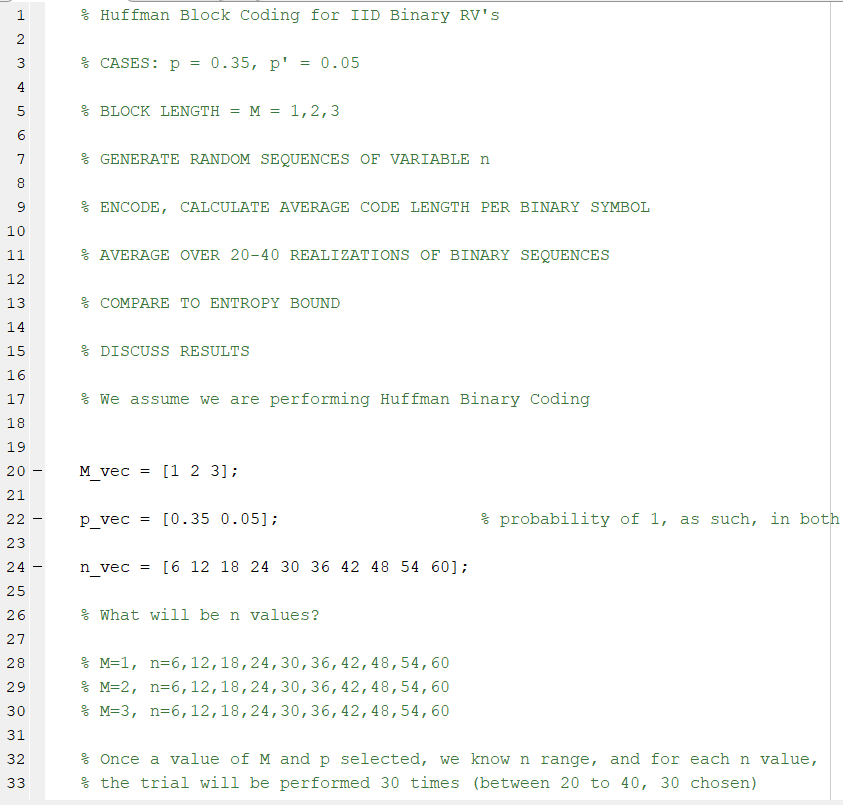
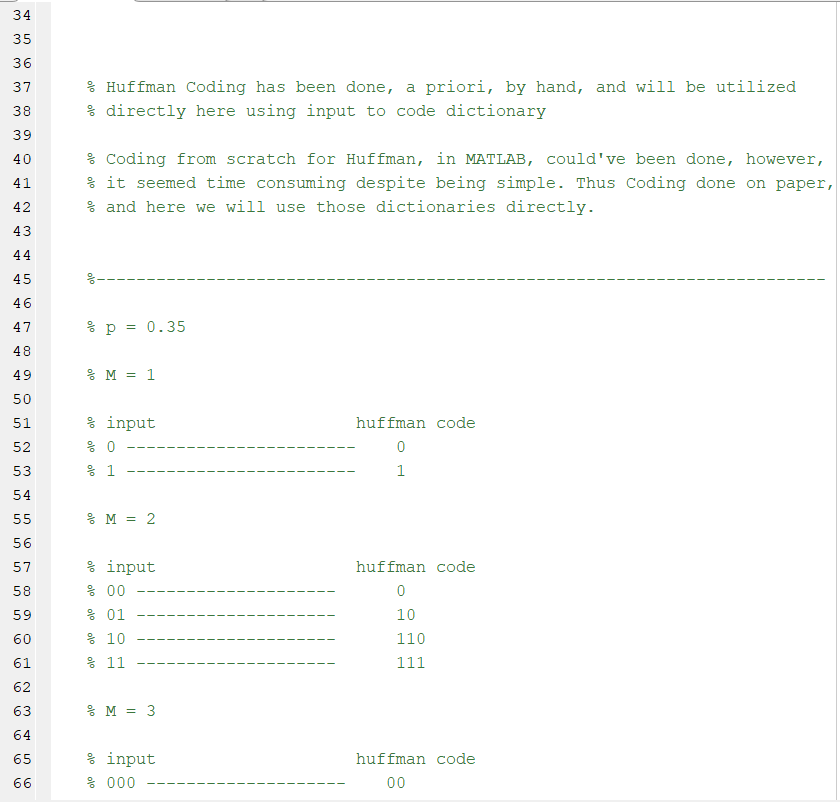
