实验报告（NFA 到 DFA 的转化）

课程编号：3132112040　实践课程名称：形式语言与自动机　　学年：2022-2023　学期：4

学生姓名	张梓良	学号	2021212484
指导教师姓名	杨正球	起止时间	2023-03-21 17:08 至 2023-04-13 23:59
项目名称	NFA 到 DFA 的转化的输入、输出函数设计		

项目内容

负责内容：

1. 输入函数 input 和输出函数 output 的设计、编写以及测式。
2. 报告中设计思想的部分编写。

input()函数的设计思想：

第一行的数据是固定的，因此并不是有效输入。第一列的原始状态下标是按行数递增的，因此不用特地存储，用数组或 vector 向量下标代替即可。

对于存储 NFA 的状态，以及其对应的的转移状态，考虑到有 2 个要素“自身”“输入 0\1 的转移状态”。可以用一个三维 vector 类型的变量 Q1 来存储，其中 Q1 的 size 就是 NFA 的状态个数。Q1[0][0]存储 q0 输入‘0’后转移的状态，Q1[0][1]存储 q0 输入‘1’后转移的状态，Q1[1][0]存储 q1 输入‘0’后转移的状态……其中，若转移后的状态为‘N’，则用‘-1’来代替。

在实际处理过程中，新开两个临时变量 vector<vector<int>> temp1;和 vector<int> temp2;分别用于临时存储一个状态的所有转移后的状态集合以及每一个转移后的状态集合。

对于读入的每一行转移函数，采用 getline(cin, s)的方式以 string 的形式读入。对于读入的每一行字符串，首先使用 string 类的内置 find()函数查找‘e’判断是否为终止状态，再通过 find()函数查找‘]’快速定位到转移后的状态处，对转移后的状态进行处理，将它们的下标存储到 temp2 对应的位置中。

对于状态下标的处理，由于是采用字符串的形式读入的，因此需要将一串数字字符转换成对应的 int 型整数，首先采用 isdigit()函数判断当前字符是否为数字字符，再采用 num = num * 10 + (int)(s[loc] - '0')方式将数字字符转换成对应的数字。

对于读到空集，采用简单 if 语句判断当前字符是否为‘N’即可。若为‘N’,说明读到空集，将-1 存储到 temp2 对应的位置中。

对于区分输入‘0’和‘1’的不同转移状态，采用判断当前字符是否为‘]’的方式来判断是否已经将一个状态转移集合[q0,q1,q2...]读完，若已经读完，则将 temp2 整体存储到 temp1 中，并清空 temp2 用于下一个转移状态集合的存储。此时需要注意，读到空集属于特殊情况，不能采用判断是否读到‘]’的方式来判断一个状态集合是否已经读完，而应是读到‘N’就认为一个转移状态集合已经读完，再按上述方法进行处

	理。 最终，每处理完一行，将 temp1 整体存储到 Q1 中，并清空 temp1 和 temp2，进行下一行的处理。 **output()函数的设计思想：** 按照题目的要求，将 Q2 中的状态集编号排序后，以表格的形式依次输出即可。需要注意的是，起始状态需要额外输出(s)，而终结状态需要额外输出(e)，遇到‘-1’需要输出‘N’。在每一行的输出中，首先判断原始状态是否为终止状态，通过对存储终止状态的集合 isfinal2 使用 set 内置函数 count()进行判断。其次再判断转移后的状态是否为空，通过使用 if 语句判断 Q2[i].to[j]是否为‘-1’进行判断。
结论	1. 通过该实验对 NFA 的形式定义有了更好的理解，对 NFA 相较于 DFA 最大的区别：输入一个符号，可能存在多个转移状态，得到的是转移状态集有了更加深刻的认识。 2. 通过该实验对 NFA 到 DFA 转换的形式化的处理有了更加深刻的理解，掌握了如何通过程序语言模拟和实现这一过程。 3. 通过该实验，通过编程进行模拟的能力得到了提高。能够更加熟练地处理比较不规整的输入，从中准确提取所需的信息并采用合适的方法进行存储。 4. 通过该实验对 C++STL 库的使用有了更加熟练的掌握，能够灵活应用 STL 容器及其内置函数降低编程难度。 5. 通过该实验，提升了团队合作和多人协同开发的能力。