

**课程 实 验 报 告**

**课程名称： C语言程序设计课程实验**

**专业班级： 自实1901**

**学 号： U201916457**

**姓 名： 张皓然**

**指导教师：**

**报告日期： 2020年4月13日**

**人工智能与自动化学院**

# 实验1 结构的定义与结构变量的定义有什么区别？结构变量的使用形式，结构指针访问目标变量的使用形式？分别举例说明

## 实验目的

区分结构的定义与结构变量的定义。结构变量的使用形式，结构指针访问目标变量的使用形式

## 设计思路与代码实现

### 设计思路

### 代码实现

1. #include<stdio.h>
2. **struct** Date
3. {
4. **int** month;
5. **int** day;
6. **int** year;
7. };
8. **int** main()
9. {
10. **struct** Date today,\*date\_p;
11. date\_p=&today;
12. date\_p->month=4;
13. date\_p->day=14;
14. date\_p->year=2001;
15. printf("Today is %d/%d/%d\n",date\_p->month,date\_p->day,date\_p->year);
16. **return** 0;
17. }

## 实验过程

（1）编写程序，编译运行（CTRL\_F9）

（2）在用户界面查看运算结果（ALT\_F5）

（3）分析实验结果，总结归纳

## 实验结果

Today is 4/14/2001

## 实验分析

结构的定义并不会引起内存的分配，结构的定义仅仅是定义了一种特定的数据构造类型。结构变量的定义会引起系统按照结构定义时制定的内存模式，为被定义的结构变量分配一定的内存空间。结构变量的使用形式为：结构变量名.成员名。其中’.’是C语言中的一个运算符，它规定的运算是访问结构的成员。注意，访问成员运算’.’是第一运算优先级中的运算，它的结合律是从左向右。结构指针访问目标变量有两种形式，第一种，使用’.’和取内容符\*，形如：(\*结构指针名).成员名。第二种，使用->运算符，形如：结构指针名->成员名。

# 实验2 熟悉结构变量在函数间的传递方法，分别举例说明

## 实验目的

熟悉结构变量在函数间的传递方法。

## 设计思路与代码实现

### 设计思路

### 代码实现

值传递

1. #include<stdio.h>
2. **struct** Date
3. {
4. **int** day;
5. **int** month;
6. **int** year;
7. **int** yearday;
8. **char** mon\_name[4];
9. };
10. **int** day\_of\_year(**struct** Date pd);
11. **int** main()
12. {
13. **struct** Date HireDate;
14. **float** laborage;
16. printf("请输入每天的工资：");
17. scanf("%f",&laborage);
18. printf("请输入年份：");
19. scanf("%d",&HireDate.year);
20. printf("请输入月份：");
21. scanf("%d",&HireDate.month);
22. printf("请输入日：");
23. scanf("%d",&HireDate.day);
24. HireDate.yearday=day\_of\_year(HireDate);
25. printf("从%d年元月1日到%d年%d月%d日的雇佣期限：%d天。\n \
26. 应付给你的工钱：%-.2f元。\n",HireDate.year,HireDate.year,HireDate.month,HireDate.day,\
27. HireDate.yearday,laborage\*HireDate.yearday);
29. **return** 0;
30. }
31. **int** day\_of\_year(**struct** Date pd)
32. {
33. **int** day\_tab[2][13]={{0,31,28,31,30,31,30,31,31,30,31,30,31},{0,31,29,31,30,31,30,31,31,30,31,30,31}};
34. **int** i,day,leap;
35. day=pd.day;
36. leap=(pd.year%4==0&&pd.year%100!=0)||(pd.year%400==0);
37. **for**(i=0;i<pd.month;i++)
38. day+=day\_tab[leap][i];
39. **return** day;
40. }

地址传递

1. #include<stdio.h>
2. #define MAX 3
3. **struct** Address
4. {
5. **char** name[20]="\0";
6. **char** addr[50];
7. **char** tel[15];
8. };
9. **int** input(**struct** Address \*pt);
10. **void** display(**struct** Address\*pt,**int** n);
11. **int** main()
12. {
13. **struct** Address man[MAX];
14. **int** i;
15. **for**(i=0;i<MAX;i++)
16. {
17. **if**(input(&man[i])==0)
18. {
19. **break**;
20. }
21. display(man,i);
22. }
23. }
24. **int** input(**struct** Address\*pt)
25. {
26. printf("Name");
27. gets(pt->name);
28. **if**(pt->name[0]=='\0')
29. **return** 0;
30. printf("address");
31. gets(pt->addr);
32. printf("Telephone");
33. gets(pt->tel);
34. **return** 1;
35. }
36. **void** display(**struct** Address\*pt,**int** n)
37. {
38. **int** i;
40. printf("name\taddress\ttel\n");
41. printf("-------------------------------------\n");
42. **for**(i=0;i<=n;i++,pt++)
43. printf("%s\t%s\t%s\n",pt->name,pt->addr,pt->tel);
44. }

