**实验报告撰写模版**

1、需求分析：说明程序设计的任务；

2、概要设计：说明程序中用到的抽象数据类型的定义、主程序的流程以及各程序模块之间的调用关系；

3、详细设计：实现程序模块的具体算法；

4、调试分析：调试过程中遇到的问题及解决方法；经验和体会等；

5、测试结果：列出测试结果，包括输入和输出；

6、附录：带注释的源程序。

**关于上机实验的几点说明**

1 、问题分析和任务定义

在进行设计之前，首先应该明确问题要求做什么，限制条件是什么。**注意：本步骤强调的是做什么，而不是怎么做。**对问题的描述应避开算法和所涉及的数据类型，而是对所需完成的任务作出明确的回答。例如：输入数据的类型、值的范围以及输入的形式；输出数据的类型、值的范围及输出的形式；若是会话式的输入，则结束标志是什么，是否接受非法的输入，对非法输入的回答方式是什么等等。这一步还应该为调试程序准备好测试数据，包括合法的输入数据和非法形式输入的数据。

2 、数据类型和系统设计

在设计这一步骤中需分逻辑设计和详细设计两步实现。

逻辑设计指的是，对问题描述中涉及的操作对象定义相应的数据类型，并按照以数据结构为中心的原则划分模块，定义主程序模块和各抽象数据类型。作为逻辑设计的结果，应写出每个抽象数据类型的定义（包括数据结构的描述和每个基本操作的规格说明），各个主要模块的算法，并画出模块之间的调用关系图。

详细设计则为定义相应的存储结构并写出各过程和函数的伪码算法。在这个过程中，要综合考虑系统功能，使得系统结构清晰、合理、简单和易于调试，抽象数据类型的实现尽可能做到数据封装，基本操作的规格说明尽可能明确具体。详细设汁的结果是对数据结构和基本操作的规格说明作出进一步的求精，写出数据存储结构的类型定义，按照算法书写规范用类 C 语言写出过程或函数形式的算法框架。在求精的过程中，应尽量避免陷入语言细节，不必过早表述辅助数据结构和局部变量。

3 、编码实现和静态检查

编码是把详细设计的结果进一步求精为程序设计语言程序。如何编写程序才能较快地完成调试是特别要注意的问题。程序的每行不要超过 60 个字符。每个过程（函数）体一般不要超过 40 行，最长不得超过 60 行，否则应该分割成较小的过程（函数）。要控制 if 语句连续嵌套的深度，分支过多时应考虑使用switch 语句。对函数功能和重要变量进行注释。一定要按格式书写程序，分清每条语句的层次，对齐括号，这样便于发现语法错误。

在上机之前，应该用笔在纸上写出详细的程序编码，并做认真地静态检查。多数初学者在编好程序后处于以下两种状态之一：一种是对自己的“ 精心作品” 的正确性确信不疑；另一种是认为上机前的任务已经完成，纠查错误是上机的工作。这两种态度是极为有害的。对一般的程序设计者而言，当编写的程序长度超过 50 行时，通常会含有语法错误或逻辑错误。上机动态调试决不能代替静态检查，否则调试效率将是极低的。静态检查主要有两种方法，一是用一组测试数据手工执行程序（通常应先检查单个模块）；二是通过阅读或给别人讲解自己的程序而深入全面地理解程序逻辑，在这个过程中再加入一些注解。

4 ．上机准备和上机调试

上机准备包括以下几个方面：

􀁺 熟悉 C 语言用户手册。

􀁺 注意 Turbo C 、 VC 与标准 C 语言之间的细微差别。

􀁺 熟悉机器的操作系统和语言集成环境的用户手册，尤其是最常用的命令操作，以便顺利进行上机的基本活动。

􀁺 掌握调试工具，考虑调试方案，设计测试数据并手工得出正确结果。“ 磨刀不误砍柴工” 。学生应该熟练运用高级语言的程序调试器 DEBUG 调试程序。

上机调试程序时要带一本高级语言教材或手册。调试最好分模块进行，自底向上，即先调试低层过程或函数。必要时可以另写一个调用驱动程序。这种表面上麻烦的工作实际上可以大大降低调试所面临的复杂性，提高调试工作效率。

在调试过程中可以不断借助 DEBUG 的各种功能，提高调试效率。调试中遇到的各种异常现象往往是预料不到的，此时不应“ 苦思冥想 ” ，而应借助系统提供的调试工具确定错误。调试正确后，认真整理源程序及其注释，印出带有完整注释的且格式良好的源程序清单和结果。

5 、总结和整理实验报告