# 实验3-4：词法分析之DFA化简

## 一、实验目的

通过上机实习，掌握DFA化简的算法原理及实现技术，培养分析问题和解决问题的能力。

## 二、实验要求

在实验3-3的基础上，将实验3-3创建的DFA进行化简。

## 三、参考设计思路

对DFA进行化简，需要明确有穷状态机的多余状态的概念。多余状态是指从该自动机的开始状态出发，任何可识别的输入串也不能到达的状态。其次，需要明确等价状态的概念。设 =，。若对任何 ， 当且仅当 ，则称状态 和 是等价的。如果 和 不等价，则称 和 是可区别的。

DFA M的化简方法是把 M 的状态集 Q 分划成一些不相交的子集，使得每个子集中任何两个状态是等价的，而任何两个属于不同子集的状态都是可区别的；然后在每个子集中任取一个状态作“代表”，而删去子集中其余状态,并把射向其余状态的箭弧都改为射向作为“代表”的状态中。化简的算法思路如下：

（1）将DFA M 的状态集 Q 分成两个子集：终态集 和非终态集 ，形成初始分划 。

（2）对 建立新的分划 。对 的每个状态子集 ，进行如下变换:

1）把 分划成新的子集，使得 的两个状态 和 属于同一子集，当且仅当对任何输入符号，状态 和 转换到的状态都属于 的同一子集。

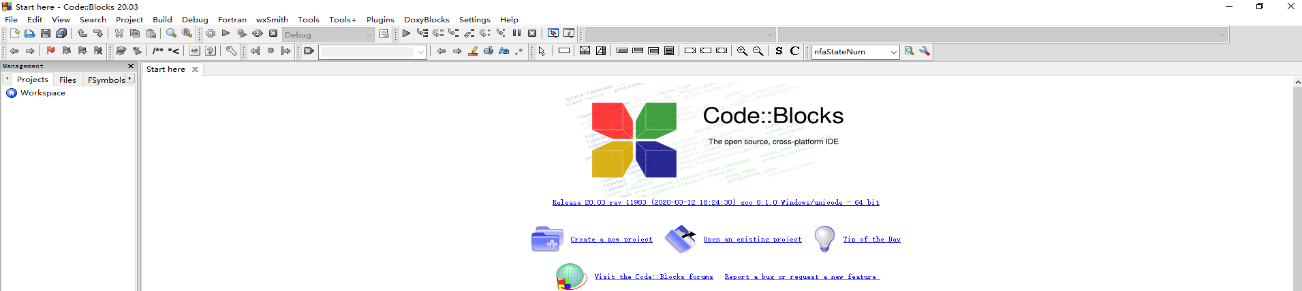
2）用 分划出的所有新子集替换 ，形成新的分划。

（3）如果 ，则执行第(4)步；否则令 ，重复第(2)步。

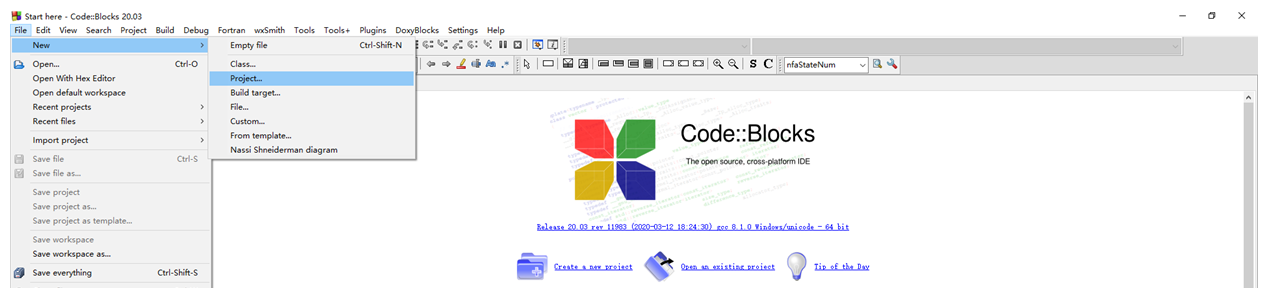
（4）分划结束后，对分划中的每个状态子集，选出一个状态作代表，而删去其他一切等价的状态，并把射向其他状态的箭弧改为射向这个作为代表的状态。这样得到的 DFA M' 是与DFA M 等价的一切 DFA 中状态数最少的DFA。

实验具体步骤如下：

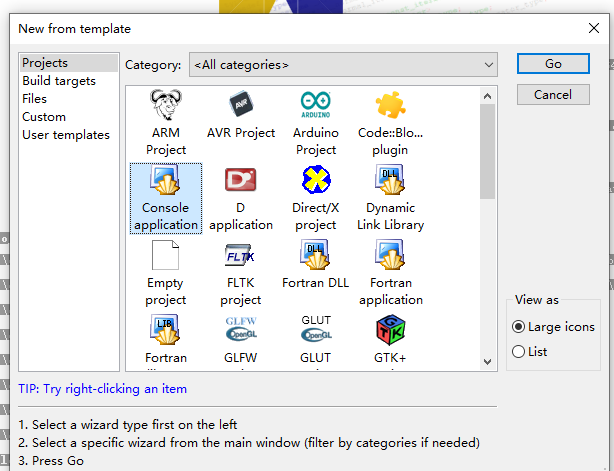
在虚拟机中，双击桌面上的“Code::Blocks”应用程序图标，启动“Code::Blocks”集成开发环境，如下图所示：