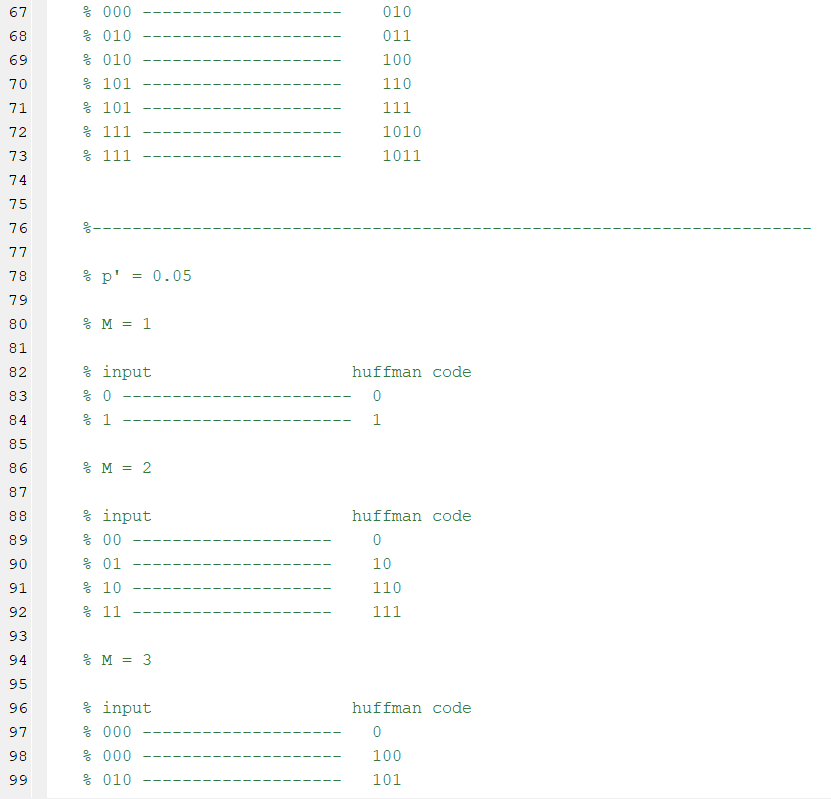
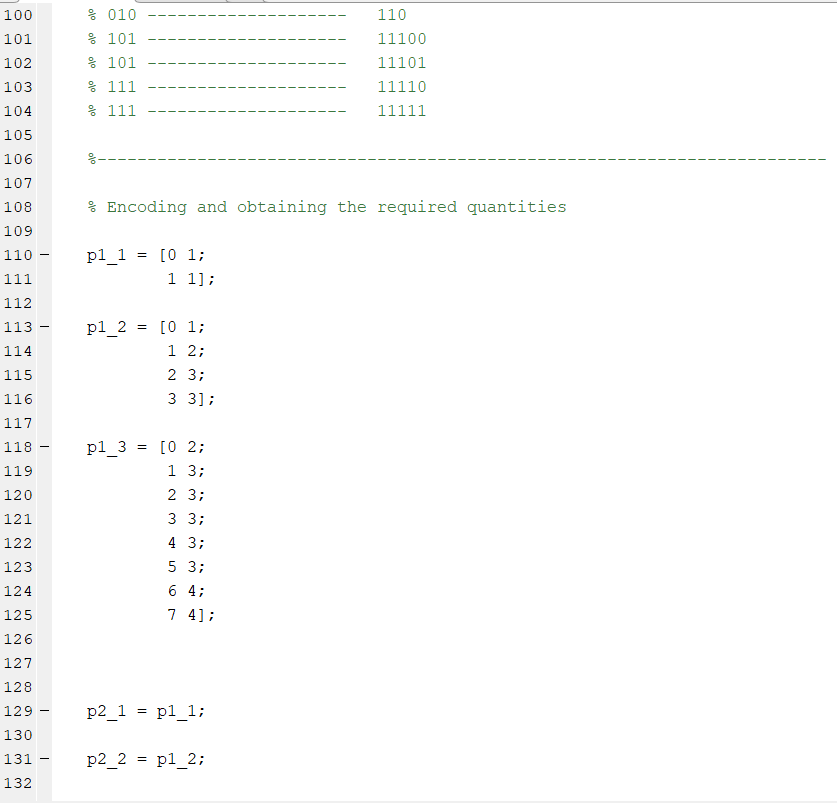
**QUESTION 6**

