Polytechnic Tutoring Center

Midterm 1 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

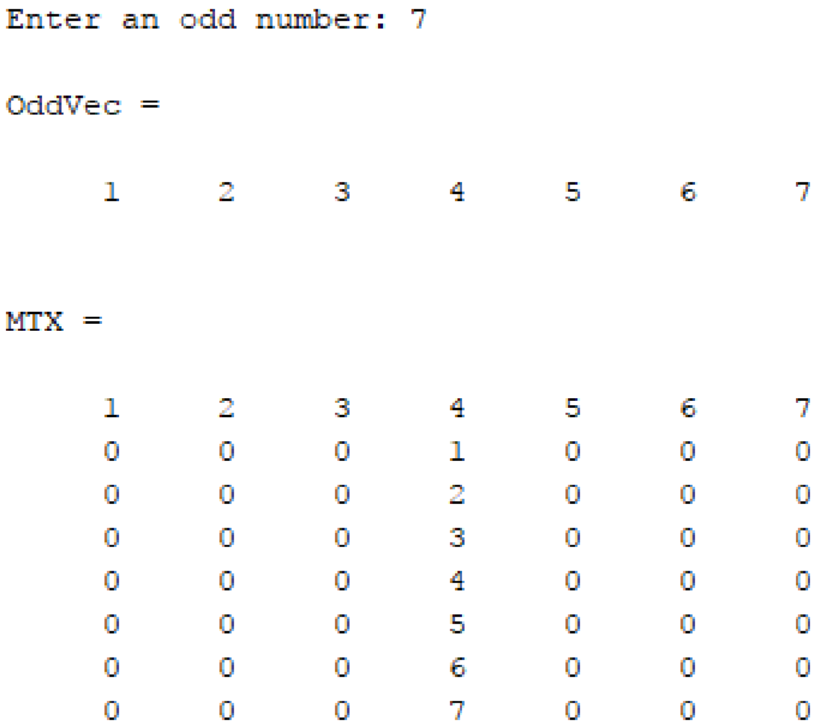
Jerry's family just moved to a new town. It's the middle of summer, and Jerry wants to take a walk in the park with his kids. Jerry uses the internet to find information about all the parks in town, including how far each park is from his house, how long it takes to circle around each park one time, and whether each park has a concession stand.

Jerry wants to go to a park that is close enough to his house so he and his kids can get there by walking. However, he wants the park to be far enough from the house that the kids will have the chance to see a new part of town. He decides that a reasonable distance is between 100 and 800 meters (excluding the values 100 and 800 meters). Jerry also wants to make a note of the parks he considers to be too far away from his house. Jerry wants to walk around the park 3 full times. He does not want to spend more than 2 hours at the park so that the kids will not get bored. He wants the park to have a concession stand so that he can bribe the kids with snacks if he needs to.

Write a program which will select the parks that fit all of Jerry's specifications. All the information Jerry found is contained in a matrix, which is given to you (you do not have to write or create this matrix). The distance from Jerry's house, which is in the first column of the matrix, is given in meters. The time it takes to walk around each park is in the second column, and is given in minutes. The third column of the matrix contains a logical value reading true if the park has a concession stand and false if the park does not have a concession stand. Your program should output the row indices of all the applicable parks, as well as the row indices of the parks that are too far from the house (in two separate vectors). There is no need to display.

Question 2

Write a program that turns a vector of sequential odd numbers into a T-matrix. That is to say that the top row of the matrix is the initial vector and the middle column of the matrix consists of the middle value of the vector, followed by the vector again. Then the remaining terms in the matrix are all zeros. Your code should allow the user to input the length of the odd vector during run-time. Below is a sample output.



```
Enter an odd number: 7

OddVec =

     1     2     3     4     5     6     7


MTX =

     1     2     3     4     5     6     7
     0     0     0     1     0     0     0
     0     0     0     2     0     0     0
     0     0     0     3     0     0     0
     0     0     0     4     0     0     0
     0     0     0     5     0     0     0
     0     0     0     6     0     0     0
     0     0     0     7     0     0     0
```

Question 3

Maxine needs a break from the mental strain she experiences in engineering school, so she decides to perform a mindless task. She rolls a standard 6-sided die and flips two coins at the same time. Run a Monte Carlo simulation of the situation to determine the probability that both coins land on heads and the die lands on either the number 2 or 5 in a single trial.