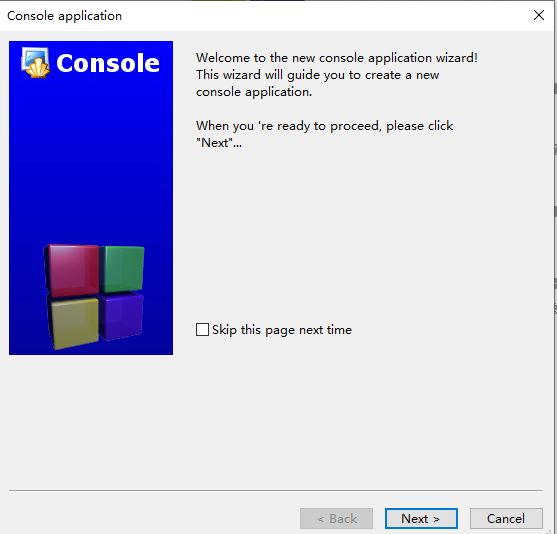
点击“File”-“New”-“project”，新建一个项目，



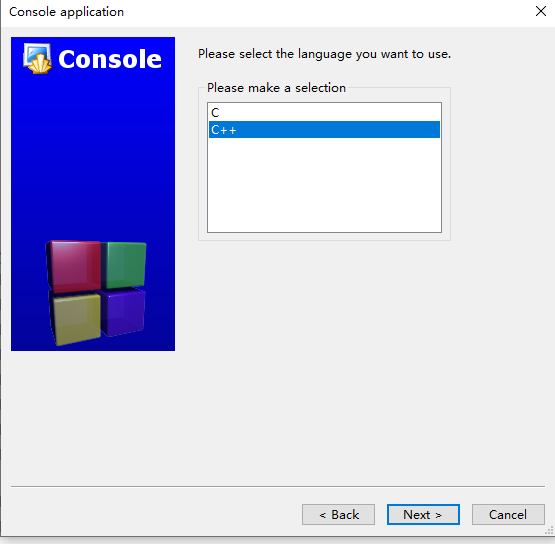
在新建项目类型中选择“Console application”，点击“Go”按钮，



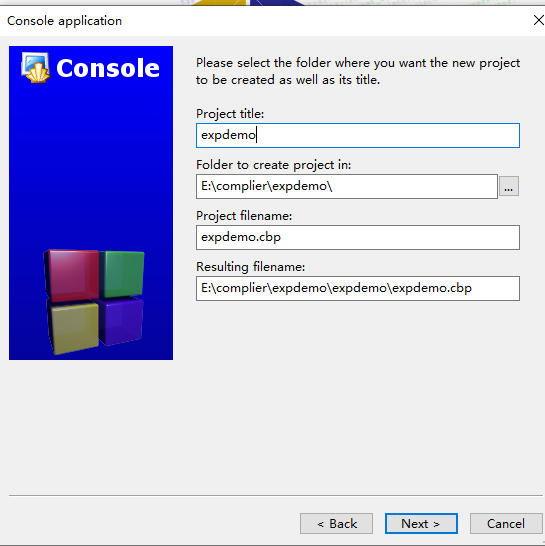
进入下面页面：



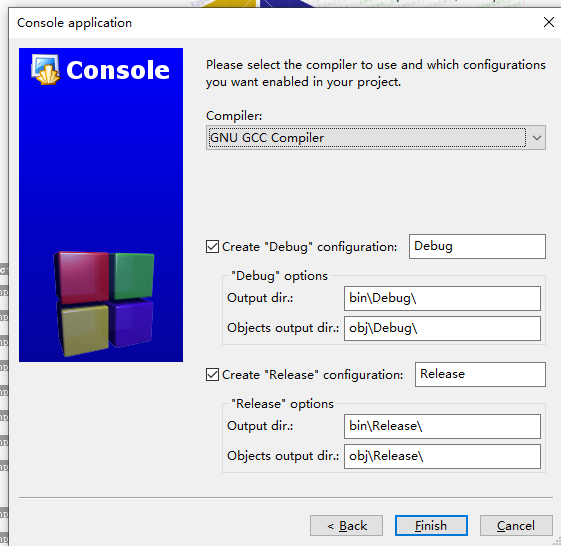
点击“Next”按钮，进入下面页面：



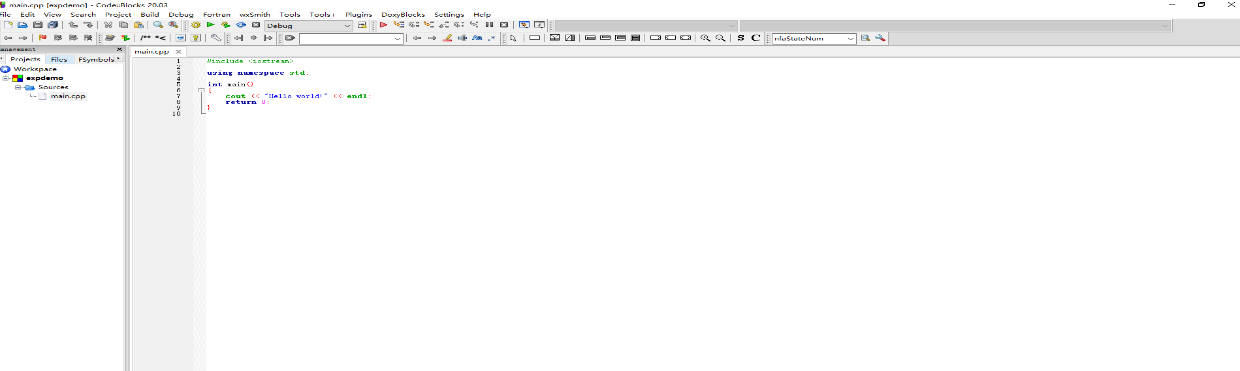
选择“C++”选项后，点击“Next”进入下一页面，在该页面中填入项目名“project title”，选择项目所在的位置，



