ENGR 20 - MATLAB

Homework 3

Joseph Park

1. Gradebook Function

```
function [gradeLetter] = grade(gradeNumber)
    % Get grade letter
    if (gradeNumber >= 90)
        gradeLetter = "A";
    elseif (gradeNumber >= 80)
        gradeLetter = "B";
    elseif (gradeNumber >= 70)
        gradeLetter = "C";
    else
        gradeLetter = "D";
    end

    % Get grade plus or minus
    if (gradeNumber >= 97 || gradeNumber < 70)
        return
    else
        onesDigit = mod(gradeNumber, 10);
        if (onesDigit < 3)
            gradeLetter = append(gradeLetter, "-");
        elseif (onesDigit > 7)
            gradeLetter = append(gradeLetter, "+");
        end
    end
end

% Gradebook function test
exampleGrades = [68, 70, 72, 75, 79, 80, 81, 85, 87, 89, 90, 92, 93, 99, 100, 101];
for exampleGrade = exampleGrades
    disp(exampleGrade + " = " + grade(exampleGrade));
end
```

```
68 = D
70 = C-
72 = C-
75 = C
79 = C+
80 = B-
81 = B-
85 = B
87 = B
89 = B+
90 = A-
92 = A-
93 = A
99 = A
```

```
100 = A
101 = A
```

2. Logical Indexing in Arrays

```
% 1x100 array of random integers between 1 and 100
arr = [randi(100, [1, 100])]
```

```
arr = 1×100
    32    32    34    66    43    84    88    55    34    55    26    37    85 ⋅⋅⋅
```

```
% The find() function searches through the input vector and returns
% the indices of elements that meet the given condition.
% For example, find(arr < 5) returns the indices of elements less than 5.
find(arr < 5)
```

```
ans = 1×6
    17    40    42    48    73    92
```

```
% Instead of returning the indices of elements that meet the given
% condition, arr(*some condition*) returns the actual elements that meet
% the given condition. Running the function below with return all values in
% the array that is less than 5.
arr(arr < 5)
```

```
ans = 1×6
     2     4     3     2     1     2
```

```
% This combination of functions has the same output as arr(*some
% condition*). This is because the arr(x) function returns the element in
% the xth column. find() returns an array of indices, and since those feed
% into the arr() function, this combo with return the same thing as
% arr(*some condition*).
arr(find(arr < 5))
```

```
ans = 1×6
     2     4     3     2     1     2
```

3. Primes

```
primes = [];
for i = 1:200
    if (isprime(i))
        primes = [primes, i];
    end
end
primes
```

```
primes = 1×46
     2     3     5     7    11    13    17    19    23    29    31    37    41 ⋅⋅⋅
```

4. Factorials

```
for i = 1:10
    fprintf("%.0f! = %.0f\n", i, factorial(i));
```

```
end
```

```
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
```

5. Fibonacci Sequence

```
fib = [];
for i = 1:40
    if (i < 3)
        fib = [fib, 1];
    else
        fib = [fib, fib(i-1)+fib(i-2)];
    end
end
fib
```

```
fib = 1×40
           1           1           2           3           5           8⋯
```

6. Pythagorean Triples

```
pythag_triple_permutations = [];
pythag_triple_combinations = [];
pythag_hypothenuses = [];
for a = 1:30
    for b = 1:30
        for c = 1:30
            if (a^2 + b^2 == c^2 || a^2 + c^2 == b^2 || c^2 + b^2 == a^2)
                % Permutations
                pythag_triple_permutations = [pythag_triple_permutations; [a b c]];
                % Combinations
                if (~ismember([a^2, b^2, c^2], pythag_hypothenuses)) % Tried
~pythag_hypothenuses(pythag_hypothenuses == c^2)
                    pythag_hypothenuses = [pythag_hypothenuses; c^2];
                    pythag_triple_combinations = [pythag_triple_combinations; [a b
c]];
                end
            end
        end
    end
end

pythag_triple_permutations
```

```
pythag_triple_permutations = 66×3
```

```
     3     4     5
     3     5     4
     4     3     5
     4     5     3
     5     3     4
     5     4     3
     5    12    13
     5    13    12
     6     8    10
     6    10     8
     ⋮
```

```
pythag_triple_combinations
```

```
pythag_triple_combinations = 7×3
     3     4     5
     6     8    10
     7    24    25
     8    15    17
     9    12    15
    12    16    20
    18    24    30
```