

河南工业大学

程序设计实践 实验报告

田劲锋

计算机 1303班 201316920311

2014 年 6 月 30 日

目录

1 函数与程序结构	1	2.4 P145 编程题 (1)	9
1.1 P123 调试示例	1	2.5 P145 编程题 (2)	10
1.2 P132 编程题 (5)	3	2.6 P145 编程题 (3)	10
1.3 P132 编程题 (7)	4		
1.4 P132 编程题 (8)	5	3 文件	12
2 指针进阶	6	3.1 P148 改错题	12
2.1 P138 编程题 (2)	6	3.2 P150 编程题 (3)	12
2.2 P138 编程题 (3)	6	3.3 P150 编程题 (7)	14
2.3 P140 调试示例	7	4 实验总结	17

1 函数与程序结构

1.1 P123 调试示例

主要是预编译指令的写法，和工程的文件包含。

Listing 1: error10_1_main.c

```
1 #include <stdio.h>
2 #include <math.h>
3
4 #define PI M_PI
5
6 #include "error10_1_cal.c"
7 #include "error10_1_vol.c"
8
9 int main()
10 {
11     int sel;
12     while (1) {
13         printf("    1 - 计算球体体积\n");
14         printf("    2 - 计算圆柱体体积\n");
```

```

15     printf("    3 - 计算圆锥体体积\n");
16     printf(" 其他 - 退出程序运行\n");
17     printf("请输入对应指令: ");
18     scanf("%d", &sel);
19     if (sel < 1 || sel > 3) {
20         break;
21     } else {
22         cal(sel);
23     }
24 }
25 return 0;
26 }

```

Listing 2: error10_1_cal.c

```

1 void cal(int sel)
2 {
3     double vol_ball();
4     double vol_cylind();
5     double vol_cone();
6     switch (sel) {
7         case 1:
8             printf("球体体积为: %.2f\n", vol_ball());
9             break;
10        case 2:
11            printf("圆柱体体积为: %.2f\n", vol_cylind());
12            break;
13        case 3:
14            printf("圆锥体体积为: %.2f\n", vol_cone());
15            break;
16    }
17 }

```

Listing 3: error10_1_vol.c

```

1 double vol_ball()
2 {
3     double r;
4     printf("请输入球体的半径: ");
5     scanf("%lf", &r);
6     return 4.0 / 3.0 * PI * r * r * r;
7 }
8
9 double vol_cylind()
10 {
11     double r, h;
12     printf("请输入圆柱体的底面半径和高: ");
13     scanf("%lf%lf", &r, &h);
14     return PI * r * r * h;
15 }
16
17 double vol_cone()
18 {
19     double r, h;
20     printf("请输入圆锥体的底面半径和高: ");
21     scanf("%lf%lf", &r, &h);

```

```

22     return PI * r * r * h / 3.0;
23 }

```

运行结果:

```

1 - 计算球体体积
2 - 计算圆柱体体积
3 - 计算圆锥体体积
其他 - 退出程序运行
请输入对应指令: 1
请输入球体的半径: 2
球体体积为: 33.51
1 - 计算球体体积
2 - 计算圆柱体体积
3 - 计算圆锥体体积
其他 - 退出程序运行
请输入对应指令: 3
请输入圆锥体的底面半径和高: 2.4 3
圆锥体体积为: 18.10
1 - 计算球体体积
2 - 计算圆柱体体积
3 - 计算圆锥体体积
其他 - 退出程序运行
请输入对应指令: 2
请输入圆柱体的底面半径和高: 2.4 3
圆柱体体积为: 54.29
1 - 计算球体体积
2 - 计算圆柱体体积
3 - 计算圆锥体体积
其他 - 退出程序运行
请输入对应指令: 4

```

另外, 一些复杂的预编译指令在 Microsoft Visual C++ 6.0 上是无法编译通过的, 需要使用符合 C89 标准的编译器如 GCC。

1.2 P132 编程题 (5)

实现阿克曼函数:

$$\text{Ack}(m, n) = \begin{cases} n + 1, & m = 0, \\ \text{Ack}(m - 1, 1), & n = 0, \\ \text{Ack}(m - 1, \text{Ack}(m, n - 1)), & m > 0 \text{ 且 } n > 0. \end{cases}$$

