```
GHCi, version 8.0.2: http://www.haskell.org/ghc/  :? for help
Prelude> :l hw9
[1 of 2] Compiling Turing            ( Turing.hs, interpreted )
[2 of 2] Compiling Main              ( hw9.hs, interpreted )
Ok, modules loaded: Main, Turing.
*Main> stepRun countDGGD ""
0: ScanD1        [ ]
*Main> stepRun countDGGD "gd"
0: ScanD1        [g]d
1: ScanD1        g[d]
2: ScanG1        gd[ ]
*Main> stepRun countDGGD "dgd"
0: ScanD1        [d]gd
1: ScanG1        d[g]d
2: ScanG2        dg[d]
3: ScanG1        dgd[ ]
*Main> stepRun countDGGD "dggg"
0: ScanD1        [d]ggg
1: ScanG1        d[g]gg
2: ScanG2        dg[g]g
3: ScanD2        dgg[g]
4: ScanD1        dggg[ ]
*Main> stepRun countDGGD "dggd"
0: ScanD1        [d]ggd
1: ScanG1        d[g]gd
2: ScanG2        dg[g]d
3: ScanD2        dgg[d]
4: tally         dggD[ ]
5: back          dgg[D]1
6: ScanG1        dggd[1]
*Main> stepRun countDGGD "ddggddddggdggdggggd"
0: ScanD1        [d]dggddddggdggdggggd
1: ScanG1        d[d]ggddddggdggdggggd
3: ScanG2        ddg[g]ddddggdggdggggd
4: ScanD2        ddgg[d]dddggdggdggggd
5: tally         ddggD[d]ddggdggdggggd
19: tally        ddggDdddggdggdggggd[ ]
20: back         ddggDdddggdggdgggg[d]1
34: back         ddgg[D]dddggdggdggggd1
38: ScanG1       ddggdddd[g]gdggdggggd1
39: ScanG2       ddggddddg[g]dggdggggd1
40: ScanD2       ddggddddgg[d]ggdggggd1
60: back         ddggddddgg[D]ggdggggd11
61: ScanG1       ddggddddggd[g]gdggggd11
62: ScanG2       ddggddddggdg[g]dggggd11
63: ScanD2       ddggddddggdgg[d]ggggd11
80: ScanG1       ddggddddggdggd[g]gggd111
81: ScanG2       ddggddddggdggdg[g]ggd111
82: ScanD2       ddggddddggdggdgg[g]gd111
83: ScanD1       ddggddddggdggdggg[g]d111
84: ScanD1       ddggddddggdggdgggg[d]111
85: ScanG1       ddggddddggdggdggggd[1]11
*Main>
*Main> stepRun rev ""
0: read          [ ]
1: Clean         [ ]
*Main> stepRun rev "m"
0: read          [m]
1: writeM        [ ]x
2: scroll        m[x]
3: read          mx[ ]
4: Clean         m[x]
5: Clean         [m]
*Main> stepRun rev "qm"
0: read          [q]m
1: writeQ        [ ]xm
2: scroll        q[x]m
3: read          qx[m]
4: writeM        q[x]x
6: writeM        [ ]qxx
7: scroll        m[q]xx
8: scroll        mq[x]x
```

```
9: read          mqx[x]
10: read         mqxx[ ]
11: Clean        mqx[x]
12: Clean        mq[x]
13: Clean        m[q]
*Main> stepRun rev "mqqmmmq"
0: read          [m]qqmmmq
1: writeM        [ ]xqqmmmq
2: scroll        m[x]qqmmmq
3: read          mx[q]qmmmq
4: writeQ        m[x]xqmmmq
6: writeQ        [ ]mxxqmmmq
7: scroll        q[m]xxqmmmq
8: scroll        qm[x]xqmmmq
9: read          qmx[x]qmmmq
10: read         qmxx[q]mmmq
11: writeQ       qmx[x]xmmmq
16: scroll       q[q]mxxxmmmq
21: read         qqmxxx[m]mmq
28: writeM       [ ]qqmxxxxmmq
36: read         mqqmxxxx[m]mq
55: read         mmqqmxxxxx[m]q
78: read         mmmqqmxxxxxx[q]
105: read        qmmmqqmxxxxxxx[ ]
106: Clean       qmmmqqmxxxxxx[x]
107: Clean       qmmmqqmxxxxx[x]
113: Clean       qmmmqq[m]
*Main>
*Main> stepRun more4's ""
0: start         [ ]
1: Done          N[ ]
*Main> stepRun more4's "3"
0: start         [3]
1: dropY          [ ]
2: less          [ ]
3: Done          N[ ]
*Main> stepRun more4's "4"
0: start         [4]
1: dropN          [ ]
2: more          [ ]
3: Done          Y[ ]
*Main> stepRun more4's "5443"
0: start         [5]443
1: dropY          [4]43
2: back          [ ]x43
3: start          [x]43
4: start           [4]3
5: dropN            [3]
6: back            [ ]x
