Polytechnic Tutoring Center

Midterm 2 REVIEW – CS1133, Fall 2020

Disclaimer: This mock exam is only for practice. It was made by tutors in the Polytechnic Tutoring Center and is not representative of the actual exam given by the Academic Department.

Question 1

```
clear; clc;
isMem = input('The customer has an account on the store website: ');
timE1 = 1; timE2 = 3; timE3 = 5;
switch isMem
    case true
        % customers who are members
        isPre = input('The customer is a premium member: ');
        memTime = input('Number of years for which the customer has been a ...
           member: ');
        disp('The customer receives the following prize(s): ');
        disp('  pencil');
        if memTime>=timE1 && memTime<timE2
            disp('  pen'); % customers between 1 and 3 years
        elseif memTime>=timE2 && memTime<timE3
            disp('  keychain'); % customers between 3 and 5 years
        elseif memTime>=timE3
            disp('  teddy bear'); % customers 5 years or over
        end
        switch isPre
            case true
                % premium customers
                disp('  water bottle');
        end
    otherwise
        disp('The customer receives no prize.');
end
```

Question 2

```
clear; clc;
% you can use any matrix, including the one provided with the problem
MTX = [ -40    79    31    53     5    -9
         64    36   -72    53    78    54
        -74     2   -67    38    26   -65
          0    -3     5   -54    50   -57
        -51   -69   -63    28    -5   -51];
[nRows, mCols] = size(MTX); % size is only vectorized operation allowed
evenCount = 0; % initialize count of even numbers to zero
for i = 1:nRows
    for j = 1:mCols
        % one of many ways to see if a value is even:
        isEven = ceil(MTX(i,j)/2) == MTX(i,j)/2;
```

```
        if isEven
            evenCount = evenCount+1; % increase count of even numbers
            MTX(i,j) = -MTX(i,j); % change the value
        else
            % if statement to see if the odd number is positive
            if MTX(i,j)>0
                MTX(i,j) = 3*sqrt(MTX(i,j)); % change positive odd numbers
            end
        end
    end
end
toT = nRows*mCols; % total number of elements in the matrix
oddCount = toT-evenCount; % # of odds is total number minus # of evens
```

Question 3

```
clear; clc;
nTrials = 1e7; % some large number of trials for the Monte Carlo simulation
% minimum and maximum amount of candy a single niece or nephew will have
leastCandy = 12; mostCandy = 30;
nNib = 17; % number of nieces and nephews
maxCount = 350; % number of candies at which Evan will stop counting
soLittleCandy = 300; % number of candies for which Evan buys them more candy
% Initialize variables for the number of times Evan counts all the nieces/
% nephew's candy without stopping, and the number of times Evan gives his
% nieces and nephews extra candy.
countTilEnd = 0;
giveThemCandy = 0;
% each time this for loop runs represents a single trial
for i = 1:nTrials
    % initialize the total number of candies for a single trial to zero
    nCandies = 0;
    % initialize the count of the number of times each loop runs to 1
    % each time this while loop runs represents Evan counting how much
    % candy a single child gets
    nChild = 1;
    while (nChild<=nNib) && (nCandies<=maxCount)
        % randomly generate the number of candies a single child collected
        nThisKid = randi([leastCandy mostCandy]);
        % add the number of pieces of candy
        % this niece/nephew to the total for all the kids that have been
        % counted so far
        nCandies = nCandies+nThisKid;
        % increase the counter by one to move on to the loop for the next child
        nChild = nChild+1;
    end
    if nCandies>maxCount
        % when Evan does not count all the pieces of candy
        countTilEnd = countTilEnd+1;
    elseif nCandies<=soLittleCandy
        % when the total number of pieces of candy is small, so Evan buys
        % them extra candy
        giveThemCandy = giveThemCandy+1;
    end
end
```

```
ratiO1 = countTilEnd/nTrials; % fraction of times Evan counts to the end
perCent1 = ratiO1*100; % make the ratio a percentage
ratiO2 = giveThemCandy/nTrials; % fraction of times Evan counts to the end
perCent2 = ratiO2*100; % make the ratio a percentage
disp(['Evan will finish counting all of his nieces'' and nephews'' . . .
   candy ' num2str(perCent1) '% of the time.']);
disp(['Evan will buy his nieces and nephews candy ' . . .
   num2str(perCent2) '% of the time.']);
```