使用递归可以实现:

Listing 4: t5.c

```

1 #include <stdio.h>
2
3 #define inputi(x) ({ \
4     printf("请输入 " #x ": "); \

```

```

5     scanf("%i", &x); \
6 })
7
8 int Ack(int m, int n)
9 {
10     if (m == 0) {
11         return n + 1;
12     } else if (n == 0) {
13         return Ack(m - 1, 1);
14     } else if (m > 0 && n > 0) {
15         return Ack(m - 1, Ack(m, n - 1));
16     } else {
17         return 1;
18     }
19 }
20
21 int main()
22 {
23     int m, n;
24     inputi(m);
25     inputi(n);
26     printf("Ackerman(%d, %d) = %d\n", m, n, Ack(m, n));
27     return 0;
28 }

```

运行结果:

请输入 m: 2

请输入 n: 3

Ackerman(2, 3) = 9

1.3 P132 编程题 (7)

求斐波那契数列项:

$$f(n) = \begin{cases} 0, & n = 0, \\ 1, & n = 1, \\ f(n-2) + f(n-1), & n > 1. \end{cases}$$

使用递归完成:

Listing 5: t7.c

```

1 #include <stdio.h>
2
3 #define inputi(x) ({ \
4     printf("请输入 " #x ": "); \
5     scanf("%i", &x); \
6 })
7
8 int f(int n)
9 {
10     if (n > 1) {
11         return f(n - 2) + f(n - 1);
12     } else {

```

```

13         return n;
14     }
15 }
16
17 int main()
18 {
19     int n;
20     inputi(n);
21     printf("fib(%d) = %d\n", n, f(n));
22     return 0;
23 }

```

运行结果:

```

请输入 n: 6
fib(6) = 8

```

1.4 P132 编程题 (8)

实现二进制转换函数 `dectobin(n)`，其中 n 是正整数。
二除取余，倒序输出。仍然可以使用递归：

Listing 6: t8.c

```

1  #include <stdio.h>
2
3  #define inputi(x) ({ \
4      printf("请输入 " #x ": "); \
5      scanf("%i", &x); \
6  })
7
8  void dectobin(int n)
9  {
10     if (n > 0) {
11         dectobin(n / 2);
12         printf("%d", n % 2);
13     }
14 }
15
16 int main()
17 {
18     int n;
19     inputi(n);
20     dectobin(n);
21     printf("\n");
22     return 0;
23 }

```

运行结果:

```

请输入 n: 100
1100100

```

2 指针进阶

指针虽好，减少滥用。

2.1 P138 编程题 (2)

输入星期的英文单词，输出对应序号。

顺序查找比较即可：

Listing 7: t2.c

```
1 #include <stdio.h>
2 #include <string.h>
3
4 const char *week[] = {
5     "Sunday",
6     "Monday",
7     "Tuesday",
8     "Wednesday",
9     "Thursday",
10    "Friday",
11    "Saturday"
12 };
13
14 int getindex(const char **list, size_t n, const char *key)
15 {
16     int i = 0;
17     for (i = 0; i < n; i++) {
18         if (strcmp(list[i], key) == 0) {
19             return i + 1;
20         }
21     }
22     return -1;
23 }
24
25 int main()
26 {
27     char s[16];
28     scanf("%s", s);
29     printf("%d\n", getindex(week, 7, s));
30     return 0;
31 }
```