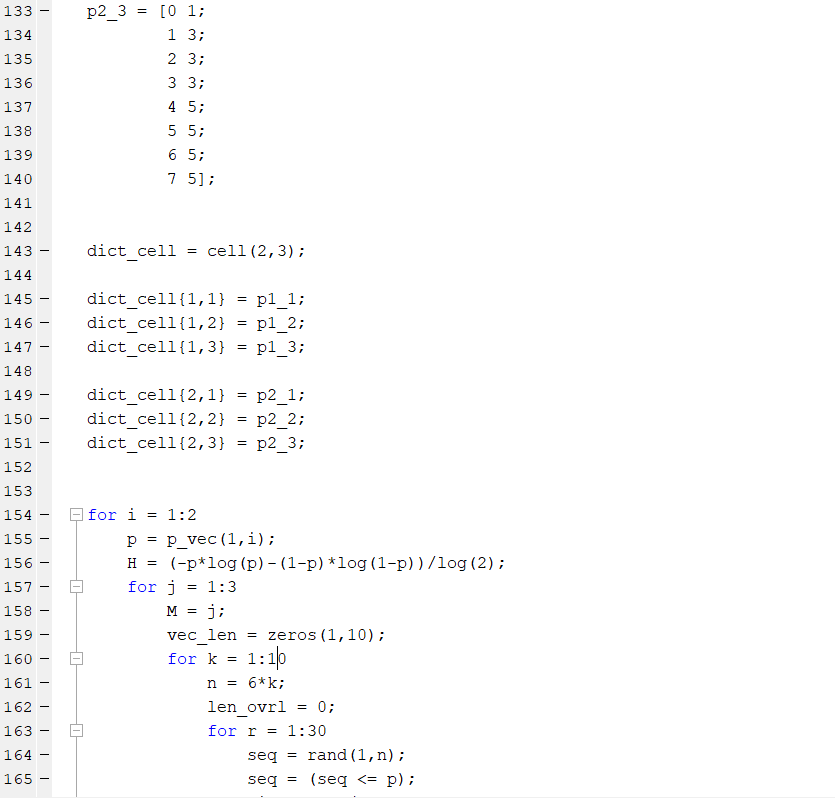
Code

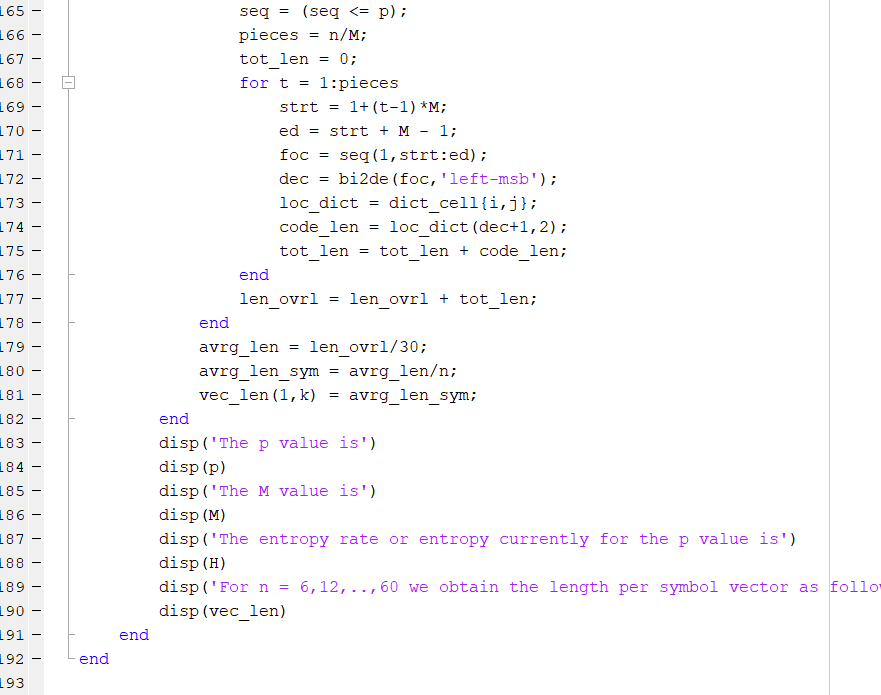












Results

>> HW8\_6

The p value is

0.350000000000000

The M value is

1

The entropy rate or entropy currently for the p value is

0.934068055375491

For n = 6,12,..,60 we obtain the length per symbol vector as follows (for each n, this is the averaged value of 30 repeats)

1 1 1 1 1 1 1 1 1 1

The p value is

0.350000000000000

The M value is

2

The entropy rate or entropy currently for the p value is

0.934068055375491

For n = 6,12,..,60 we obtain the length per symbol vector as follows (for each n, this is the averaged value of 30 repeats)

Columns 1 through 9

0.983333333333333 0.969444444444444 0.974074074074074 0.976388888888889 1.002222222222222 0.975925925925926 0.980952380952381 0.977083333333333 0.951234567901235

Column 10

0.941666666666667

The p value is

0.350000000000000

The M value is

3

The entropy rate or entropy currently for the p value is

0.934068055375491

For n = 6,12,..,60 we obtain the length per symbol vector as follows (for each n, this is the averaged value of 30 repeats)

Columns 1 through 9

0.994444444444444 0.922222222222222 0.951851851851852 0.930555555555555 0.944444444444444 0.953703703703704 0.954761904761905 0.950694444444444 0.950000000000000

Column 10

0.953333333333333

The p value is

0.050000000000000

The M value is

1

The entropy rate or entropy currently for the p value is

0.286396957115956

For n = 6,12,..,60 we obtain the length per symbol vector as follows (for each n, this is the averaged value of 30 repeats)

1 1 1 1 1 1 1 1 1 1

The p value is

0.050000000000000

The M value is

2

The entropy rate or entropy currently for the p value is

0.286396957115956

For n = 6,12,..,60 we obtain the length per symbol vector as follows (for each n, this is the averaged value of 30 repeats)