## 测试集

## 实验过程

（1）编写程序，编译运行（CTRL\_F9）

（2）在用户界面查看运算结果（ALT\_F5）

（3）分析实验结果，总结归纳

## 实验结果

## 实验分析

结构变量在函数间的传递有三种方法。第一种，用结构变量的成员作参数，不过这种方法完全没有利用到结构变量的优势，在实际应用中基本不用。第二种，用结构体变量做实际参数。（以上两种算是值传递）第三种，用结构变量的地址或结构数组的首地址作为实际参数。（这是地址传递），传递的实际形式与其他变量相比并无明显差异。

# 实验3 上机调试综合举例中的8.16

## 实验目的

验证调试例8.16

## 设计思路与代码实现

处理N个学生4门课的成绩，它们分别是数学，物理，英语，计算机。按编号从小到大的顺序输入学生的姓名，性别和四门课的成绩。计算每个学生的平均分，并按照从高分到低分的顺序打印平均分高于全班平均成绩的男生的成绩单。

### 代码

1. #include<stdio.h>
2. #include<stdlib.h>
3. #include<math.h>
4. #define N 3
5. **struct** student
6. {
7. **char** name[20];
8. **char** sex;
9. **float** score[4];
10. **float** aver;
11. };
12. **void** input(**struct** student \*p,**int** n);
13. **void** sort(**struct** student \*p,**int** n);
14. **float** average(**struct** student \*p,**int** n);
15. **void** output(**struct** student \*p,**int** n,**float** aver);
16. **int** main()
17. {
18. **struct** student stu[N];
19. **float** t\_aver;
20. input(stu,N);
21. sort(stu,N);
22. t\_aver=average(stu,N);
23. output(stu,N,t\_aver);
24. **return** 0;
25. }
26. **void** input(**struct** student \*p,**int** n)
27. {
28. **int** i,j;
29. **float** per\_aver;
30. **char** str[20];
31. **char**\*subject[4]={"maths","physics","english","computer"};
33. **for**(i=0;i<n;i++,p++)
34. {
35. printf("input student name:\n");
36. gets(p->name);
37. printf("input student sex:\n");
38. gets(str);
39. p->sex=str[0];
40. printf("input student score:\n");
41. **for**(j=0,per\_aver=0;j<4;j++)
42. {
43. puts(subject[j]);
44. gets(str);
45. p->score[j]=atof(str);
46. per\_aver+=p->score[j];
47. }
48. p->aver=per\_aver/4;
49. }
50. }
51. **void** sort(**struct** student\*p,**int** n)
52. {
53. **struct** student temp;
54. **int** i,j;
55. **for**(i=0;i<n-1;i++)
56. {
57. **for**(j=i+1;j<n;j++)
58. {
59. **if**(p[i].aver<p[j].aver)
60. {
61. temp=p[i];
62. p[i]=p[j];
63. p[j]=temp;
64. }
65. }
66. }
67. }
68. **float** average(**struct** student \*p,**int** n)
69. {
70. **int** i;
71. **float** temp;
73. **for**(i=0,temp=0;i<n;i++)
74. temp=temp+p[i].aver;
75. **return** temp/n;
76. }
77. **void** output(**struct** student\*p,**int** n,**float** aver)
78. {
79. **int** i;
80. printf("Name\tSex\t\tmaths\t\tphysics\t\tenglish\t\tcomputer\t\taverage\n");
81. printf("-----------------------------------------------------------------------\n");
82. **for**(i=0;i<n;i++)
83. {
84. **if**( (p[i].aver>aver) && (p[i].sex=='m' || p[i].sex=='M') )
85. {
86. printf("%s\t%c\t%f\t%f\t%f\t%f\t%f\n",p[i].name,p[i].sex,p[i].score[0],p[i].score[1],p[i].score[2],p[i].score[3],p[i].aver );
87. }
88. }
89. }