填完项目名，选择好项目所在位置后，点击“Next”按钮，进入下面页面，不需要修改选项，点击“Finish”按钮，完成项目创建：

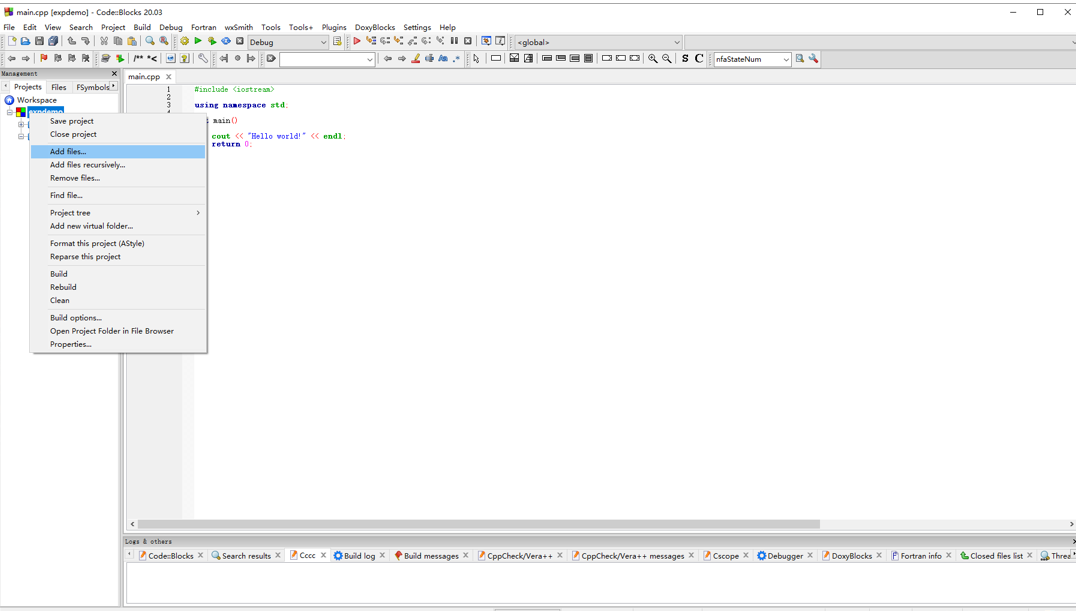


项目创建成功后，页面如下图所示，

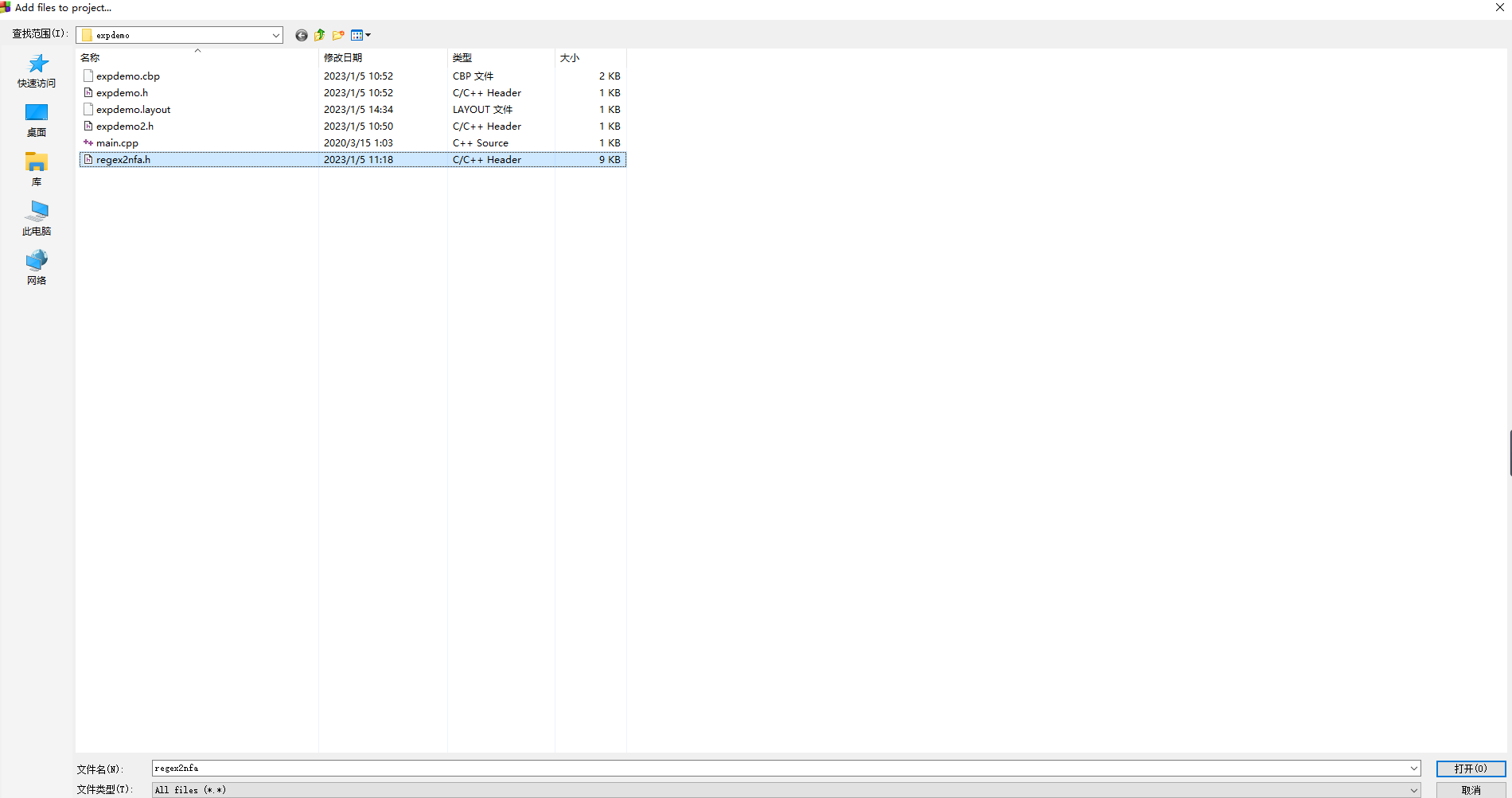


因为本次实验的NFA可由实验3-2方法构建，因此需要将实验3-2中编写的regex2nfa.h导入到本项目中，具体步骤如下：

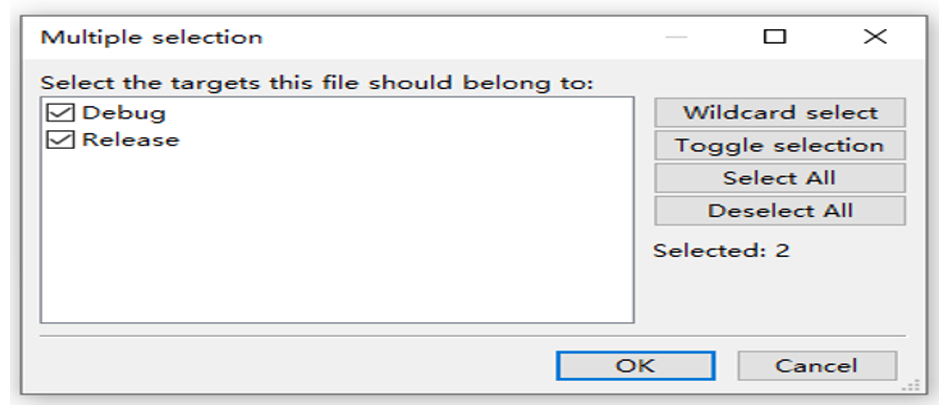
将regex2nfa.h文件拷贝到本次实验新建项目所在文件夹，打开项目之后，右键点击项目名，出现如下界面，



