实验报告：

1. 你采用什么数据结构存储状态集合 Q和输入符号集合Sigma的？为什么采用这种结构？

采用的是字符串双向键值对，字符对应编码，用不同编码来表示不同字符。

2. 你采用什么方式存储状态转移函数delta的？为什么采用这种结构？

采用二进制字符串存储了状态转移函数delta，与前面状态集合Q和输入集合Sigma的编码进行比对 然后 按delta进行图灵机的运行。

3. 对于具体图灵机和纸带信息的读入，你做了哪些检查？为什么需要这些检查？

对图灵机做了各个集合符号数统计，用来分配空间，对纸带信息进行了是否是Sigma字符的检查，若不是则退出程序。这些检查保证了图灵机运行的正确性。

4. 自行编写一个字符串翻转的具体图灵机程序，给出其描述规范，使用你给出的通用图灵机程序执行该具体图灵机，并检测字符串翻转图灵机知否正确。

实现字符串反转的规则：

Q: s0, s1, s2, s3, s4, s5, s6, s7, s8, s9, s10;

Sigma: #, a, b, c, x, y, z;

q0: s0;

Accept: s10;

Reject: s1, s2;

Delta:

( s0, #)->( s1, #, R),

( s1, #)->( s1, #, R),

( s1, a)->( s2, a, R),

( s1, b)->( s2, b, R),

( s1, c)->( s2, c, R),

( s2, a)->( s2, a, R),

( s2, b)->( s2, b, R),

( s2, c)->( s2, c, R),

( s2, #)->( s3, #, L),

( s3, a)->( s4, x, R),

( s3, b)->( s5, y, R),

( s3, c)->( s6, z, R),

( s4, a)->( s4, a, R),

( s4, b)->( s4, b, R),

( s4, c)->( s4, c, R),

( s4, x)->( s4, x, R),

( s4, y)->( s4, y, R),

( s4, z)->( s4, z, R),

( s4, #)->( s7, a, L),

( s5, a)->( s5, a, R),

( s5, b)->( s5, b, R),

( s5, c)->( s5, c, R),

( s5, x)->( s5, x, R),

( s5, y)->( s5, y, R),

( s5, z)->( s5, z, R),

( s5, #)->( s7, b, L),

( s6, a)->( s6, a, R),

( s6, b)->( s6, b, R),

( s6, c)->( s6, c, R),

( s6, x)->( s6, x, R),

( s6, y)->( s6, y, R),

( s6, z)->( s6, z, R),

( s6, #)->( s7, c, L),

( s7, a)->( s7, a, L),

( s7, b)->( s7, b, L),

( s7, c)->( s7, c, L),

( s7, x)->( s8, x, L),

( s7, y)->( s8, y, L),

( s7, z)->( s8, z, L),

( s7, #)->( s7, #, L),

( s8, x)->( s8, x, L),

( s8, y)->( s8, y, L),

( s8, z)->( s8, z, L),

( s8, a)->( s4, x, R),

( s8, b)->( s5, y, R),

( s8, c)->( s6, z, R),

( s8, #)->( s9, #, R),

( s9, x)->( s9, #, R),

( s9, y)->( s9, #, R),

( s9, z)->( s9, #, R),

( s9, a)->( s10, a, -),

( s9, b)->( s10, b, -),

( s9, c)->( s10, c, -);

5. 图灵机与程序、计算机有什么差别与联系？

计算机也是一种图灵机模型，图灵机是一种计算模型，是一种核心驱动力，只有给出规则，才能转化为实现具体功能的图灵机。程序是一条条指令的集合。