## 实验过程

（1）编写程序，编译运行（CTRL\_F9）

（2）在用户界面查看运算结果（ALT\_F5）

（3）分析实验结果，总结归纳

## 实验分析

例8.16代码的第84行存在错误（已修正）。该行原本为：

if(p[i].aver>aver&&p[i].sex==’m’||’M’)

考虑括号内各种运算的优先级。>优先级最高，==其次，然后是&&，最后是||。按照书上原本的代码，则先算p[i].aver>aver，得到1或0.再算(p[i].aver>aver)&&(p[i].sex==’m’),最后计算( (p[i].aver>aver)&&(p[i].sex==’m’) ) || (‘M’)，由于’M’是非零的，所以不论前面的结果是不是0，最终这个表达式的值都一定是0.

# 实验4 一个班有N个学生，修5门功课，从键盘输入他们的姓名、学号、性别和成绩。（1）按每个学生的平均成绩从高到低的顺序打印全班的成绩单。（2）求第三门课的平均分（3）找出平均成绩高于全班总平均成绩的女生。要求：输入、输出、计算、排序和查找分别用函数实现，主函数只是调用这些函数

## 实验目的

完成题目

## 设计思路与代码实现

### 设计思路

该题目与例8.16类似，采用类似的算法。

### 代码实现

1. #include<stdio.h>
2. #include<stdlib.h>
3. #include<math.h>
4. #define N 3
5. **struct** student
6. {
7. **char** name[20];
8. **char** number[20];
9. **char** sex;
10. **float** score[5];
11. **float** aver;
12. };
13. **void** inputs(**struct** student stu[],**int** n);
14. **void** outputs(**struct** student \*p[]);
15. **void** sortaver(**struct** student stu[],**int** n,**struct** student \*sort\_aver[]);
16. **float** aver3(**struct** student stu[],**int** n);
17. **void** f\_high\_aver(**struct** student stu[],**int** n,**struct** student \*sort\_female[]);
18. **int** main()
19. {
20. **struct** student stu[N];
21. **struct** student \*sort\_aver[N+1]={NULL},\*sort\_female[N+1]={NULL};;
22. **float** aver\_sub3;
23. inputs(stu,N);
25. printf("按照平均成绩排序：\n");
26. sortaver(stu,N,sort\_aver);
27. outputs(sort\_aver);
29. printf("第三门课的平均分\n");
30. aver\_sub3=aver3(stu,N);
31. printf("%f\n",aver\_sub3);
33. printf("平均分较高的女同学：\n");
34. f\_high\_aver(stu,N,sort\_female);
35. outputs(sort\_female);
37. **return** 0;
38. }
40. **void** inputs(**struct** student stu[],**int** n)
41. {
42. **int** i,j;
43. **char** str[20];
44. **for**(i=0;i<n;i++)
45. {
46. printf("姓名?\n");
47. gets(stu[i].name);
49. printf("学号?\n");
50. gets(stu[i].number);
52. printf("性别?\n");
53. gets(str);
54. stu[i].sex=str[0];
56. **for**(j=0;j<5;j++)
57. {
58. printf("第%d门课程成绩?\n",j+1);
59. gets(str);
60. stu[i].score[j]=atof(str);
61. }
62. }
63. }
65. **void** outputs(**struct** student \*p[])
66. {
67. **int** i;
68. printf("姓名\t学号\t性别\t\t课程1\t\t课程2\t\t课程3\t\t课程4\t\t课程5\n");
69. printf("----------------------------------------------------------------------------------------\n");
71. **for**(i=0;p[i]!=NULL;i++)
72. {
73. printf("%s\t%s\t%c\t%f\t%f\t%f\t%f\t%f\n",p[i]->name,p[i]->number,p[i]->sex,\
74. p[i]->score[0],p[i]->score[1],p[i]->score[2],p[i]->score[3],p[i]->score[4]);
75. }
76. }
78. **void** sortaver(**struct** student stu[],**int** n,**struct** student \*sort\_aver[])
79. {
80. **int** i,j;
81. **struct** student \*temp;
82. //求平均数
83. **for**(i=0;i<n;i++)
84. stu[i].aver=(stu[i].score[0]+stu[i].score[1]+stu[i].score[2]+stu[i].score[3]+stu[i].score[4])/5;
86. **for**(i=0;i<n;i++)
87. sort\_aver[i]=&stu[i];
89. //给平均数排序
90. **for**(i=0;i<n-1;i++)
91. {
92. **for**(j=i+1;j<n;j++)
93. {
94. **if**( sort\_aver[i]->aver < sort\_aver[j]->aver )
95. {
96. temp=sort\_aver[i];
97. sort\_aver[i]=sort\_aver[j];
98. sort\_aver[j]=temp;
99. }
100. }
101. }
102. }
104. **float** aver3(**struct** student stu[],**int** n)
105. {
106. **float** aver=0;
107. **int** i;
108. **for**(i=0;i<n;i++)
109. aver+=stu[i].score[2];
110. aver/=n;
111. **return** aver;
112. }
113. **void** f\_high\_aver(**struct** student stu[],**int** n,**struct** student \*sort\_female[])
114. {
115. **int** i,j;
116. **float** allaver=0;
117. //求全班总平均成绩
118. **for**(i=0;i<n;i++)
119. allaver+=stu[i].aver;
120. allaver/=n;
121. //筛选
122. **for**(i=0,j=0;i<n;i++)
123. {
124. **if**( (stu[i].sex=='f') && (stu[i].aver>allaver) )
125. sort\_female[j++]=&stu[i];
126. }
127. }