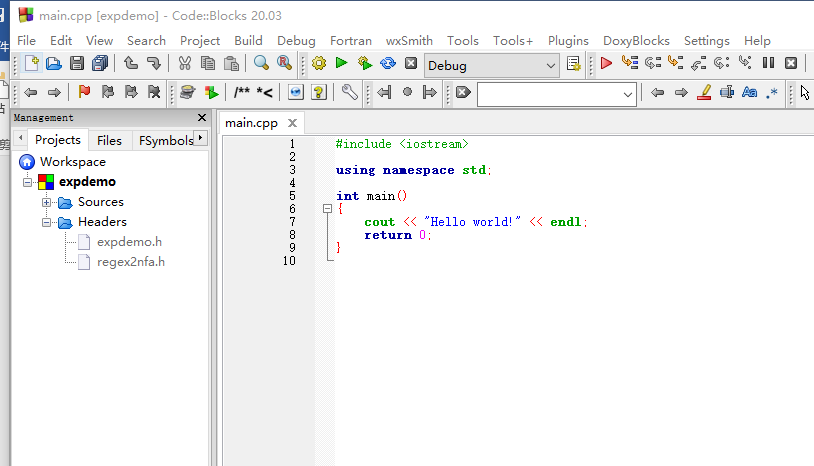
左键点击“Add files…”，出现如下对话框，



选择regex2nfa.h文件后，左键点击“打开”按钮，出现如下对话框：



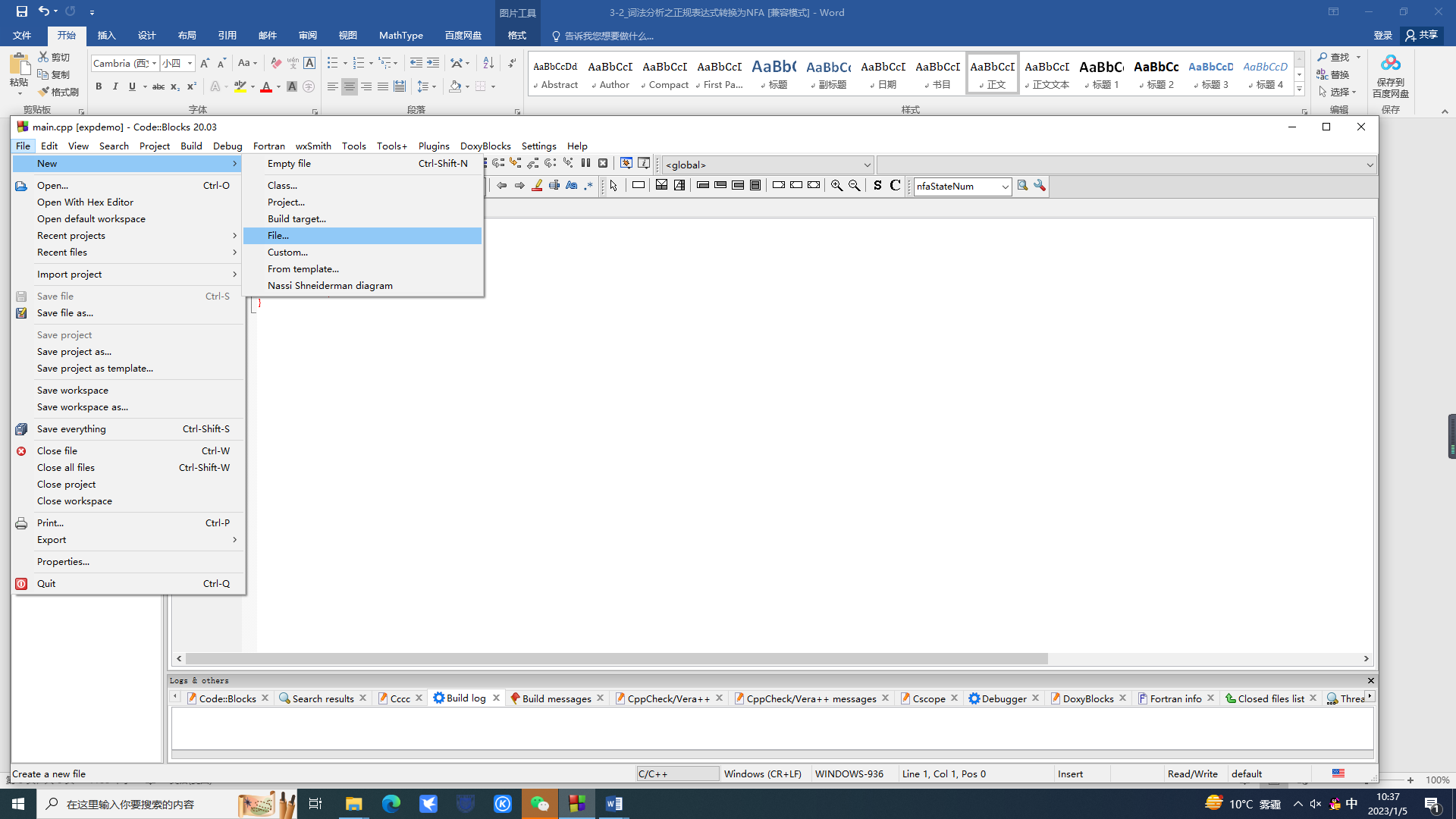
左键点击“OK”按钮后，就可以将regex2nfa.h文件加入到本实验所建的工程中了。



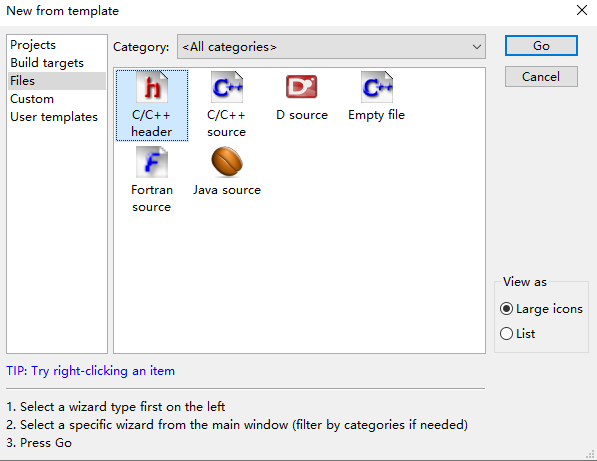
因为本次实验的DFA可由实验3-3方法构建，因此需要将实验3-3中编写的nfa2dfa.h导入到本项目中，具体步骤可参考上述regex2nfa.h文件导入方法。