7: start            [x]
8: start             [ ]
9: Done             N[ ]
*Main> stepRun more4's "345544454"
0: start         [3]45544454
1: dropY          [4]5544454
2: back          [ ]x5544454
3: start          [x]5544454
4: start           [5]544454
6: dropY            5[4]4454
9: start            [5]x4454
11: dropY            x[4]454
13: back            [ ]xx454
16: start              [4]54
17: dropN               [5]4
18: back               [ ]x4
19: start               [x]4
20: start                [4]
21: dropN                 [ ]
22: more                 [ ]
23: Done                 Y[ ]
*Main>
*Main> stepRun xor "1#1"
```

```
0: Read          [1]#1
1: left1          [#]1
2: xor1           #[1]
3: back           [#]F
4: back          [ ]#F
5: Read           [#]F
6: Read            [F]
7: Read            0[ ]
*Main> stepRun xor "0#1"
0: Read          [0]#1
1: left0          [#]1
2: xor0           #[1]
3: back           [#]T
4: back          [ ]#T
5: Read           [#]T
6: Read            [T]
7: Read            1[ ]
*Main> stepRun xor "10101#10011"
0: Read          [1]0101#10011
1: left1          [0]101#10011
6: xor1           0101#[1]0011
7: back           0101[#]F0011
13: Read          [0]101#F0011
14: left0          [1]01#F0011
18: xor0           101#[F]0011
19: xor0           101#F[0]011
20: back           101#[F]F011
26: Read           [1]01#FF011
32: xor1            01#FF[0]11
33: back            01#F[F]T11
39: Read            [0]1#FFT11
45: xor0             1#FFT[1]1
52: Read             [1]#FFTT1
58: xor1              #FFTT[1]
65: Read              [#]FFTTF
66: Read               [F]FTTF
67: Read               0[F]TTF
68: Read               00[T]TF
71: Read               00110[ ]
*Main>
```

```
GHCi, version 8.0.2: http://www.haskell.org/ghc/  :? for help
Prelude> :l hw9
[1 of 2] Compiling Turing            ( Turing.hs, interpreted )
[2 of 2] Compiling Main              ( hw9.hs, interpreted )
Ok, modules loaded: Main, Turing.
*Main> stepRun countDGGD ""
0: ScanD1        [ ]
*Main> stepRun countDGGD "gd"
0: ScanD1        [g]d
1: ScanD1        g[d]
2: ScanG1        gd[ ]
*Main> stepRun countDGGD "dgd"
0: ScanD1        [d]gd
1: ScanG1        d[g]d
2: ScanG2        dg[d]
3: ScanG1        dgd[ ]
*Main> stepRun countDGGD "dggg"
0: ScanD1        [d]ggg
1: ScanG1        d[g]gg
2: ScanG2        dg[g]g
3: ScanD2        dgg[g]
4: ScanD1        dggg[ ]
*Main> stepRun countDGGD "dggd"
0: ScanD1        [d]ggd
1: ScanG1        d[g]gd
2: ScanG2        dg[g]d
3: ScanD2        dgg[d]
4: tally         dggD[ ]
5: back          dgg[D]1
6: ScanG1        dggd[1]
*Main> stepRun countDGGD "ddggddddggdggdggggd"
0: ScanD1        [d]dggddddggdggdggggd
1: ScanG1        d[d]ggddddggdggdggggd
3: ScanG2        ddg[g]ddddggdggdggggd
4: ScanD2        ddgg[d]dddggdggdggggd
5: tally         ddggD[d]ddggdggdggggd
19: tally        ddggDdddggdggdggggd[ ]
20: back         ddggDdddggdggdgggg[d]1
34: back         ddgg[D]dddggdggdggggd1
38: ScanG1       ddggdddd[g]gdggdggggd1
39: ScanG2       ddggddddg[g]dggdggggd1
40: ScanD2       ddggddddgg[d]ggdggggd1
60: back         ddggddddgg[D]ggdggggd11
61: ScanG1       ddggddddggd[g]gdggggd11
62: ScanG2       ddggddddggdg[g]dggggd11
63: ScanD2       ddggddddggdgg[d]ggggd11
80: ScanG1       ddggddddggdggd[g]gggd111
81: ScanG2       ddggddddggdggdg[g]ggd111
82: ScanD2       ddggddddggdggdgg[g]gd111
83: ScanD1       ddggddddggdggdggg[g]d111
84: ScanD1       ddggddddggdggdgggg[d]111