## 测试集

zhr u201916457 m 80 80 80 80 80

wzc u201916458 f 90 90 90 90 90

zll u201916459 f 10 10 10 10 10

## 实验过程

（1）编写程序，编译运行（CTRL\_F9）

（2）在用户界面查看运算结果（ALT\_F5）

（3）分析实验结果，总结归纳

## 实验结果

直接将结果复制进word会失去原有的排版，所以以截图的形式呈现，如图1。

|  |
| --- |
|  |
| 图1. 实验结果图 |

# 实验5 第八章习题（编程题）4、5、10

## 题目4

### 代码实现

1. #include<stdio.h>
2. #include<stdlib.h>
3. #include<math.h>
4. #include<time.h>
5. #define N 10
6. **struct** student
7. {
8. **char** name[20];
9. **char** number[20];
10. **float** score[4];
11. **float** aver;
12. **int** ranks;
13. };
14. **void** inputs(**struct** student stu[], **int** n);
15. **void** outputs(**struct** student\* p[], **int** n);
16. **void** sortaver(**struct** student stu[], **int** n, **struct** student\* sort\_aver[]);

19. **int** main()
20. {
21. **struct** student stu[N];
22. **struct** student\* sort\_aver[N] = { NULL };
23. **char** clas[20];
25. printf("班号？\n");
26. gets(clas);
28. inputs(stu, N);
29. sortaver(stu, N, sort\_aver);
30. **time\_t** timep;
31. time(&timep);
32. printf("%s\n", ctime(&timep));
33. puts(clas);
34. outputs(sort\_aver, N);
35. **return** 0;
36. }
38. **void** inputs(**struct** student stu[], **int** n)
39. {
40. **int** i, j;
41. **char** str[20];
43. **for** (i = 0; i < n; i++)
44. {
45. printf("姓名?\n");
46. gets(stu[i].name);
48. printf("学号?\n");
49. gets(stu[i].number);
51. **for** (j = 0; j < 4; j++)
52. {
53. printf("第%d门课程成绩?\n", j + 1);
54. gets(str);
55. stu[i].score[j] = atof(str);
56. }
57. }
58. }
60. **void** outputs(**struct** student\* p[], **int** n)
61. {
62. **int** i;

65. printf("姓名\t学号\t课程1\t课程2\t课程3\t课程4\t名次\n");
66. printf("----------------------------------------------------------------------------------------\n");
68. **for** (i = 0; i < n; i++)
69. {
70. printf("%s\t%s\t%.2f\t%.2f\t%.2f\t%.2f\t%d\n", p[i]->name, p[i]->number, \
71. p[i]->score[0], p[i]->score[1], p[i]->score[2], p[i]->score[3], p[i]->ranks);
72. }
73. }
75. **void** sortaver(**struct** student stu[], **int** n, **struct** student\* sort\_aver[])
76. {
77. **int** i, j;
78. **struct** student\* temp;
79. //求平均数
80. **for** (i = 0; i < n; i++)
81. stu[i].aver = (stu[i].score[0] + stu[i].score[1] + stu[i].score[2] + stu[i].score[3]) / 4;
83. **for** (i = 0; i < n; i++)
84. sort\_aver[i] = &stu[i];
85. **for** (i = 0; i < n; i++)
86. stu[i].ranks = 1;