本次实验核心代码放在mindfa.h文件中，方便后续实验调用，新建\*.h文件步骤如下：

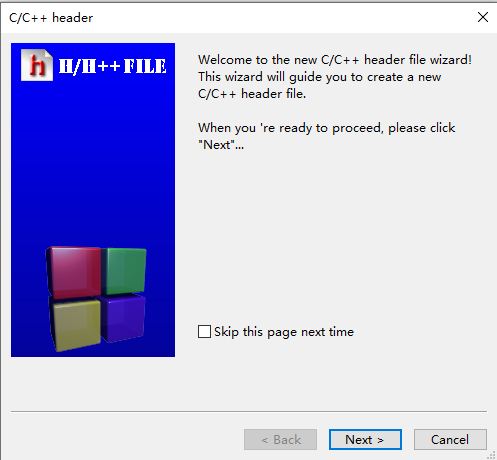
点击“File”-“New”-“file”，新建一个文件，



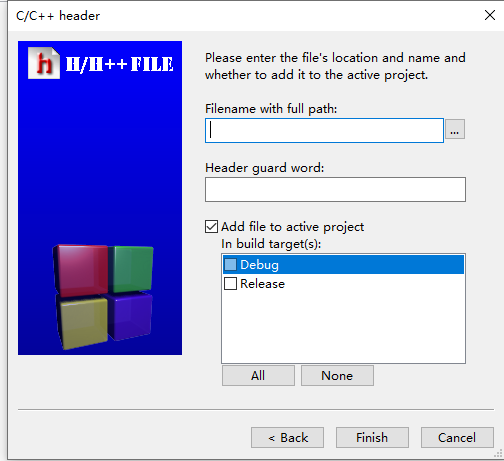
在新建文件类型中选择“C/C++ header”类型，点击“Go”按钮，



