

程序设计实践实验报告

田劲锋

201316920311

目录

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

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:

```
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:

```
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

2.4 P145 编程题 (1)

2.5 P145 编程题 (2)

2.6 P145 编程题 (3)

3 文件

3.1 P148 改错题

3.2 P150 编程题 (3)

3.3 P150 编程题 (7)