89. **for** (i = 0; i < n ; i++)
90. {
91. **for** (j = i + 1; j < n; j++)
92. {
93. **if** (sort\_aver[i]->aver < sort\_aver[j]->aver)
94. {
95. temp = sort\_aver[i];
96. sort\_aver[i] = sort\_aver[j];
97. sort\_aver[j] = temp;
98. }
99. }
101. **if** (i > 0)
102. **if** (sort\_aver[i]->aver != sort\_aver[i - 1]->aver)
103. sort\_aver[i]->ranks += sort\_aver[i-1]->ranks;
104. }
105. }

### 实验过程

（1）编写程序，编译运行（CTRL\_F9）

（2）在用户界面查看运算结果（ALT\_F5）

（3）分析实验结果，总结归纳

### 实验结果

经验证，实验结果与预期相符。

## 题目5

### 代码实现

1. #include<stdio.h>
2. #include<stdlib.h>
3. #include<math.h>
4. #include<time.h>
5. #define N 10
6. **struct** student
7. {
8. **char** name[20];
9. **char** number[20];
10. **float** score[4];
11. **float** aver;
12. **int** ranks;
13. };
14. **void** inputs(**struct** student stu[], **int** n);
15. **void** outputs(**struct** student\* p[], **int** n);
16. **void** sortaver(**struct** student stu[], **int** n, **struct** student\* sort\_aver[]);
17. **void** findmax(**struct** student stu[],**int** n,**struct** student\* maxs);
18. **void** find\_loser(**struct** student stu[],**int** n,**struct** student\* loser[]);
19. **int** main()
20. {
21. **struct** student stu[N];
22. **struct** student\* sort\_aver[N] = { NULL },\*maxs,\*loser[N+1]={NULL};;
23. **char** clas[20];
25. printf("班号？\n");
26. gets(clas);
28. inputs(stu, N);
29. sortaver(stu, N, sort\_aver);
30. **time\_t** timep;
31. time(&timep);
32. printf("%s\n", ctime(&timep));
33. puts(clas);
34. outputs(sort\_aver, N);
35. findmax(stu,N,maxs);
36. find\_loser(stu,N,loser);
37. **return** 0;
38. }
40. **void** inputs(**struct** student stu[], **int** n)
41. {
42. **int** i, j;
43. **char** str[20];
45. **for** (i = 0; i < n; i++)
46. {
47. printf("姓名?\n");
48. gets(stu[i].name);
50. printf("学号?\n");
51. gets(stu[i].number);
53. **for** (j = 0; j < 4; j++)
54. {
55. printf("第%d门课程成绩?\n", j + 1);
56. gets(str);
57. stu[i].score[j] = atof(str);
58. }
59. }
60. }
62. **void** outputs(**struct** student\* p[], **int** n)
63. {
64. **int** i;

67. printf("姓名\t学号\t平均成绩\t名次\n");
68. printf("----------------------------------------------------------------------------------------\n");
70. **for** (i = 0; i < n && p[i]!=NULL; i++)
71. {
72. printf("%s\t%s\t%.2f\t%d\n", p[i]->name, p[i]->number, \
73. p[i]->aver, p[i]->ranks);
74. }
75. }
77. **void** sortaver(**struct** student stu[], **int** n, **struct** student\* sort\_aver[])
78. {
79. **int** i, j;
80. **struct** student\* temp;
81. //求平均数
82. **for** (i = 0; i < n; i++)
83. stu[i].aver = (stu[i].score[0] + stu[i].score[1] + stu[i].score[2] + stu[i].score[3]) / 4;
85. **for** (i = 0; i < n; i++)
86. sort\_aver[i] = &stu[i];
87. **for** (i = 0; i < n; i++)
88. stu[i].ranks = 1;