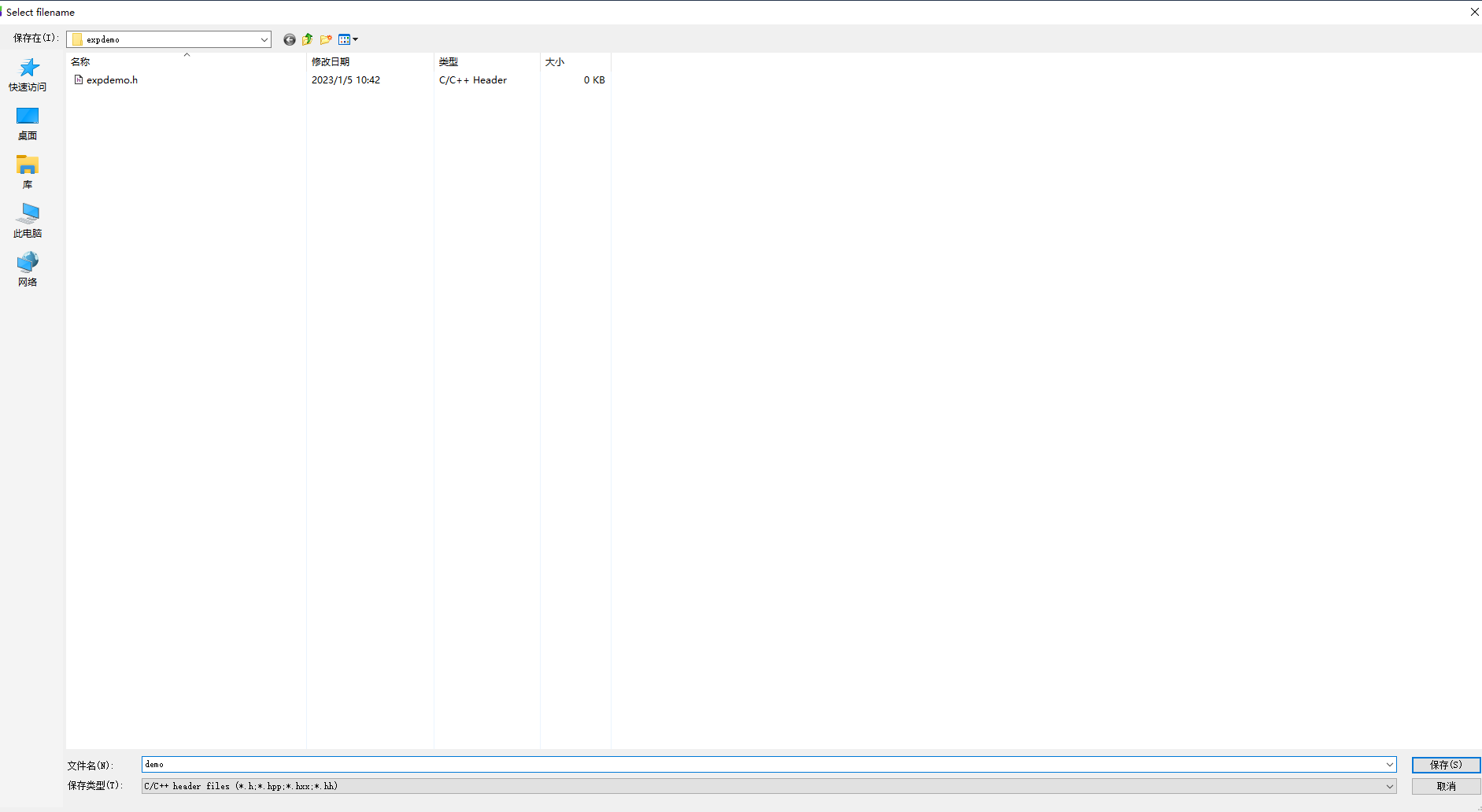
进入下面页面，



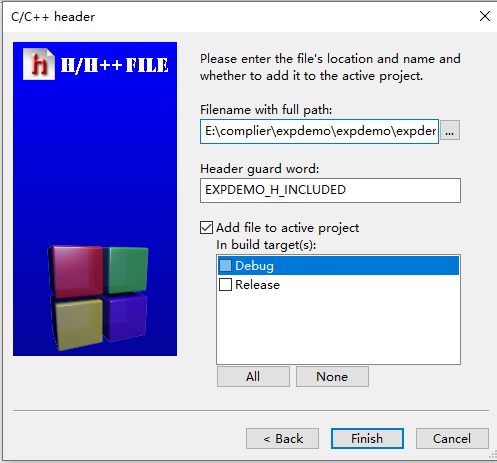
点击“Next”按钮，进入下面页面，



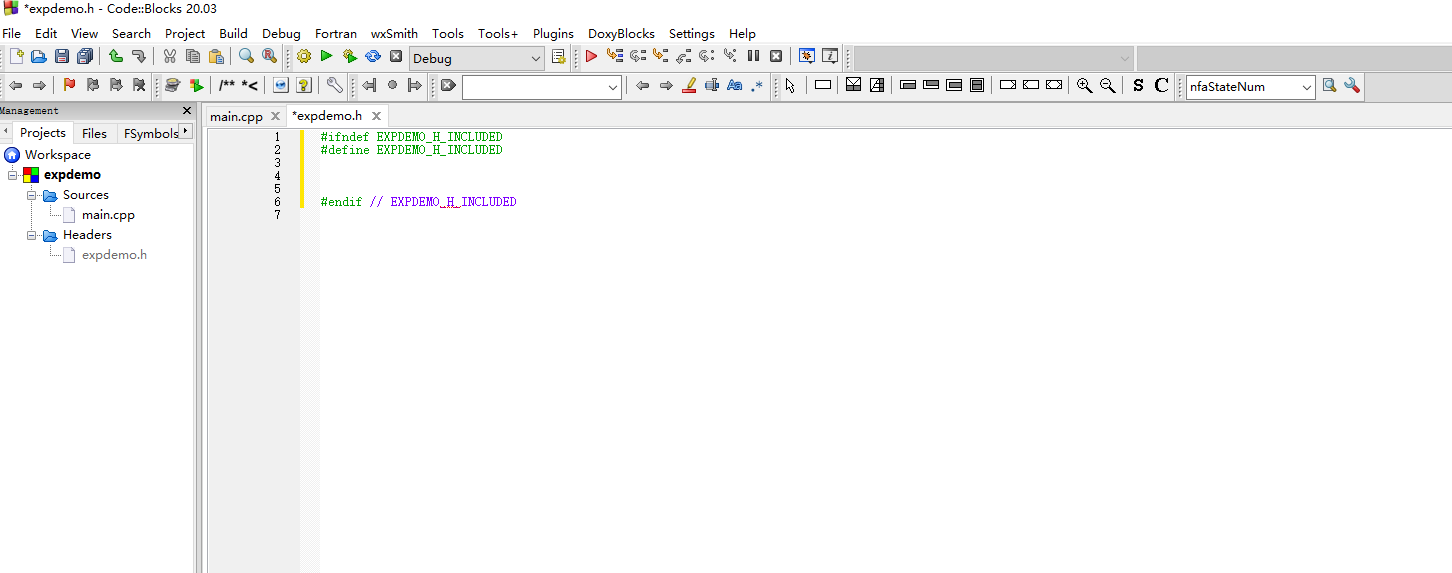
在“Filename with full path”下，通过点击“…”进入下面页面，选择文件所在的位置（建议和主函数放在同一个文件夹下），