运行结果：

Tuesday
3

Kinyoubi
-1

星期五（金曜日「きんようび」）

2.2 P138 编程题 (3)

统计输入的最长字符串。

小心指针类型的问题，分清楚指针数组和数组指针。这里我每读入一个字符串，就分配相应大小的内存：

Listing 8: t3.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 #define inputi(x) ({ \
6     printf(#x "="); \
7     scanf("%i", &x); \
8 })
9
10 inline int max(int a, int b) { return a > b ? a : b; }
11
12 int max_len(char *s[], int n)
13 {
14     int i = 0, l = 0;
15     for (i = 0; i < n; i++) {
16         l = max(l, strlen(s[i]));
17     }
18     return l;
19 }
20
21 int main()
22 {
23     int n, i;
24     char *s[16], t[256];
25     inputi(n);
26     for (i = 0; i < n; i++) {
27         scanf("%s", t);
28         s[i] = (char *) calloc(strlen(t), sizeof(t));
29         strcpy(s[i], t);
30     }
31     printf("%d\n", max_len(s, n));
32     return 0;
33 }
```

运行结果：

n=4	
ao	# 蓝色（青「あお」）
kiiro	# 黄色（黄色「きいろ」）
aka	# 红色（赤「あか」）
midori	# 绿色（緑「みどり」）
6	

2.3 P140 调试示例

调试一个有问题的单向链表程序。

错误代码没有处理开始时表为空的情况，输出时没有输出最有一个结点。改正如下：

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 struct stud_node {
6     int num;
7     char name[20];
8     int score;
9     struct stud_node *next;
10 };
11
12 int main()
13 {
14     struct stud_node *head, *tail, *p;
15     int num, score;
16     char name[20];
17     int size = sizeof(struct stud_node);
18     head = tail = NULL;
19     printf("Input num, name and score:\n");
20     scanf("%d", &num);
21     while (num != 0) {
22         p = malloc(size);
23         scanf("%s%d", name, &score);
24         p->num = num;
25         strcpy(p->name, name);
26         p->score = score;
27         p->next = NULL;
28         if (head == NULL)                /* 首指针 */
29             head = p;
30         else
31             tail->next = p;
32         tail = p;
33         scanf("%d", &num);
34     }
35     for (p = head; p != NULL; p = p->next) /* 走到头 */
36         printf("%d %s %d\n", p->num, p->name, p->score);
37     return 0;
38 }
```

运行结果:

Input num, name and score:

```
1 zhang 78
2 wang 80
3 Li 75
4 zhao 85
0
1 zhang 78
2 wang 80
3 Li 75
4 zhao 85
```


2.4 P145 编程题 (1)

输入学生建立单链表，输出大于指定分数的学生信息。

可以直接改用调试示例 Listing 9 的代码，只需稍作改动即可：

Listing 10: t1.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  typedef struct student {
6      int id;
7      char name[20];
8      int score;
9      struct student *next;
10 } student;
11
12 int main()
13 {
14     student *head, *tail, *p;
15     int id, score;
16     char name[20];
17     int size = sizeof(student);
18     head = tail = NULL;
19     scanf("%d", &id);
20     while (id != 0) {
21         p = malloc(size);
22         scanf("%s%d", name, &score);
23         p->id = id;
24         strcpy(p->name, name);
25         p->score = score;
26         p->next = NULL;
27         if (head == NULL)
28             head = p;
29         else
30             tail->next = p;
31         tail = p;
32         scanf("%d", &id);
33     }
34     scanf("%d", &score);                                /* 输入成绩 */
35     for (p = head; p != NULL; p = p->next)
36         if (p->score >= score)                            /* 判断输出 */
37             printf("%d %s %d\n", p->id, p->name, p->score);
38     return 0;
39 }
```

运行结果：

```
1 zhang 78
2 wang 80
3 Li 75
4 zhao 85
0
80
```

2 wang 80
4 zhao 85

2.5 P145 编程题 (2)

输入若干整数，逆序建立链表，并输出。
头插法即可解决：

Listing 11: t2.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 typedef struct list {
5     int num;
6     struct list *next;
7 } list;
8
9 int main()
10 {
11     list *head, *tail, *p;
12     int num;
13     int size = sizeof(list);
14     head = tail = NULL;
15     while (scanf("%d", &num), num != -1) {
16         p = malloc(size);
17         p->num = num;
18         if (tail == NULL)
19             tail = p;
20         p->next = head;
21         head = p;
22         // fprintf(stderr, "%p->%p\n", p, p->next);
23     }
24     for (p = head; p != NULL; p = p->next)
25         printf("%d ", p->num);
26     printf("\n");
27     return 0;
28 }
```

运行结果：

```
1 2 3 4 5 6 7 -1
7 6 5 4 3 2 1
```

2.6 P145 编程题 (3)

输入若干整数，建立链表，删除其中的偶数输出。

尾插法建立链表，遍历删除。只能删除下一个元素，因为会单链表遍历会丢失上一节点指针，最后作为补救措施，特别删除首结点，这个是因为首指针本身也存储数据的原因。代码如下：

