

Assignment # 1

Course Instructor: Abdul Rehman

Course TA: Saif Ullah

Please note that you must implement the key python functions and scripts for the given problem statements on your own. You have chosen this course on your own; if you copy from the internet or others, you will not get the learning experience intended through this assignment and would be a complete waste of time for us all. If you need help in understanding specific concepts, do ask, we will try to clarify the concepts. The choice is yours. Keep on grinding because it's worth it. We are lucky because we work hard, the more we work hard, the more we get lucky. Happy Learning

Write all these tasks in a Jupyter Notebook and save it with the name "assignment_1_your_name.ipynb"

Python Programming Portion

Question 1: Write a program which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 3200 (both included). The numbers obtained should be printed in a comma-separated sequence on a single line.

Hints: Consider using range(begin, end) method.

Question 2: With a given integral number n, write a program to generate a dictionary that contains (i, i*i) such that i is an integral number between 1 and n (both included). and then the program should print the dictionary. Suppose the following input is supplied to the program: 8 Then, the output should be: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}

Hints: In case of input data being supplied to the question, it should be assumed to be a console input. Consider using dict().

Question 3: Write a program that calculates and prints the value according to the given formula: $Q = \text{Square root of } [(2 * C * D)/H]$. Following are the fixed values of C and H: C is 50. H is 30. D is the variable whose values should be input to your program in a comma-separated sequence. Example Let us assume the following comma separated input sequence is given to the program: 100,150,180. The output of the program should be: 18,22,24.

Hints: If the output received is in decimal form, it should be rounded off to its nearest value (for example, if the output received is 26.0, it should be printed as 26) In case of input data being supplied to the question, it should be assumed to be a console input.

Question 4: Write a program which takes 2 digits, X, Y as input and generates a 2-dimensional array. The element value in the i-th row and j-th column of the array should be $i*j$. Note: $i=0, 1, \dots, X-1$; $j=0, 1, \dots, Y-1$. Example Suppose the following inputs are given to the program: 3,5 Then, the output of the program should be: `[[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8]]`.

Hints: In case of input data being supplied to the question, it should be assumed to be a console input in a comma-separated form.

Question 5: Write a program that accepts a comma separated sequence of words as input and prints the words in a comma-separated sequence after sorting them alphabetically. Suppose the following input is supplied to the program: *without, hello, bag, world*. Then, the output should be: *bag, hello, without, world*.

Hints: In case of input data being supplied to the question, it should be assumed to be a console input.

Question 6: Write a program that accepts a sequence of whitespace separated words as input and prints the words after removing all duplicate words and sorting them alphanumerically. Suppose the following input is supplied to the program: *"hello world and practice makes perfect and hello world again"*. Then, the output should be: *"again and hello makes perfect practice world"*.

Hints: In case of input data being supplied to the question, it should be assumed to be a console input. We use set container to remove duplicated data automatically and then use `sorted()` to sort the data.

Question 7: Write a program that accepts a sentence and calculates the number of letters and digits. Suppose the following input is supplied to the program: *"hello world! 123"* Then, the output should be: `LETTERS 10 DIGITS 3`. Store the output in dictionary. Keys are *"LETTERS"* and *"DIGITS"* and values are the counts of each.

Hints: In case of input data being supplied to the question, it should be assumed to be a console input.

Question 8: Use a list comprehension to square each odd number in a list. The list is input by a sequence of comma-separated numbers. Suppose the following input is supplied to the program: 1,2,3,4,5,6,7,8,9 Then, the output should be: 1,9,25,49,81...

Hints: In case of input data being supplied to the question, it should be assumed to be a console input.

Question 9: Write a program that computes the net amount of a bank account based on a transaction log from console input. The transaction log format is shown as follows: D 100 W 200.

D means deposit while W means withdrawal. Suppose the following input is supplied to the program: D 300 D 300 W 200 D 100 Then, the output should be: 500.

Hints: In case of input data being supplied to the question, it should be assumed to be a console input.

Question 10: With two given lists [1,3,6,78,35,55] and [12,24,35,24,88,120,155], write a program to make a list whose elements are intersection of the above given lists.

Hints: Use set () and "&=" to do set intersection operation.

Question 11: Write a program to reverse any integer without changing the datatype of the given integer in the process. All the operation must be done keeping the number as integer.

Hints: Use multiplication, division and modulus operation.