Extra Practice

Given an *m* by *n* matrix of any size, MTX0, where *n* may be even or odd, create two *m* by *n* matrices, MTX1 and MTX2, so that

1. The left-hand half of MTX1 looks exactly like the left-hand half of MTX0, and the right-hand half of MTX1 consists of the same columns as its left-hand half but ordered in the opposite direction.
2. The right-hand half of MTX2 looks exactly like the right-hand half of MTX0, and the left-hand half of MTX2 consists of the same columns as its right-hand half but ordered in the opposite direction.

There is no need to generate MTX0, but for testing purposes you can use the following code to generate MTX0:

```
nRows = 3; nCols = 5;
```

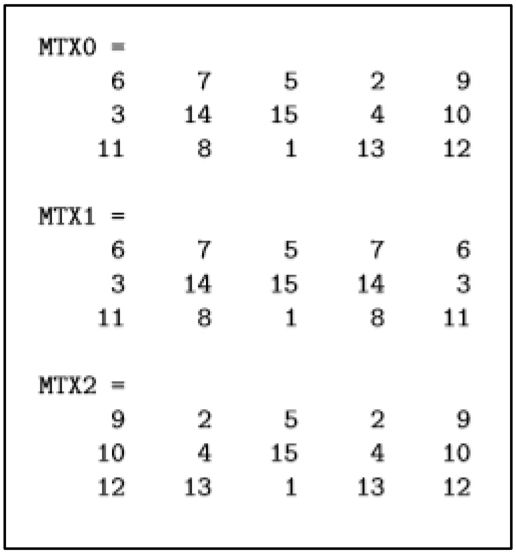
```
MTX0 = zeros(nRows,nCols);
MTX0(:) = randperm(nRows*nCols)’;
```

The function randperm(k) gives a row vector whose elements are a random permutation of the positive integers 1,2,…(*k*−1),*k* if *k* is a positive integer.

You may need to use “rounding” in order that your program will work for even or odd *n*.

Here are two examples, one when *n* is odd, and the other for *n* even.

For an odd number of columns see Figure 1. For an even number of columns see Figure 2.

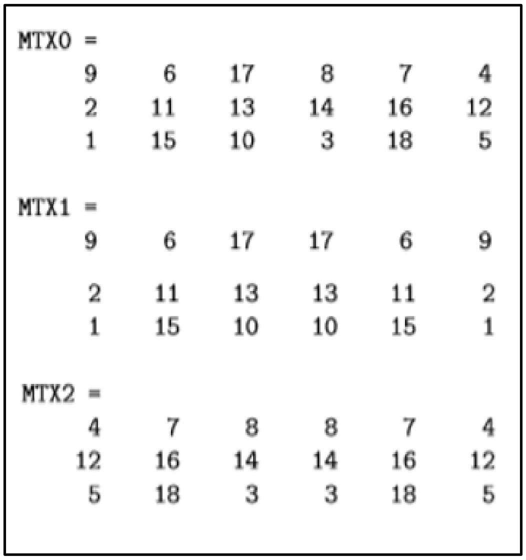


```
MTX0 =
     6     7     5     2     9
     3    14    15     4    10
    11     8     1    13    12

MTX1 =
     6     7     5     7     6
     3    14    15    14     3
    11     8     1     8    11

MTX2 =
     9     2     5     2     9
    10     4    15     4    10
    12    13     1    13    12
```

Figure 1: Odd Number of Columns



```
MTX0 =
     9     6    17     8     7     4
     2    11    13    14    16    12
     1    15    10     3    18     5

MTX1 =
     9     6    17    17     6     9

     2    11    13    13    11     2
     1    15    10    10    15     1

MTX2 =
     4     7     8     8     7     4
    12    16    14    14    16    12
     5    18     3     3    18     5
```

Figure 2: Even Number of Columns