

高级程序设计（C 语言）实验指导书

使用说明.....	3
实验一 C 语言程序初步.....	4
实验二 数据类型.....	6
实验三 运行符与表达式.....	7
实验四 输入与输出.....	8
实验五 选择结构.....	10
实验六 条件型循环结构.....	12
实验七 计数型循环结构.....	14
实验八 函数的定义及调用.....	17
实验九 数组.....	18
实验十 指针（一）.....	21
实验十一 指针（二）.....	23
实验十二 结构体和共用体.....	26
实验十四 文件.....	28
实验十六 综合应用举例（二）.....	35
附录一： Turbo C 程序设计上机指导.....	43
Turbo C 程序设计初步.....	43
Turbo C 常用的编辑命令.....	46
Turbo C 程序的调试.....	48
Turbo C 编译、连接和运行时的常见错误.....	50
附录二： C 语言程序设计调试技术.....	50
C 语言程序设计调试技术（一）—— 运行错误的判断与调试.....	50
C 语言程序设计调试技术（二）—— 基本调试手段.....	50
C 语言调试技术（三） -Turbo C 集成环境的调试功能.....	50
C 语言调试技术（四） --图形程序运行的条件.....	50

实验一 C 语言程序初步

一、目的和要求

- 1、熟悉 C 语言运行环境。
- 2、掌握 C 语言程序的书写格式和 C 语言程序的结构。
- 3、掌握 C 语言上机步骤，了解运行一个 C 程序的方法。

二、实验内容

1、C 语言上机步骤：

A 、进入系统

然后进行编辑源程序->编译->连接->执行程序->显示结果

B 、常用命令

编辑切换 (F6)，编译 (F9)，运行 (CTRL+F9)，显示结果 (ALT+F5) 其它常用命令见“附录一”。

2、有下面的 C 程序，目的是想计算由键盘输入的任何两个整数的积。

```
/* * * * * * * * ex1.c * * * * * * * /
```

```
#include <stdio.h>;
main()
{
scanf(“%x,%y”,&x,%y)
p=product(x,t)
printf(“The product is :”,p)
}
int product(int a ,int b )
{
int c
c=a*b
return c
}
```

请调试上述程序。

三、实验步骤

- 1、静态地检查上述程序，改正程序中的错误。
- 2、在编辑状态下照原样键入上述程序。
- 3、编译并运行上述程序，记下所给出的出错信息。
- 4、按照事先静态检查后所改正的情况，进行纠错。
- 5、再编译执行纠错后的程序。如还有错误，再编辑改正，直到不出现语法错误为止。

6、下面给出 6 组测试用例，你认为哪几组较好？为什么？

- (1) 0, 0
- (2) 0, 9 9
- (3) 2 0, 5 0
- (4) 3 3 0 0 0, 2 0
- (5) - 5, - 2
- (6) - 5, 2

要测试出上述程序中所有错误，你认为应当用几组测试用例？

四、分析与讨论

- 1、记下在调试过程中所发现的错误、系统给出的出错信息和对策。分析讨论对策成功或失败的原因。
- 2、总结 C 程序的结构和书写规则。

实验二 数据类型

一、目的和要求

- 1、了解C语言中数据类型的意义。

二、实验内容和步骤

- 1、下面的程序试图计算由键盘输入的任何两个整数的平均值：

```
#include <stdio.h>
main()
{
    int x,y,a;
    scanf("%x%y",&x,&y);
    a=(x+y)/2;
    printf("The average is :”a);
}
```

调试无语法错误后，分别使用下列测试用例对上述程序进行测试：

- (1) 2, 6
- (2) 1, 3
- (3) -2, -6
- (4) -1, -3
- (5) -2, 6
- (6) -1, 3
- (7) 1, 0
- (8) 1, 6
- (9) 3 2 8 0 0, 3 3 0 0 0
- (10) -3 2 8 0 0, 3 3 0 0 0

1、分析上述哪几组测试用例较好？通过测试，你发现程序有什么错误了吗？若有错误，请指出错误原因。

2、操作符sizeof用以测试一个数据或类型所占用的存储空间的字节数。请编写一个程序，测试各基本数据类型所占用的存储空间大小。

三、分析与讨论

如何正确地选用数据类型？（提示：给合前面做过的两个实验及书本进行讨论总结）

实验三 运行符与表达式

一、目的和要求

- 1、理解常用运行符的意义。
- 2、掌握C语言表达式的运行规则。

二、实验内容和步骤

- 1、编写一个C语言程序，测试下列各表达式

`i, j`

`i + 1, j + 1`

`i ++, j ++`

`++ i, ++ j`

`i + ++ + j`

`++ i + ++ + j ++`

要求在各表达式中 `i` 和 `j` 都分别有相同的初值。在实验中注意下列问题：

- (1) 哪些表达式是错误的？为什么？
- (2) 理解 `+`, `++`, `++ i`, `i ++` 的意义和优先级别。

- 2、编写一个程序，测试常用的十个运算符的优先顺序。

四、分析与讨论

分析总结运算符的优先级。

实验四 输入与输出

一、目的和要求

- 1、掌握 C 语言程序输入、输出的方法和格式。

二、实验内容与步骤

- 1、输入并编辑下面的程序

```
main()
{
    int a,b;
    float c,d;
    long e,f;
    unsigned int u,v;
    char c1,c2;
    scanf("%d,%d",&a,&b);
    scanf("%f,%f",&c,&d);
    scanf("%ld,%ld",&e,&f);
    scanf("%o,%o",&u,&v);
    scanf("%c,%c",&c1,&c2);
    printf("\n");
    printf("a=%7d,b=%7d\n",&a,&b);
    printf("c=%10.2f,d=%10.2f\n",&c,&d);
    printf("e=%17ld,f=%17ld\n",&e,&f);
    printf("u=%o,d=%o\n",&u,&v);
    printf("c1=%c,d=%c\n",&c1,&c2);
}
```

