

江苏财经职业技术学院

《软件技术基础》 实验指导书

胡永祥 编

专业 _____

班级 _____

学号 _____

姓名 _____

2011 年 1 月

目 录

《软件技术基础》上机实验说明.....	1
实验一 C语言程序初步.....	2
实验二 数据类型、运行符与表达式.....	6
实验三 输入与输出.....	9
实验四 选择结构.....	12
实验五 循环结构.....	15
实验六 数组.....	20
实验七 函数的定义及调用.....	24
实验八 指针（一）.....	26
实验九 指针（二）.....	30
实验十 结构体和共用体.....	34
实验十一 线性表及其应用.....	36
实验十二 栈和队列.....	39
实验十三 树及二叉树.....	43
实验十四 查找.....	45
实验十五 排序.....	48
附录一 Turbo C 程序设计上机指导.....	51
附录二 C语言程序设计调试技术.....	62
附录三 参考文献.....	69

《软件技术基础》上机实验说明

本课程是我院计算机专业的一门综合性的专业基础课。着重培养学生在软件设计领域的基本素质，基本方法和设计理念。通过本课程的学习，掌握 C 语言和数据结构的基本理论，为学生今后在软件设计领域的学习打下坚实的基础。通过本课程的学习和上机实践，使学生能够编制实用的 C 语言应用程序，培养出扎实的软件开发基本技能，编制高效可靠的程序，并养成良好的编程风格，为进一步学习后续课程和将来从事应用软件开发奠定良好的基础。

上机实验是学习程序设计必不可少的实践环节，特别是 C 语言灵活、简洁，更需要通过编程的实践来真正掌握它。对于程序设计语言的学习目的，可以概括为学习语法规则、掌握程序设计方法、提高程序开发能力，这些都必须通过充分的实际上机操作才能完成。

学习程序设计除了课堂讲授以外，必须保证有不少于课堂讲授学时的上机时间。因为学时所限，课程不能安排过多的统一上机实验，所以希望学生有效地利用课程上机实验的机会，尽快掌握用 C 语言开发程序的能力，为今后的继续学习打下一个良好的基础。为此，我们结合课堂讲授的内容和进度，安排了 15 次上机实验。课程上机实验的目的，不仅仅是验证教材和讲课的内容、检查自己所编的程序是否正确，更要通过上机实验，培养软件开发的基本技能，并养成良好的编程风格。课程安排的上机实验的目的可以概括为如下几个方面：

- 加深对课堂讲授内容的理解
- 熟悉程序开发环境、学习计算机系统的操作方法
- 学习上机调试程序
- 培养软件开发的基本技能
- 养成良好的编程风格

实验要求

上机实验一般经历上机前的准备（编程）、上机调试运行和实验后的总结三个步骤。实验结束后，要整理实验结果并认真分析和总结，根据教师要求写出实验报告。

实验报告一般包括如下内容：

实验内容、算法说明、程序清单、运行结果、分析与思考

实验一 C语言程序初步

实验时间_____

实验地点_____

实验成绩_____

一、实验目的

1. 了解TC环境的组成
2. 学习TC环境的使用方法
3. 掌握C语言程序从编辑、编译、连接到运行并得到运行结果的过程

二、实验内容

1. 了解TC环境的组成

开机后进入Windowd系统的“资源管理器”，找到TC环境所在的文件夹。查看在TC目录下是否包括了TC.EXE、TCC.EXE、TLINK.EXE、TCCONFIG.TC等文件；查看INCLUDE、LIB两个子目录下的文件，了解这些文件的作用。

2. 进入、退出和定制TC环境

运行TC目录下的TC.EXE文件，就可进入TC环境。

退出TC环境，可在“FILE”菜单下选择"QIUT"菜单项，或用热键<Alt+X>，计算机返回到操作系统的控制下。

在程序开发的过程中，有时需要返回到操作系统界面下观察程序的运行情况，但是又要保留运行的TC环境，操作方法是：选择“File”菜单下的“OS shell”菜单项返回DOS操作系统界面，用“EXIT”命令可重新进入TC环境；用<Alt>+<空格>键返回到Windows界面，TC环境缩小为屏幕下方任务条上的一图标，用鼠标单击该图标重新进入TC环境。

注意此时是返到操作系统，TC环境没有真正退出，如果此时再次运行TC.EXE文件，刚才是返到Windows环境时，会重新打开一个TC窗口，返到DOS环境时会给出一个错误提示：“Program too big to fit in memory”。这是因为DOS操作系统只管理640K的内存，放不下两个TC环境。Windows操作系统把TC环境作为一个窗口进行管理，因此Windows窗口的一些属性也是有效的。从TC环境返回Windows系统时会出现一个快捷菜单（在任务条的TC图标上单击鼠标右键也可以出现快捷菜单），选择“属性”菜单项，屏幕出现一个名为“TC属性”的窗口，在该窗口里选择“屏幕”一页，在“用法”一项里选中“屏