填入文件名，点击确认后进入下面页面：

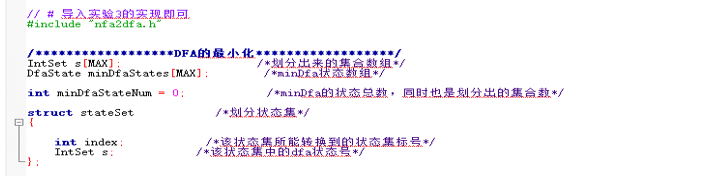


点击“Finish”按钮后，生成\*.h文件，如下图所示：



可以在新建的头文件中（下面以mindfa.h文件名），按下面步骤完成程序编写：

首先进行初始化定义如下：

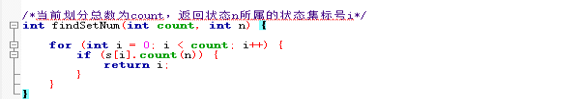


编写mindfa()函数如下：





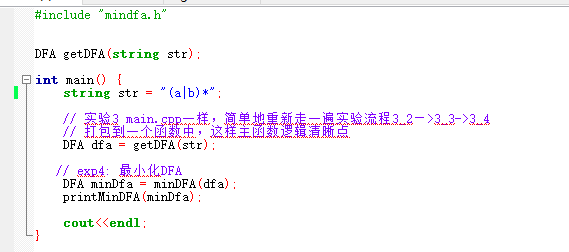
需要自行编写findSetNum()函数如下：



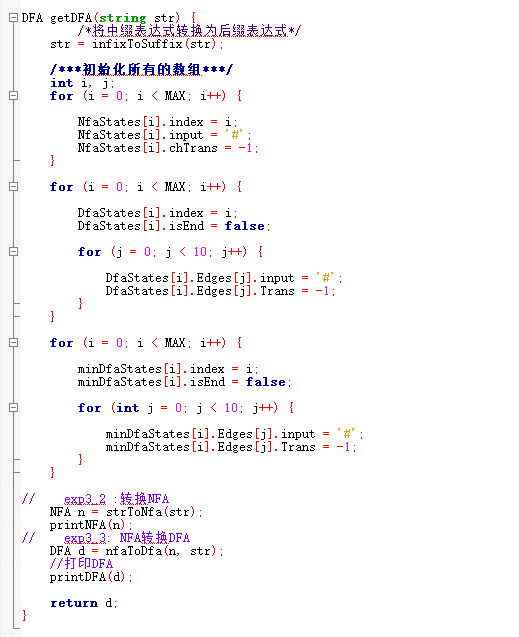
为了 将化简后的DFA打印出来，可以编写printMinDFA ()函数如下：