这个程序有语法错误吗？为什么？

- 2、调试上述程序无语法错误后，用下面的测试数据，对程序进行测试：

a=123,b=456,c=17.6,d=71837.65,e=70000,f=2174506,u=62000,v=58765,c1='a',c2='b'

分析运行结果。特别注意输入 c1,c2 的值是什么？什么原因？

- 3、将输入 e 和 f 的语句改为：

```
scanf("%d,%d",&e,&f);
```

再用上述测试数据测试并分析结果。

- 4、将输入 u、v 的语句改为：

```
scanf("%d,%d",&u,&v);
```

再用上述测试数据测试并分析结果。

- 5、将输出 e，f 的语句改为：

```
printf("e=%17d,f=%17d\n",e,f);
```

再用上述测试数据测试并分析结果。

- 6、将输出 u、v 的语句改为：

```
printf("u=%u,v=%u\n",u,v);
```

或

```
printf("u=%d,v=%d\n",u,v);
```

再用上述测试数据测试并分析结果。

7、请读者自己修改程序和改变数据输入的形式，分析各种情况下的输入与输出。

8、在 `scanf("%c,%c",&c1,&c2)`；语句之前加一个语句：

```
getchar();
```

9、验证转义字符 `\n` 与 `\r` 的意义有何不同。

三、分析与讨论

1、总结在 `printf` 函数中可以使用的各种格式指定符，并给出样例。

2、总结在 `printf` 函数中可以使用的各转义字符及其功能。

实验五 选择结构

一、实验目的

- 1、了解条件与程序流程的关系
- 2、了解用不同的数据使程序的流程覆盖不同的语句、分支和路径。

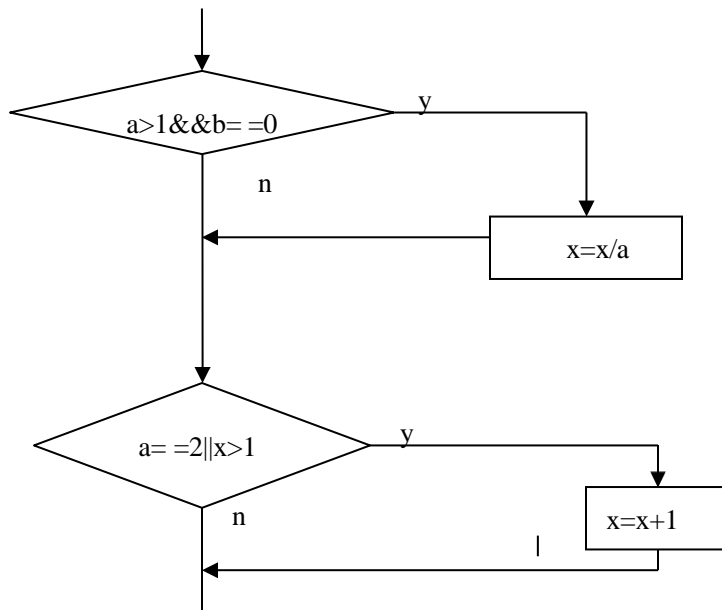
二、实验内容和步骤

1、题目

有如下程序段：

```
{.....  
    if (a>1&& b==0) x=x/a;  
    if(a==2||x>1) x=x+1;  
}
```

为了更容易明白程序的逻辑结构，我们用图 5.1 所示流程图来加以描述。



要求增加一些输入语句和输出语句，以便使上述程序能在不同的 a, b 和 x 值下运行，并且能观察程序流程经过（覆盖）了哪些语句、哪些分支及哪些路径。

2、实验步骤

记下分别使用下列各组数据运行时的操作流程。

- (1) $a=1, b=1, x=1$;
- (2) $a=1, b=1, x=2$;
- (3) $a=3, b=0, x=1$;
- (4) $a=2, b=1, x=4$;
- (5) $a=2, b=1, x=1$;
- (6) $a=1, b=0, x=2$;
- (7) $a=2, b=1, x=1$;
- (8) $a=3, b=0, x=2$ 。

三、分析讨论

- (1) 用哪一组数据就可使程序中的每个处理语句都执行一次？为了找出程序中各条处理语句中的错误，应该使用什么样的数据对程序进行测试？请上机验证自己的结论。
- (2) 用哪两组数据就可以使程序段中的每个分支都运行一次？这种情形与上面的讨论有何不同？如果为了找出程序中积压分支中的错误，应该使用什么样的数据对程序进行测试？请上机验证自己的结论。
- (3) 用哪两组数据就可以把判定框中的每个条件运算都进行一次？如果为了测试出判定条件中的错误，应使用哪些数据对程序进行测试？请上机验证自己的结论。
- (4) 用哪四组数据才可以把各种条件的组合都检测一遍？如果为了测试各种条件的组合的情形，应该使用什么样的测试数据？请上机验证自己的结论。
- (5) 用哪四组数据才可以把起始到终止的各条路径都覆盖一次？如果为了测试出程序在不同路径下的错误，应该使用什么样的测试数据？请上机验证自己的结论。

四、进一步的实验

1. 题目

从键盘上输入三个数，让它们代表三条线段的长度，请写一个判断这三条线段所组成的三角形属于什么类型（不等边，等腰，等边或不构成三角形）的 C 程序。

2. 请分别设计下列数据对自己的程序进行测试：

- (1) 找出各条语句中的错误。
- (2) 找出积压分支中的错误。
- (3) 找出各条件中的错误。
- (4) 找出各种条件组合中的错误。
- (5) 找出各条路径中的错误。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/196100231030010044>