幕”单选钮，单击“应用”按钮，关闭“TC属性”的窗口，TC环境以Windows窗口形式出现。在窗口上方出现工具条，可以对窗口进行定制，注意“全屏幕”和“中文”两个工具钮的作用。

TC环境下的“Options”菜单下可对TC环境进行设置，初学者要了解“Directories”的作用，一般不要改变系统的其它设置。关于“Directories”各项的作用和设置参阅本指导书附录一中关于TC环境的介绍，学会改变输出文件的目录。

3. 运行演示程序

在TC目录下有一个名为“BGIDEMO.C”的源程序，这本是为TC图形函数提供的演示程序，运行这个演示程序。在“File”菜单下选“Load”项，在屏幕出现的“Load File Name”窗口里输入“BGIDEMO.C”，该程序被装入编辑窗口，按<Alt+R>键，程序被编译、连接并运行。注意运行此程序需要图形库文件，一般被装在TC目录下，没有这个图形库文件就不能运行图形演示程序。

4. 编写自己的第一个程序

按键盘<Alt+E>键，激活编辑窗口，录入如下C源程序：

```
main()  
{  
    printf("This is a C Program.\n")  
    printf("I am a student.\n");  
}
```

按<Alt+R>键，编译、连接、运行程序。屏幕出现错误提示：

通过提示，可以知道上面程序第二行的最后漏敲一个分号，改正后程序运行。按<Alt>+<F5>键观察输出结果。

按<F2>键，程序存入硬盘，文件名自定（如：test1）。通过资源管理器观察当前目录下名为test1的几个文件，它们的扩展名分别是什么。

改变“Options”菜单下“Directories”项下的输出文件目录，用<F2>键把程序再存一次并运行。在新设定的输出目录下观察名为test1的文件的存储情况。

用“File”菜单下的“Write to”项，把文件存在新设定的输出目录处，文件名仍为test1。

录入如下程序：

```
main()
{
    printf("This is another C Program.\n");
}
```

按<F2>键并用test1文件名保存，然后运行这个程序。

按<Alt>+<F3>键，屏幕出现刚才操作过的几个文件的名称，将刚才设定目录下的test1.c装入编辑器后运行它，我们观察到输出的仍是后来键入的程序的内容。这是因为VC编译系统在接收“RUN”命令后，对test1.c、test1.obj、test1.exe三个文件的建立时间进行比较，如果扩展名为.c的文件建立时间晚于.exe文件，它就认为源文件进行了修改，所以对源文件重新进行编译连接，如果.exe文件的时间晚于源文件的建立时间，就直接运行这个文件，不再重新编译连接。因为我们两次输入的程序名称都是test1，新设定目录中存放的是第一个程序的test1.c和第二个程序的test1.exe，就出现了现在的情况。

5. 分别编译、连接、运行程序

“Run”命令是将编译、连接、运行一次完成，实际完成了三件工作，下面分别进行编译、连接和运行。

用<ALT+C>命令打开“Compile”菜单，并选择“Compile to object”命令编译该源程序文件，然后选择“Compile”菜单的“Link EXE file”命令调用连接程序连接成可执行文件，最后用“Run”菜单的“Run”命令运行程序，用“Run”菜单的“User screen”命令查看运行结果。由于编译、连接、运行是分别进行的，所以编译系统不再对相关三个文件的建立时间进行比较，我们看到的就是编辑器里当前的程序输出结果。

6. 编写程序，实现求整数10、20和35的平均值。

三、实验要求

1. 学习TC的基本操作，编写程序。
2. 运行程序并记录运行结果。
3. 将源程序、目标文件、可执行文件和实验报告存在U盘上。

四、分析讨论

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、运行符与表达式

实验时间_____

实验地点_____

实验成绩_____

一、目的和要求

- 1、了解C语言中数据类型的意义
- 2、理解常用运行符的意义
- 3、掌握C语言表达式的运行规则

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( "%d, %y" , &x, &y);          /*格式字符串有错*/
```

```
    a=(x+y)/2;
```

```
    printf( "The average is : " a);      /*格式字符串有错*/
```

```
}
```

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

(1 0) - 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、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。

三、分析与讨论

1、如何正确地选用数据类型？（提示：给合前面做过的实验及书本进行讨论总结）

修改后的程序如下：

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

```
void main(void)
```

```
{
```

```

int x, y, a;
scanf( "%x%d" , &x, &y);
a = (x+y) /2;
printf( "The average %d and %d is : %d\n" , x, y, a);
printf( "The length of a is %d.\n" , sizeof(a)) ;
printf( "The length of x is %d.\n" , sizeof(x));
printf( "The length of y is %d.\n" , sizeof(y));

```

2、分析总结运算符的优先级。

可能有如下程序：

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

