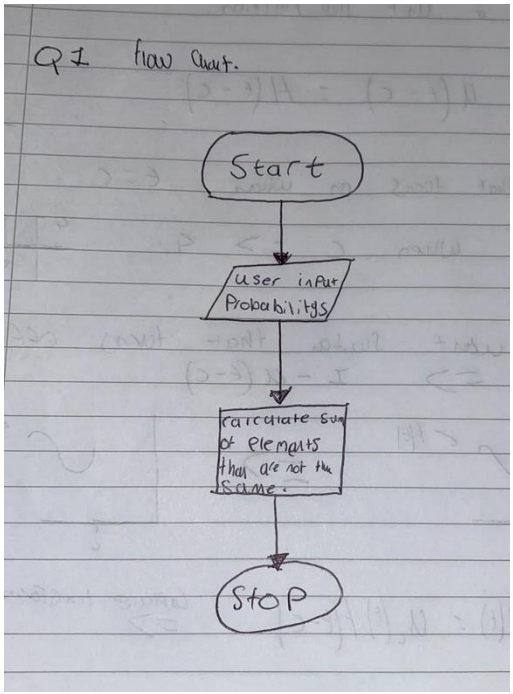
James bird
Communication systems.

Q1.
For this question I found the hamming distance by calculating the sum of the elements that are not equal, this will work for all sizes of binary numbers the user supply's if the instructions above are followed.
https://www.dropbox.com/s/6q1tckn49rvpogp/q1hamming.m?dl=0 – (.m file)



Q2.

q2.m | q3.m

```
1  clear
2  symbols = (1:6); % Alphabet vector
3  prob = [.03 .27 .15 .15 .2 .2]; % Symbol probability vector
4  [dict,avglen] = huffmandict(symbols,prob)
5  % Example can be found at: https://uk.mathworks.com/help/comm/ref/huffmandict.html
```

Command Window

New to MATLAB? See resources for Getting Started.

```
>> q2

dict =

  6×2 cell array

    {[1]}    {[0 0 0 1]}
    {[2]}    {[   0 1]}
    {[3]}    {[0 0 0 0]}
    {[4]}    {[  0 0 1]}
    {[5]}    {[   1 1]}
    {[6]}    {[   1 0]}


avglen =

    2.5100
```

James bird
Communication systems.

Q3.

For this question to calculate the binary values using the Shannon method I ask the user to input their symbol probabilities, then I have sorted the probabilities into descending order then calculated the required values for the binary code, including the number of bits, cumulative probability, and left shift values. I then display these values in a table as seen bellow.

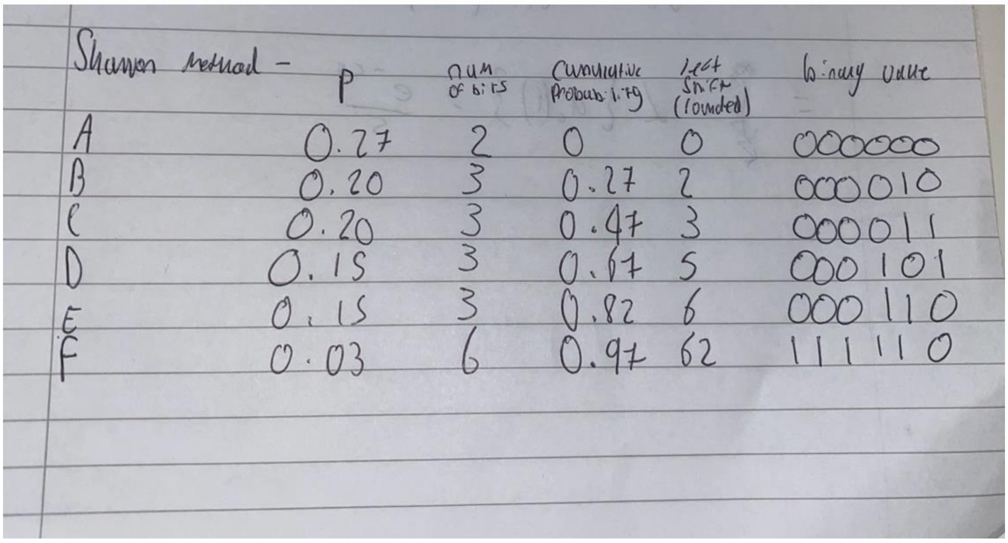
```
ans =

  6×7 table

    sym_prob    sorted_probabilities    num_bits    cumulative_probability    left_shift    left_shift_int    binary_values
    ________    ____________________    ________    ______________________    __________    ______________    _____________

    1   0.03       2      0.27             2                  0                   0               0              000000
    2   0.27       5       0.2             3               0.27                2.16               2              000010
    3   0.15       6       0.2             3               0.47                3.76               3              000011
    4   0.15       3      0.15             3               0.67                5.36               5              000101
    5    0.2       4      0.15             3               0.82                6.56               6              000110
    6    0.2       1      0.03             6               0.97               62.08              62              111110

fx >>
```



Shanon method -	P	num of bits	Cumulative Probability	left Shift (rounded)	binary value
A	0.27	2	0	0	000000
B	0.20	3	0.27	2	000010
C	0.20	3	0.47	3	000011
D	0.15	3	0.67	5	000101
E	0.15	3	0.82	6	000110
F	0.03	6	0.97	62	111110

Figure 1 shanon method by hand

https://www.dropbox.com/s/ynq4h7q9d6xmo1z/q3.m?dl=0 – (.m file)

Q5.
I have calculated different codes using the methods as each method will have different efficiency with the Huffman method being most efficient and closest to the entropy limit, whilst being easily decodable compared to my Shannon method.