

实验任务：

- 一、 学习和掌握 C 语言函数的定义与调用；
- 二、 进一步掌握随机数、选择结构、循环控制、交互式输入输出等程序设计方法；
- 三、 进一步掌握 C 语言中的数组定义和引用方法；
- 四、 学习和掌握根据流程图进行模块化程序设计的思想。

五、 提交正式的纸质版实验报告。对实验报告的要求：

- a) 实验报告文档的书写格式见实验报告模板。
- b) 实验报告内容要求有：
 - 实验目的
 - 实验要求
 - 实验程序详细流程图
 - 对流程图的简要说明
 - 实验结果
 - 对本课程的意见和建议
- c) 编译并运行你的程序。调试正确后将原程序工程文件目录压缩后提交到 Blackboard。其中压缩文件名称的前两个字母为你的姓与名的拼音的首字母。

六、 实验内容：给学龄前儿童编写一个 100 以内的四则运算游戏。程序的具体要求如下：

- 1) 在 VC++ 中建立一个工程；
- 2) 在程序开始的注释行中写上你自己的信息；
- 3) 定义一个函数 `randN1N2(int rN1, int rN2)`。该函数有两个整型的输入参数 `rN1` 和 `rN2`，其中， $rN1 \leq rN2$ 。该函数的功能是生成 `rN1` 到 `rN2` 之间均匀分布的随机整数，该函数的返回值类型是整型。每调用一次该函数，函数返回一个 `rN1` 至 `rN2` 之间的随机整数（随机数的生成方法可参考 Lab Assessment 3）。
- 4) 定义一个函数 `randOper4()`。该函数没有输入参数，函数的返回类型是整型。该函数内部调用上述 `randN1N2()` 函数。每调用一次函数 `randOper4()`，返回一个 1 至 4 之间的随机整数。
- 5) 定义一个函数 `randLmt100()`。该函数没有输入参数，函数的返回类型是整型。该函数内部调用上述 `randN1N2()` 函数。每调用一次函数 `randLmt100()`，返回一个 0-99 之间的随机整数。
- 6) 定义一个函数 `addLmt100(int num1, int num2)`。该函数有两个整型的输入参数 `num1` 和 `num2`；函数返回值类型为整型。该函数的功能是：在屏幕上列出一个算式 `num1 + num2 =`，然后接受一个键盘输入的计算答案 `answer`。若结果正确，则屏幕输出一句赞扬的语句，然后返回函数值 1；若结果错误，则屏幕输出一句鼓励的语句，并输出正确答案，然后返回函数值 0。
- 7) 定义一个函数 `subLmt100(int num1, int num2)`。该函数有两个整型输入参数 `num1` 和 `num2`；函数返回值类型为整型。该函数的功能是：在屏幕上列出一个算式 `numM1 - numM2 =`，其中：`numM1` 是 `num1` 和 `num2` 中较大的

一个数，numM2 是 num1 和 num2 中较小的一个数；然后接受一个键盘输入的计算答案 answer。若结果正确，则屏幕输出一句赞扬的语句，然后返回函数值 1；若结果错误，则屏幕输出一句鼓励的语句，并输出正确答案，然后返回函数值 0。

- 8) 定义一个函数 mulLmt100(int num1, int num2)。该函数有两个整型输入参数 num1 和 num2；函数返回值类型为整型。该函数的功能是：在屏幕上列出一个算式 $\text{num1} \times \text{num2} =$ ，然后接受一个键盘输入的计算答案 answer。若结果正确，则屏幕输出一句赞扬的语句，然后返回函数值 1；若结果错误，则屏幕输出一句鼓励的语句，并输出正确答案，然后返回函数值 0。
- 9) 定义一个函数 divLmt100(int num1, int num2)。该函数有两个整型输入参数 num1 和 num2；函数返回值类型为整型。该函数的功能是：首先判定 num1 是否可以整除 num2，若不能整除，则调用 randLmt100() 函数重新生成两个新的 num1 和 num2，然后再进行整除性判断，直到生成的 num1 可以整除 num2；在 num1 可以整除 num2 的情况下，在屏幕上列出一个算式 $\text{num2} \div \text{num1} =$ ，然后接受一个键盘输入的计算答案 answer。若结果正确，则屏幕输出一句赞扬的语句，然后返回函数值 1；若结果错误，则屏幕输出一句鼓励的语句，并输出正确答案，然后返回函数值 0。
- 10) **主函数的功能如下：**在主函数中调用上面定义的五個函数，在屏幕上随机循环出一道 100 以内的四则运算题。其中，定义一个整型数组 proNum[4] 和整型数组 ansNum[4]，这两个数组中分别存储在上述游戏循环过程中加、减、乘、除四种类型的题出现的次数，以及这四种类型的题被答对的次数。在用户不想继续游戏退出时，在屏幕上显示本次游戏中的加、减、乘、除四种类型题出现的数量，以及每一种题被答对的次数。主程序可参照附录 A 给出的参考流程图编写。
- 11) 在实验报告中要求说明：程序中那些地方使用了选择结构，那些地方使用了循环结构，那些地方使用了函数的嵌套调用。
- 12) 在实验报告中，画出实现函数 subLmt100(int num1, int num2) 和 divLmt100(int num1, int num2) 功能的流程图。

七、 思考题：对这个游戏，你还有什么改进的想法？请简要说明。

附 A：主程序的参考流程图（见下一页）：