```
void main(void)
```

```

int i, j;
i = 1;
j = 101;
printf("i = %d, and j = %d\n", i, j);
printf("i++ = %d, and j++ = %d\n", i++, j++);
printf("++i = %d, and ++j = %d\n", ++i, ++j);
printf("i+1 = %d, and j+1 = %d\n", i+1, j+1);
printf("i+++j = %d\n", i+++j);
printf("++i+++j++ = %d\n", (++i)+++j++);
printf("i = %d, and j = %d\n", i, j);    /*注意i, j的值，分析i+++j 的执行过程*/
printf("i-- = %d, and j-- = %d\n", i--, j--);
printf("--i = %d, and --j = %d\n", --i, --j);
printf("i-1 = %d, and j-1 = %d\n", i-1, j-1);
printf("i---j = %d\n", i---j);
printf("--i---j--- = %d\n", (--i)---(j--));

```

实验三 输入与输出

实验时间_____

实验地点_____

实验成绩_____

一、目的和要求

- 1、掌握C语言程序格式输入/输出函数的使用
- 2、掌握字符的输入/输出函数的使用

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。

可能的修改程序如下：

```
#include <stdio.h>

void main(void)
{
    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);          /*注意八进制表示输入时要用0打头*/
    scanf("%c,%c", &c1, &c2);      /*当在此处输入字符变量时，会是什么效果？*/
    printf("-----*-----*-----*-----*");
    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, v=%o\n", u, v);
    printf("c1=%c, c2=%c\n", c1, c2);
    printf("-----*-----*-----*-----*\n");
}
```

实验四 选择结构

实验时间_____

实验地点_____

实验成绩_____

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、有如下程序片段：

```
{ .....  
    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4.1所示流程图来加以描述。

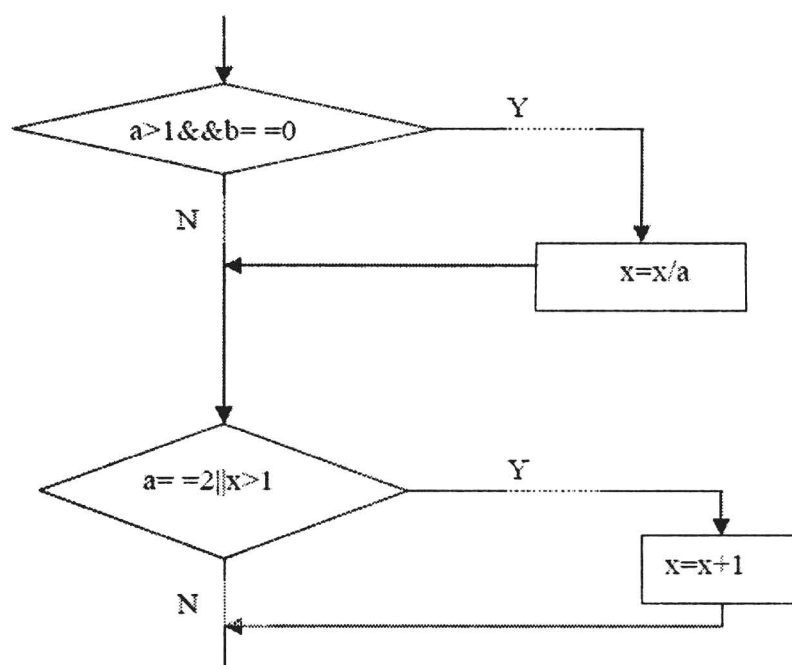


图4.1 分支选择流程

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) $a=3, b=0, x=2$ 。

3、程序清单

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。

实验五 循环结构

实验时间_____

实验地点_____

实验成绩_____

一、目的和要求

1、掌握在程序设计条件型循环结构时，如何正确地设定循环条件，以及如何控制循环的次数

2、了解条件型循环结构的基本测试方法

3、掌握如何正确地控制计数型循环结构的次数

4、了解对计数型循环结构进行测试的基本方法

5、了解在循环嵌套结构中，提高程序效率的方法

二、实验内容与步骤

1、下面是一个计算e的近似值（使误差小于给定的 δ (假定为 10^{-6})）的程序，修改并运行之。

```
/*缺？？*/  
main( )  
{  
    double e=1.0,x=1.0,y,detax;  
    int i=1;  
    printf("\n please input enter a small number:");  
    scanf("%lf",&detax);  
    y=1/x;  
    while(y>=detax)  
    {  
        x=x*I;  
        y=1/x;  
        e=e+y;  
        ++i;  
    }  
}
```

/*未定义函数*/

/*未定义函数*/

/*未定义变量I*/