Exercise 1. Write a Turing machine that decides the predicate "x is even".[1]

Solution. The idea is the following: we are given an input n represented as a sequence of n symbols '1' delimited by two '#'. After having erased the initial #, we start reading the sequence of 1s, and simultaneously we erase these 1s. We start reading the body of the number in a state q_y, which tracks the fact that the number of 1s seen so far (namely, 0) is even. If we encounter a 1, we switch to a state q_n, which tracks the fact that the number of 1s seen so far is odd. If we encounter one more 1, we switch back to state q_y, and so on. Now, if the input is even, when we hit upon the final delimiter # we will be in state q_y; if the input is odd, we will be in state q_n. In the latter case, we give the machine the instructions to go left, print #, and stop, which leaves the code for 0 on the tape. In the former case, we give the machine instructions to go left, print 1, go left once more, print #, and stop, which leaves the code for 1.[2]

Concretely, here is one program that carries out the procedure just described, based on the set of states $Q = \{q_0, q_y, q_n, q_t, q_1\}$, where q_0 is the initial state.

1. $\langle q_0, \#, B, \rightarrow, q_y\rangle$
2. $\langle q_y, 1, B, \rightarrow, q_n\rangle$
3. $\langle q_n, 1, B, \rightarrow, q_y\rangle$
4. $\langle q_n, \#, \#, \leftarrow, q_t\rangle$
5. $\langle q_y, \#, \#, \leftarrow, q_1\rangle$
6. $\langle q_1, B, 1, \leftarrow, q_t\rangle$
7. $\langle q_t, B, \#, -, q_t\rangle$

Exercise 2. Write a Turing machine that computes the function $f(x) = 2x$.

Solution. The idea is the following: we start erasing the left delimiter #. Then we erase the first symbol 1 from the input, go right until we meet the first blank cell, and write down two 1s. Then we come back to the leftmost 1, erase it, go right until the first blank cell, write down two 1s, etc. Once there are no more 1s in the initial string, we go right until the first blank cell, print #, and stop. Since in this procedure each occurrence of 1 was replaced by two occurrences, this will leave the code for the double of the initial number on the tape.

We base our program on the a set of states $Q = \{q_0, q_1, q_2, q_3, q_4,_5\}$, which can be read as follows:

- q_0 (initial state): erase #, move right and enter state q_1;
- q_1: if you find a 1, erase it, move right and enter state q_2; if you see #, go to the final state q_5;
- q_2: go right until the first blank, write down 1, go right and enter q_3;

[1] A Turing machine decides a predicate M if it computes the characteristic function c_M.

[2] Writing the output to the right of the delimiter #, rather than on the left, would work just as well.

- q_3: write down one more 1, go left and enter q_4;
- q_4: go left until the first blank, then go right and back to state q_1;
- q_5: go right until the first blank, then write down # and stop.

This should help reading the program instructions:

1. $\langle q_0, \#, B, \rightarrow, q_1 \rangle$
2. $\langle q_1, 1, B, \rightarrow, q_2 \rangle$
3. $\langle q_2, 1, 1, \rightarrow, q_2 \rangle$
4. $\langle q_2, \#, \#, \rightarrow, q_2 \rangle$
5. $\langle q_2, B, 1, \rightarrow, q_3 \rangle$
6. $\langle q_3, B, 1, \leftarrow, q_4 \rangle$
7. $\langle q_4, 1, 1, \leftarrow, q_4 \rangle$
8. $\langle q_4, \#, \#, \leftarrow, q_4 \rangle$
9. $\langle q_4, B, B, \rightarrow, q_1 \rangle$
10. $\langle q_1, \#, \#, \rightarrow, q_5 \rangle$
11. $\langle q_5, 1, 1, \rightarrow, q_5 \rangle$
12. $\langle q_5, B, \#, -, q_5 \rangle$