Listing 12: t3.c

```
1 #include <stdio.h>
```

```

2  #include <stdlib.h>
3
4  typedef struct list {
5      int num;
6      struct list *next;
7  } list;
8
9  int main()
10 {
11     list *head, *tail, *p, *q;
12     int num;
13     int size = sizeof(list);
14     head = tail = NULL;
15     while (scanf("%d", &num), num != -1) {
16         p = malloc(size);
17         p->num = num;
18         p->next = NULL;
19         if (head == NULL)
20             head = p;
21         else
22             tail->next = p;
23         tail = p;
24     }
25     for (p = head; p && p->next; p = p->next) {
26         if (p->next->num % 2 == 0) {
27             q = p->next;
28             p->next = p->next->next;
29             if (p->next == NULL)
30                 tail = p;
31             free(q);
32         }
33     }
34     if (head->num % 2 == 0) {
35         q = head;
36         head = head->next;
37         free(q);
38     }
39     for (p = head; p != NULL; p = p->next)
40         printf("%d ", p->num);
41     printf("\n");
42     return 0;
43 }

```

运行结果:

```

1 2 3 4 5 6 7 -1
1 3 5 7

```

```

0 1 2 3 4 5 6 7 -1
1 3 5 7

```

```

1 2 3 4 5 6 7 8 -1
1 3 5 7

```

3 文件

3.1 P148 改错题

程序从文件中读入整数，累加后写入文件末尾。要求改正程序。

错误在注释中给出：

Listing 13: error12_2.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main()
5 {
6     FILE *fp;                                /* fp 是指针类型 */
7     int n, sum = 0;                          /* sum 应初始化 */
8
9     if ((fp = fopen("Int_Data.dat", "r+")) == NULL) {
10         perror("打开文件失败");             /* 打开方式为读并更新 */
11         exit(0);
12     }
13
14     while (fscanf(fp, "%d", &n) != EOF)      /* 读到文件尾 */
15         sum = sum + n;
16
17     fprintf(fp, "%d\n", sum);
18     fclose(fp);
19
20     return 0;
21 }
```

3.2 P150 编程题 (3)

读入学生信息，写到文件中，并计算平均分，输出。

考虑使用结构体存储数据，以二进制形式直接把结构体数据写入文件（fwrite()），然后把文件指针重置，直接读入二进制数据（fread()）到结构体数组中，显示到屏幕上。具体程序如下：

Listing 14: t3.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 typedef struct student {
6     int id;
7     char name[16];
8     int score[3];
9     int total, avg;
10 } student;
11
12 int main(int argc, char *argv[])
13 {
14     char *filename = "stu.db";
```

```

15     if (argc > 1)
16         filename = argv[1];
17     FILE *file = fopen(filename, "wb+");    /* 以二进制方式打开文件 */
18     if (file == NULL) {
19         perror("打开文件失败");
20         exit(errno);
21     }
22
23     student a[10];
24     int i = 0;
25     while (scanf("%d%s%d%d%d", &a[i].id, a[i].name,
26         &a[i].score[0], &a[i].score[1], &a[i].score[2]) > 0) {
27         a[i].total = a[i].score[0] + a[i].score[1] + a[i].score[2];
28         a[i].avg = a[i].total / 3;
29         i++;                                /* 边读入, 边计算 */
30     }
31     int n = i;
32     fwrite(a, sizeof(student), n, file);    /* 直接写出二进制数据 */
33
34     rewind(file);
35     student b[10];
36     fread(b, sizeof(student), n, file);    /* 直接读入二进制数据 */
37     fclose(file);
38
39     printf("学号\t 姓名\t 数学\t 语文\t 英语\t 总成绩\t 平均分\n");
40     for (i = 0; i < n; i++) {
41         printf("%d\t %s\t %d\t %d\t %d\t %d\t %d\n", b[i].id, b[i].name,
42             b[i].score[0], b[i].score[1], b[i].score[2],
43             b[i].total, b[i].avg);
44     }
45
46     return 0;
47 }