在完成mindfa.h文件编写后，在main.cpp中按照下面步骤完成程序编写：

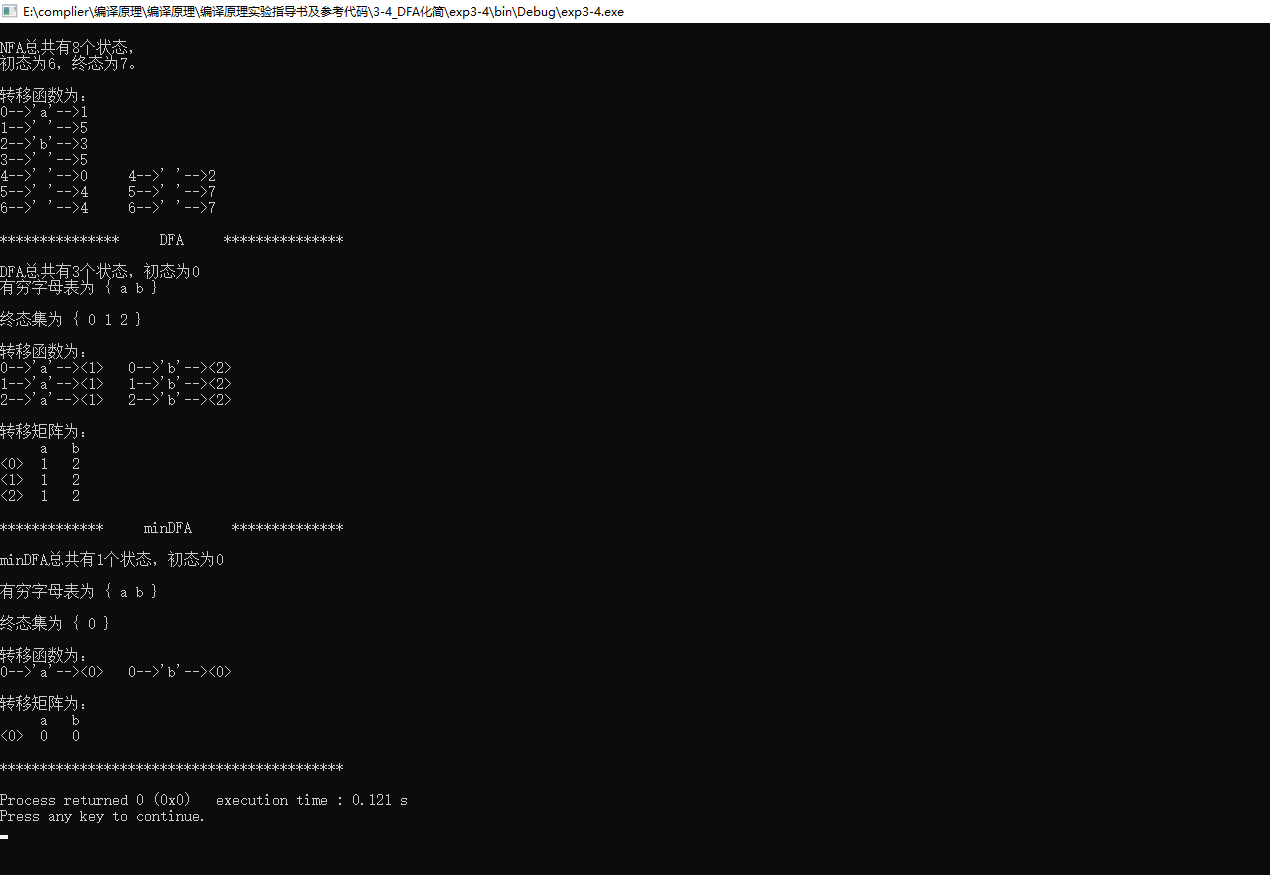


在main()函数中，需要自行编写getDFA()函数如下：

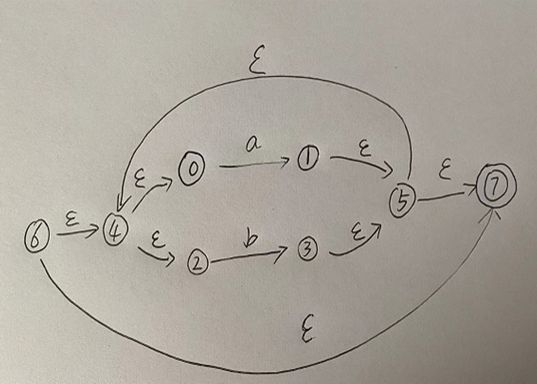


本次实验代码在“exp3\_4”文件夹下，双击“exp3-4.cbp”即可打开项目。

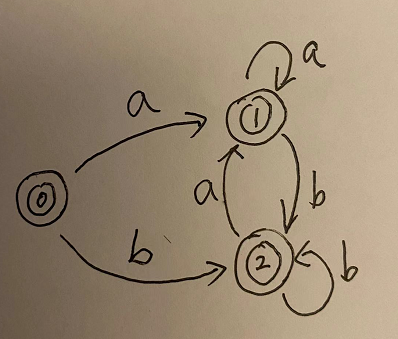
点击“Run”运行结果下图所示：



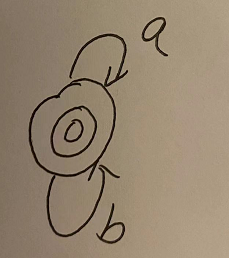
其中NFA所对应的状态转换图为：



转换后的DFA所对应的状态转换图为：



化简后的DFA所对应的状态转换图为：



**四、总结**

通过本次实验学习，要求学员掌握编写DFA化简的方法。