General Instructions:

- **Lab is due online by 11:59 pm of the due date.** The assignment must be typed, not handwritten or scanned.

- Label your Python programs \texttt{q<num>.py}, where \texttt{num} is the question number. For example, your solution to the first question is stored in the file \texttt{q1.py}.

- Make sure you read the lab before you start. Your programs must match the output exactly as given for each question. This is important as one of the keys to being a good programmer is attention to details.

- If you forgot to bring a computer, you can check out a temporary laptop from ZACH 383 for a few hours (depending on the availability).

- Grading is based on correctness and clarity. **Copying work from another source and submitting it as your own is plagiarism.** The minimum penalty for plagiarism is a grade of zero.
Lab Questions

1. **Rock-Paper-Scissors.** Write a Python program stored in a file q1.py to play Rock-Paper-Scissors. In this game, two players count aloud to three, swinging their hand in a fist each time. When both players say three, the players throw one of three gestures:

   a) Rock: a clenched fist,

   b) Scissors: index and middle fingers extended and separated, and

   c) Paper: an open hand.

Each gesture beats one of the others: rock crushes scissors, scissors cuts paper, and paper covers rock. If both players pick the same gesture it is a draw and they must throw again.

Your task is to have a user play Rock-Paper-Scissors against a computer opponent that randomly picks a throw. You will ask the user how many points are required to win the game. The Rock-Paper-Scissors game is composed of rounds, where the winner of a round scores a single point. The user and computer play the game until the desired number of points to win the game is reached. Within a round, if there is a tie (i.e., the user picks the same throw as the computer), prompt the user to throw again and generate a new throw for the computer. The computer and user continue throwing until there is a winner for the round.

**Programming notes.** The starter code q1.py is provided to you [HERE](#). The code for the main() function is provided. Your goal is to complete the code for the rock_paper_scissors() function. If you need more user-defined functions than what is provided, then that’s fine. But, your program must have at a minimum the user-defined functions main() and rock_paper_scissors().

**One final note.** The two examples below show program execution for when one or two points are required to win the game. However, your program should work for any positive point value greater than zero (e.g., 3, 5, 10) provided by the user.
Example #1. Boxes enclose user input on lines 7 and 11. Lines 1–5 are from the code that has been provided for you in the main() function. At line 7, the user types 1 when prompted for the number points required to win the game. Next, the program shows the round number, prompts for the user’s throw, shows the computer’s throw, and outputs the game score (lines 9–15). Since only one point is required to when the game, the game is over after one round of play (line 17). The computer is declared the winner (line 19).

ROCK PAPER SCISSORS in Python

Rules: 1) Rock wins over Scissors.
       2) Scissors wins over Paper.
       3) Paper wins over Rock.

How many points does it take to win? 1

************************** ROUND #1 **************************

Pick your throw: [r]ock, [p]aper, or [s]cissors? s
Computer threw rock, you lose!

Your score: 0
Computer’s score: 1

************************** GAME OVER **************************

Computer wins!

Example #2. Boxes enclose user input on lines 7, 11, 19, 27, 30, and 33. Execution follows similarly to Example #1. Here, two points are needed to win the game (line 7). The user wins the first round (lines 9–15), and the computer wins the second round (lines 17–23). In the third round, there were several ties before the user finally wins the round (lines 25–37). Since the user has accumulated two points, the game is over (line 39). The user is declared the winner (line 41).

ROCK PAPER SCISSORS in Python

Rules: 1) Rock wins over Scissors.
       2) Scissors wins over Paper.
       3) Paper wins over Rock.

How many points does it take to win? 2

************************** ROUND #1 **************************
Pick your throw: [r]ock, [p]aper, or [s]cissors? 
Computer threw scissors, you win!

Your score: 1
Computer’s score: 0

********************* ROUND #2 *********************

Pick your throw: [r]ock, [p]aper, or [s]cissors? 
Computer threw paper, you lose!

Your score: 1
Computer’s score: 1

********************* ROUND #3 *********************

Pick your throw: [r]ock, [p]aper, or [s]cissors? 
Tie!

Pick your throw: [r]ock, [p]aper, or [s]cissors? 
Tie!

Pick your throw: [r]ock, [p]aper, or [s]cissors? 
Computer threw scissors, you win!

Your score: 2
Computer’s score: 1

********************* GAME OVER *********************

You win!

2. Extra credit (Matrix multiplication): Redo Problem 3 of Lab 4 using functions. Your program should contain at least two functions. One function matrix_multiplier that calculates the product of two matrices, and another function matrix_transpose that calculates the transpose of a matrix.

   a) matrix_multiplier takes as input matrix A, matrix B, number of rows of A, number of columns of A, number of rows of B, number of columns of B.

   b) matrix_transpose takes as input matrix C.
Submitting Your Assignment

Once you have completed your programs, submit each of them (q1.py, extra_credit.py) electronically. The extra credit question is optional. You may resubmit your files as many times as you need until the due date. Only the most recent submission is graded.

You are required to include the following lines in the header of all your files:

```python
# File: filename.py
# Author: Student name
# Date: xx/xx/2019
# Section: Student section number
# E-mail: student_email@tamu.edu
# Description:
# e.g. This program asks for ...
```

- If you are in section 501 or 502, submit your files only on: mimir.io

- If you are in section 503, 504, 505 or 506, submit your files only on: gradescope.com