91. **for** (i = 0; i < n ; i++)
92. {
93. **for** (j = i + 1; j < n; j++)
94. {
95. **if** (sort\_aver[i]->aver < sort\_aver[j]->aver)
96. {
97. temp = sort\_aver[i];
98. sort\_aver[i] = sort\_aver[j];
99. sort\_aver[j] = temp;
100. }
101. }
103. **if** (i > 0)
104. **if** (sort\_aver[i]->aver != sort\_aver[i - 1]->aver)
105. sort\_aver[i]->ranks += sort\_aver[i-1]->ranks;
106. }
107. }
108. **void** findmax(**struct** student stu[],**int** n,**struct** student\* maxs)
109. {
110. **int** i;
111. maxs=stu;
112. **for**(i=0;i<n;i++)
113. **if**(maxs->aver < stu[i].aver)
114. maxs=&stu[i];
115. outputs(&maxs,1);
116. }
117. **void** find\_loser(**struct** student stu[],**int** n,**struct** student\* loser[])
118. {
119. **int** i,j=0;
120. **for**(i=0;i<n;i++)
121. {
122. **if**(stu[i].aver<60)
123. loser[j++]=&stu[i];
124. }
125. outputs(loser,10);
126. }

### 实验过程

（1）编写程序，编译运行（CTRL\_F9）

（2）在用户界面查看运算结果（ALT\_F5）

（3）分析实验结果，总结归纳

### 实验结果

经验证，实验结果与预期相符。

## 题目10

公司若干员工，每个员工有姓名，性别，工龄，工资等信息。编程输入员工档案信息和便于工资发放的各种钞票数（工资为整数，发放的工资各种钞票限定为：100元，50元，20元，10元，5元，1元，发放钞票张数要求最少）,要求输出工龄大于20年，工资高于5000元的所有男员工信息和工资发放的各种钞票数。

### 代码实现

1. #include<stdio.h>
2. #include<stdlib.h>
3. #define N 2
4. **struct** staff
5. {
6. **char** name[20];
7. **char** sex;
8. **int** seniority;
9. **int** salary;
10. **int** bill[6];
11. };
12. **void** inputs(**struct** staff\*,**int**);
13. **void** payoff(**struct** staff\*,**int**);
14. **void** condition\_ouput(**struct** staff\*,**int**);
15. **int** main()
16. {
17. **struct** staff stf[N];
18. inputs(stf,N);
19. payoff(stf,N);
20. condition\_ouput(stf,N);
21. **return** 0;
22. }
23. **void** inputs(**struct** staff \*stf,**int** n)
24. {
25. **int** i;
26. **char** str[20];
27. **for**(i=0;i<n;i++)
28. {
29. printf("员工姓名：\n");
30. gets(stf[i].name);
31. printf("员工性别：\n");
32. gets(str);
33. stf[i].sex=str[0];
34. printf("工龄：\n");
35. gets(str);
36. stf[i].seniority=atoi(str);
37. printf("工资：\n");
38. gets(str);
39. stf[i].salary=atoi(str);
40. }
41. }
42. **void** payoff(**struct** staff\* stf,**int** n)
43. {
44. **int** i,j;
45. **int** temp;
46. **int** money[6]={100,50,20,10,5,1};
47. **for**(i=0;i<n;i++)
48. {
49. temp=stf[i].salary;
50. **for**(j=0;temp!=0;j++)
51. {
52. stf[i].bill[j]=temp / money[j];
53. temp-=stf[i].bill[j]\*money[j];
54. }
55. }
56. }
57. **void** condition\_ouput(**struct** staff\* stf,**int** n)
58. {
59. **int** i,j=0;
60. **struct** staff \*p[N]={NULL};
61. **for**(i=0;i<n;i++)
62. **if**( (stf[i].seniority>20) && (stf[i].salary>5000) && (stf[i].sex=='M') )
63. p[j++]=&stf[i];
64. **for**(i=0;i<n && p[i]!=NULL ;i++ )
65. {
66. printf("姓名：%s\t",p[i]->name);
67. printf("性别：%c\t",p[i]->sex);
68. printf("工龄：%d\t",p[i]->seniority);
69. printf("工资：%d\n",p[i]->salary);
70. printf("100元%d张\t50元%d张\t20元%d张\t10元%d张\t5元%d张\t1元%d张\n",\
71. p[i]->bill[0],p[i]->bill[1],p[i]->bill[2],p[i]->bill[3],p[i]->bill[4],p[i]->bill[5]);
72. }
73. }

### 实验过程

（1）编写程序，编译运行（CTRL\_F9）

（2）在用户界面查看运算结果（ALT\_F5）

（3）分析实验结果，总结归纳

### 实验结果

经验证，实验结果与预期相符。

附录

## 开发环境

本次实验中使用的环境配置如下：

（1）操作系统版本：Arch Linux x86\_64

（2）编译器及其版本：GCC version 8.1.1

（3）自动编译工具：CMake version 3.11.4

（4）编程环境：Borland C++ 3.1