1 Sub-arrays

1.1 Examining a subarray

Write a function `vectorQuery(v,n,r)` to determine whether the number `r` appears in the first `n` cells of vector `v`. The function returns 1 if `r` is in the first `n` cells of `v` and 0 otherwise. (Preconditions: `v` is a vector of numbers; `n` is a positive integer; `r` is a number.)

1.2 Returning an array

Now write a function `sequence(m)` that generates a sequence of random *integer* numbers between 1 and m, inclusive, stopping when a value is repeated for the first time. The function returns an array containing all the numbers generated (in the order in which they were generated) except for the last value that is a repeated occurrence.

Example: If the generated sequence is 3 1 9 5 7 2 5, the array to be returned should be 3 1 9 5 7 2.

Hints: The symbol for the empty array is `[]`. When "building" an array, the space or comma separator puts two items side by side—creates a row.

2 Counting frequency

Write a MATLAB script that calls the function `sequence` from the previous question 1000 times for `m=10` and keep track of the lengths of the returned arrays. Specifically, your script should keep a vector `frequency` so that `frequency(f)` stores the number of arrays of length `f` that have been returned by function `sequence`.

Thought question: What is the *expected value*, i.e., the mean, of the array length when `m` is 10? To check your (thought) answer, type up the script and draw a bar chart to show the results! It's easy to draw a bar chart in MATLAB:

```
len= length(frequency);
bar( 1:len, frequency );  % 1st argument are the "bar labels";
                          % 2nd argument are the data (the counts)
```

3 A bit of DNA analysis

The four DNA nucleotides are represented by the letters 'A', 'C', 'G', and 'T'. Write a function `findGGT(dna)` to return a vector of the locations where the substring 'GGT' occurs in `dna`, an array of characters where each character can only be 'A', 'C', 'G', or 'T'. A "location" where the substring 'GGT' occurs is the index i for which the i-th position of `dna` is 'G' and the following two vector components store the letters 'G' and 'T'.

4 Challenge Question to try at home: A bit more DNA analysis

Write a function `findPattern(dna, pat)` to return a vector of the locations where the string `pat` appears in `dna`. As before, the strings contain only the characters 'A', 'C', 'G', and 'T'. Assume that vector `dna` is longer than or equal in length to `pat`.

Please delete your files from the computer before you leave the lab.