```

输入如下数据:

```

3020801 陈刚 81 75 82
3020802 王媛 87 68 85
3020803 李兵 73 84 80
3020804 曹新 76 81 74
3020805 方亮 83 75 71
3020806 何帆 89 78 91
3020807 季东 82 80 72
3020808 林海 72 76 88
3020809 盛天 89 87 76
3020810 高晶 93 86 85

```

屏幕显示如下结果:

学号	姓名	数学	语文	英语	总成绩	平均分
3020801	陈刚	81	75	82	238	79
3020802	王媛	87	68	85	240	80
3020803	李兵	73	84	80	237	79

3020804	曹新	76	81	74	231	77
3020805	方亮	83	75	71	229	76
3020806	何帆	89	78	91	258	86
3020807	季东	82	80	72	234	78
3020808	林海	72	76	88	236	78
3020809	盛天	89	87	76	252	84
3020810	高晶	93	86	85	264	88

之后以十六进制形式检查文件 `stu.db`，除了部分具体数据，可以看到有很多填充的0，这是编译器优化的结果。具体二进制数据如下：

```
0118 2e00 b3c2 b8d5 0000 6800 6f00 6d00
7e01 007f 5100 0000 4b00 0000 5200 0000
ee00 0000 4f00 0000 0218 2e00 cdf5 e6c2
0013 5a00 0000 0000 5800 0000 5700 0000
4400 0000 5500 0000 f000 0000 5000 0000
0318 2e00 c0ee b1f8 005f b655 fa7f 0000
1003 5a00 4900 0000 5400 0000 5000 0000
ed00 0000 4f00 0000 0418 2e00 b2dc d0c2
0000 0000 0000 0000 0000 5a00 4c00 0000
5100 0000 4a00 0000 e700 0000 4d00 0000
0518 2e00 b7bd c1c1 0000 0000 fa7f 0000
0200 0000 5300 0000 4b00 0000 4700 0000
e500 0000 4c00 0000 0618 2e00 bace b7ab
0000 0000 0000 0000 0000 0000 5900 0000
4e00 0000 5b00 0000 0201 0000 5600 0000
0718 2e00 bcbe b6ab 002d 4000 0000 0000
0200 0000 5200 0000 5000 0000 4800 0000
ea00 0000 4e00 0000 0818 2e00 c1d6 baa3
0026 4000 0000 0000 1c91 b655 4800 0000
4c00 0000 5800 0000 ec00 0000 4e00 0000
0918 2e00 caa2 ccec 002c 4000 0000 0000
0000 0000 5900 0000 5700 0000 4c00 0000
fc00 0000 5400 0000 0a18 2e00 b8df bea7
0026 4000 0000 0000 0200 0000 5d00 0000
5600 0000 5500 0000 0801 0000 5800 0000
```

当然，这样直接写出二进制的方法，虽然很快很方便，但是会遇到兼容性问题。

3.3 P150 编程题 (7)

把银行账户信息存入随机存取文件中，并能进行更新操作。

仍然使用二进制文件存放和读取数据，提供一个简单的交互界面：

Listing 15: `t7.c`

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
```

```

4  #include <ctype.h>
5
6  typedef struct account {
7      int id;
8      char name[16];
9      double bala;
10 } account;
11
12 account getbyid(int i, FILE *file) {
13     account t;
14     fseek(file, sizeof(account) * (i - 1), SEEK_SET);
15     fread(&t, sizeof(account), 1, file);
16     return t;
17 }
18
19 int main(int argc, char *argv[])
20 {
21     int i = 0, n = 0;
22     account a[10], t;
23     char c;
24     double v;
25
26     char *filename = "acct.db";
27     if (argc > 1)
28         filename = argv[1];
29     FILE *file = fopen(filename, "rb+");
30     if (file == NULL) {
31         perror("没有数据记录");
32         file = fopen(filename, "wb+");
33         goto input;
34     }
35
36     fseek(file, 0, SEEK_END);
37     n = ftell(file) / sizeof(account);
38     rewind(file);
39     fread(a, sizeof(account), n, file);
40
41     printf("目前文件中读出来 %d 个数据。 \n", n);
42     for (i = 0; i < n; i++) {
43         printf(" %d\t%8s\t%.2lf\n", a[i].id, a[i].name, a[i].bala);
44     }
45
46 input:
47     i = 0, v = 0;
48     do {
49         printf("输入账户编号退出(0): ");
50         scanf("%d", &i);
51         if (i > n) {
52             printf("没有编号为 %d 的用户\n", i);
53             printf("输入姓名新建(%d): ", t.id = i = ++n);
54             scanf("%s", t.name);
55             t.bala = 0;
56         } else if (i <= 0) {
57             break;
58         } else {

```

