(struct Student), N, fp);
42
       fclose(fp);
43
44
       // 初始化最大值和最小值
45
       maxScore = minScore = students[0].score;
46
47
       // 计算最高分、最低分和总分
48
       for (int i = 0; i < N; i++) {
49
50
           if (students[i].score > maxScore) {
               maxScore = students[i].score;
51
52
           if (students[i].score < minScore) {</pre>
53
               minScore = students[i].score;
54
           }
55
           totalScore += students[i].score;
56
57
       }
58
59
       // 输出计算结果
       printf("\n学生信息: \n");
60
       for (int i = 0; i < N; i++) {
61
           printf("学号: %s, 成绩: %.2f\n", students[i].id, students[i].score);
62
       }
63
       printf("\n最高分: %.2f\n", maxScore);
64
       printf("最低分: %.2f\n", minScore);
65
       printf("总分: %.2f\n", totalScore);
66
67
       return 0;
68
69 }
```

程序说明:

- 1. 结构体定义:使用 struct Student 定义学生的学号(字符串)和成绩(浮点数)。
- 2. 输入学生信息:
 - 。 用户从键盘输入 N 个学生的学号和成绩。
- 3. 写入文件:
 - 使用 fwrite 将学生信息写入二进制文件 student.dat。
- 4. 读取文件:
 - 使用 fread 从文件中读取数据。
- 5. 计算结果:
 - 。 遍历学生数据,找出最高分、最低分并计算总分。
- 6. 输出结果:

• 输出学生的信息、最高分、最低分和总分。

■ D:\2024\课内\C++\第十二章\12.2\Debug\12.2.exe

```
请输入 5 个学生的学号:1
请输入第 1 个学生的学号:1
请输入第 2 个学生的成绩: 67
请输入第 2 个学生的成绩: 68
请输入第 3 个学生的成绩: 68
请输入第 3 个学生的成绩: 69
请输入第 4 个学生的成绩: 96
请输入第 5 个学生的成绩: 96
请输入第 5 个学生的成绩: 98
学生信息:
学号: 1, 成绩: 67.00
学号: 2, 成绩: 68.00
学号: 3, 成绩: 69.00
学号: 5, 成绩: 98.00
最高分: 98.00
最高分: 98.00
请按任意键继续. . .
```