85: ScanG1       ddggddddggdggdggggd[1]11
*Main>
*Main> stepRun rev ""
0: read          [ ]
1: Clean         [ ]
*Main> stepRun rev "m"
0: read          [m]
1: writeM        [ ]x
2: scroll        m[x]
3: read          mx[ ]
4: Clean         m[x]
5: Clean         [m]
*Main> stepRun rev "qm"
0: read          [q]m
1: writeQ        [ ]xm
2: scroll        q[x]m
3: read          qx[m]
4: writeM        q[x]x
6: writeM        [ ]qxx
7: scroll        m[q]xx
8: scroll        mq[x]x
```

```
9: read          mqx[x]
10: read         mqxx[ ]
11: Clean        mqx[x]
12: Clean        mq[x]
13: Clean        m[q]
*Main> stepRun rev "mqqmmmq"
0: read          [m]qqmmmq
1: writeM        [ ]xqqmmmq
2: scroll        m[x]qqmmmq
3: read          mx[q]qmmmq
4: writeQ        m[x]xqmmmq
6: writeQ        [ ]mxxqmmmq
7: scroll        q[m]xxqmmmq
8: scroll        qm[x]xqmmmq
9: read          qmx[x]qmmmq
10: read         qmxx[q]mmmq
11: writeQ       qmx[x]xmmmq
16: scroll       q[q]mxxxmmmq
21: read         qqmxxx[m]mmq
28: writeM       [ ]qqmxxxxmmq
36: read         mqqmxxxx[m]mq
55: read         mmqqmxxxxx[m]q
78: read         mmmqqmxxxxxx[q]
105: read        qmmmqqmxxxxxxx[ ]
106: Clean       qmmmqqmxxxxxx[x]
107: Clean       qmmmqqmxxxxx[x]
113: Clean       qmmmqq[m]
*Main>
*Main> stepRun more4's ""
0: start         [ ]
1: Done          N[ ]
*Main> stepRun more4's "3"
0: start         [3]
1: dropY          [ ]
2: less          [ ]
3: Done          N[ ]
*Main> stepRun more4's "4"
0: start         [4]
1: dropN          [ ]
2: more          [ ]
3: Done          Y[ ]
*Main> stepRun more4's "5443"
0: start         [5]443
1: dropY          [4]43
2: back          [ ]x43
3: start          [x]43
4: start           [4]3
5: dropN            [3]
6: back            [ ]x
7: start            [x]
8: start             [ ]
9: Donc             N[ ]
*Main> stepRun more4's "345544454"
0: start         [3]45544454
1: dropY          [4]5544454
2: back          [ ]x5544454
3: start          [x]5544454
4: start           [5]544454
6: dropY            5[4]4454
9: start            [5]x4454
11: dropY            x[4]454
13: back            [ ]xx454
16: start              [4]54
17: dropN               [5]4
18: back               [ ]x4
19: start               [x]4
20: start                [4]
21: dropN                 [ ]
22: more                 [ ]
23: Done                Y[ ]
*Main>
*Main> stepRun xor "1#1"
```

```
0: Read          [1]#1
1: left1          [#]1
2: xor1           #[1]
3: back           [#]F
4: back          [ ]#F
5: Read           [#]F
6: Read            [F]
7: Read            0[ ]
*Main> stepRun xor "0#1"
0: Read          [0]#1
1: left0          [#]1
2: xor0           #[1]
3: back           [#]T
4: back          [ ]#T
5: Read           [#]T
6: Read            [T]
7: Read            1[ ]
*Main> stepRun xor "10101#10011"
0: Read          [1]0101#10011
1: left1          [0]101#10011
6: xor1           0101#[1]0011
7: back           0101[#]F0011
13: Read          [0]101#F0011
14: left0          [1]01#F0011
18: xor0           101#[F]0011
19: xor0           101#F[0]011
20: back           101#[F]F011
26: Read           [1]01#FF011
32: xor1            01#FF[0]11
33: back            01#F[F]T11
39: Read            [0]1#FFT11
45: xor0             1#FFT[1]1
52: Read             [1]#FFTT1
58: xor1              #FFTT[1]
65: Read              [#]FFTTF
66: Read               [F]FTTF
67: Read               0[F]TTF
68: Read               00[T]TF
71: Read               00110[ ]
*Main>
```