Columns 1 through 9

0.583333333333333 0.577777777777778 0.570370370370370 0.579166666666667 0.562222222222222 0.581481481481481 0.576984126984127 0.565972222222222 0.576543209876543

Column 10

0.570000000000000

The p value is

0.050000000000000

The M value is

3

The entropy rate or entropy currently for the p value is

0.286396957115956

For n = 6,12,..,60 we obtain the length per symbol vector as follows (for each n, this is the averaged value of 30 repeats)

Columns 1 through 9

0.422222222222222 0.427777777777778 0.500000000000000 0.494444444444444 0.477777777777778 0.457407407407407 0.466666666666667 0.456944444444444 0.451851851851852

Column 10

0.466666666666667

>>

**The results have been pasted for Result Tab in MATLAB due to the length.**

A few notes

p=0.35 shows a pretty good characteristic of moving towards the entropy rate as n increases

p=0.05 for the value of n we selected has the average length of codeword per symbol much farther from the entropy rate. The value of n, if we would have increased, would have shown better traits for p=0.05.

M=1 is a useless case since Huffman binary encoding is just a copy paste of the binary input.

Thus, we did, to a certain extent verify that as the value of n increases the Huffman code generates a code that has a tendency of moving towards the entropy rate bounds on average length of code per symbol.

Increase in M makes the average codeword length per symbol move closer and closer to the entropy rate. Thus, as M increases overall code optimality increases.

This again shows that to achieve true optimality via Huffman codes, the block length should be as large as possible. However, doing so is computationally expensive, which is why coding strategies such as arithmetic coding are so widely used to encode sequences of random variables.]

These are the results and the comments.