```

59         t = getbyid(i, file);
60         printf("查询到记录 %d:\n %d\t%8s\t%.2lf\n", i, t.id, t.name, t.bala);
61     }
62     printf("输入收支金额: ");
63     scanf("%lf", &v);
64     t.bala += v;
65     fseek(file, sizeof(account) * (i - 1), SEEK_SET);
66     fwrite(&t, sizeof(account), 1, file);
67     fflush(file);
68     printf("已写出记录 %d:\n %d\t%8s\t%.2lf\n", i, t.id, t.name, t.bala);
69 } while (i > 0);
70
71 fclose(file);
72 return 0;
73 }

```

写出数据时会刷新缓冲区，确保数据确实写入。

首次运行时，没有找到数据库文件，会自动创建：

没有数据记录: No such file or directory

输入账户编号(0退出): 3

没有编号为 3 的用户

输入姓名新建(1): 张三

输入收支金额: 1000

已写出记录 1:

1	张三	1000.00
---	----	---------

输入账户编号(0退出): 5

没有编号为 5 的用户

输入姓名新建(2): 李四

输入收支金额: 1300

已写出记录 2:

2	李四	1300.00
---	----	---------

输入账户编号(0退出): 6

没有编号为 6 的用户

输入姓名新建(3): 王五

输入收支金额: -100

已写出记录 3:

3	王五	-100.00
---	----	---------

输入账户编号(0退出): 0

再次运行，对已有的数据文件，还可以进行增改操作：

目前文件中读出来 3 个数据。

1	张三	1000.00
---	----	---------

2	李四	1300.00
---	----	---------

3	王五	-100.00
---	----	---------

输入账户编号(0退出): 1

查询到记录 1:

1	张三	1000.00
---	----	---------

输入收支金额: +800

已写出记录 1:


```

1          张三          1800.00
输入账户编号(0退出): 4
没有编号为 4 的用户
输入姓名新建(4): 六花
输入收支金额: 6666.66
已写出记录 4:
4          六花          6666.66
输入账户编号(0退出): 4
查询到记录 4:
4          六花          6666.66
输入收支金额: +60000
已写出记录 4:
4          六花          66666.66
输入账户编号(0退出): 3
查询到记录 3:
3          王五          -100.00
输入收支金额: -70
已写出记录 3:
3          王五          -170.00
输入账户编号(0退出): 0

```

最后一次运行，得到如下显示：

目前文件中读出来 4 个数据。

```

1          张三          1800.00
2          李四          1300.00
3          王五          -170.00
4          六花          66666.66
输入账户编号(0退出): 0

```

打开数据文件 acct.db，内容是这样的：

```

0100 0000 d5c5 c8fd 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0020 9c40
0200 0000 c0ee cbc4 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0050 9440
0300 0000 cdf5 cee5 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0040 65c0
0400 0000 c1f9 bba8 0000 0000 0000 0000
0000 0000 0000 0000 f628 5c8f aa46 f040

```

4 实验总结

本来这些内容就已经是会了的，这次课更多的是巩固和复习。这里面，函数是最简单的，包括一些预编译指令，都不是很艰深的内容。对于 C 来说，指针是其精华，但是如果不能很好地管理指针，造成的危害也很严重。指针的嵌套使用，包括指针常量、常量指针、指针数组、数组指针、字符串指针、指针字符串、文件指针等，都是比较不易理解和容易混淆的概念，即使拥有多年编程经验，也会在这里犯迷糊。还有一部分是文件，这个其实也不难，熟练掌握 `<stdio.h>` 里提供的函数和用法，就足够了。算是一个下午完成了这十几道题目吧，还是比较理想的。