Question 12: Write a program which prints all possible permutations and combinations of [1,2,3,4,5,6].

Hints: Use `itertools.permutations()`, `itertools.combinations()` to get permutations and combinations of list.

NumPy Portion

Question 1: Create a null vector of size 10 but the fifth value which is 1.

Question 2: Make a vector ranging from 1 to 50 in NumPy and reverse it. (First element becomes last).

Question 3: Create a 3x3 matrix with values ranging from 0 to 8.

Question 4: Find indices of non-zero elements from [1,2,0,0,0,4,0] by first converting it into NumPy array and then do the operation.

Question 5: Create a 10x10 array with random values and find the minimum and maximum values.

Question 6: Make a NumPy array of 5x5 size having only ones and add a border (filled with 0's) around that array.

Question 7: Create two matrix of size 5x3 and 3x2 having random values and do the matrix multiplication (real matrix product). Also check if we can multiply matrices of size 3x5 and 3x2? If the answer is no, then why?

Question 8: Create a vector having values from 1 to 13 and negate (multiply by -1) all elements which are between 3 and 8, in place.

Question 9: Create an 8x8 matrix and fill it with a checkerboard pattern.

Question 10: Make a 5x5 matrix of random values and normalize it.

Image Processing Portion

Task 1: Load the house.jpg file for this task. Use the Gaussian blur function to blur at least one of the windows. Use 13x13 size kernel for the blur.

Additionally, you will need to apply Canny edge detection on the original house image. You must apply 2 of these edge detections; one on the original picture and the second on a slightly blurred picture. Use same threshold values on both edge detections. Show all the results in Jupyter notebook using matplotlib.pyplot function.

Task 2: Load the square1 and square2 files provided in the ***image_processing_files*** folder for this task. Use the bitwise operations on these images to make an octagon as well as an 8-pointed star. (Think of the black pixels as 0 and the white pixels as 1 in the binary operations). Bitwise operations are AND operation, OR operation, NOT, XOR etc. Explore more of these operations by reading OpenCV documentations on the internet.

Task 3: Load the shapes.bmp image and change its color space to HSV (Hue, Saturation, Value). Then, use the inRange function to isolate each shape separately:

`cv2.inRange(hsv,np.array([hmin,smin,vmin]),np.array([hmax,smax,vmax]))`

You must manually determine the values for the above 6 parameters. H values go from 0 to 179. S and V values go from 0 to 255. Display all four results (black and white images) in four separate cells in Jupyter Notebook.

Hint: Experiment with V values first to remove the background white color.

Task 4: In the previous task, you acquired binary "masks" for each shape to separate it from other shapes. In this task, you will determine the center of each shape. Using the code from the previous task, place a black circle at the center point of all shapes in the shapes.jpg image. Your plotted image must show a single image (shapes.jpg) with the center points marked on all the provided shapes.

To find the center point (cx, cy) in an image (img), use the moments function as shown in the code example:

```
M = cv2.moments (img)
if M['m00'] > 0:
    cx = int(M['m10']/M['m00'])
    cy = int(M['m01']/M['m00'])
```

Task 5: Load the persp.jpg file for this task. Apply perspective transformation by using the four corners in the quadrilateral and map them to the four outer corners of the image file. The result should be a flat, rectangular image (the width/height ratio in the final image is 3/4). Try different points based on hit and trials.

Hints: click [HERE](#) for help

Task 6: Load the robot_green_bg file for this task. Using your knowledge from task 3, replace the green color of the background with blue by scanning through each pixel. You can use RGB or HSV color space for this task. The following syntax examples will help:

- To check a pixel color in blue channel:

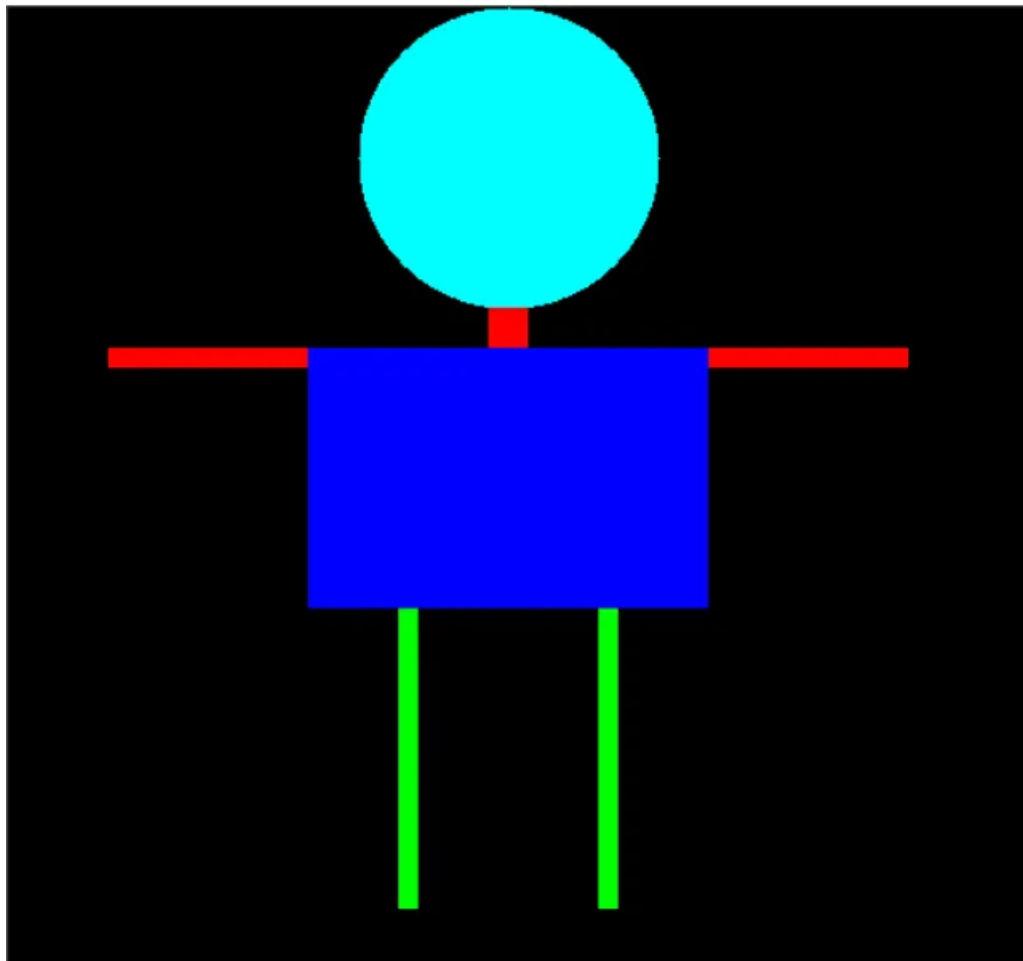
```
if img[i,j,0] < 100
```

- To set a pixel color as black:

`img[i,j,:] = (0,0,0)`

Plot the image with the modified background. Next, load the road image. Modify the code so that the robot is placed on the road. You must go through each pixel in the image. You can place the robot anywhere, but the entire robot must be visible. Plot the robot on the road image.

Task 7: Draw the following image using OpenCV Python built-in shapes:



Bonus Questions

Bonus Questions are not mandatory. If you want to show up or boast about your programming skills, do give it a try on your own. Let's see who can do these tasks xD.

Question 1: A website requires the users to input username and password to register. Write a program to check the validity of password input by users. Following are the criteria for checking the password:

1. At least 1 letter between [a-z]
2. At least 1 number between [0-9]
3. At least 1 letter between [A-Z]
4. At least 1 character from [\$#@]
5. Minimum length of transaction password: 6
6. Maximum length of transaction password: 12 Your program should accept a sequence of comma separated passwords and will check them according to the above criteria. Passwords that match the criteria are to be printed, each separated by a comma. Example If the following passwords are given as input to the program: "ABd1234@1", "aF1#,2w3E*", "2We3345" Then, the output of the program should be: "ABd1234@1"

Hints: In case of input data being supplied to the question, it should be assumed to be a console input.

Question 2: You are given an integer; your task is to print an alphabet rangoli of size. (Rangoli is a form of Indian folk art based on creation of patterns.)
Different sizes of alphabet rangoli are shown below:

#size 3

```
----c----  
--c-b-c--  
c-b-a-b-c  
--c-b-c--  
----c----
```

#size 5

```
-----e-----  
-----e-d-e-----  
----e-d-c-d-e----  
--e-d-c-b-c-d-e--  
e-d-c-b-a-b-c-d-e  
--e-d-c-b-c-d-e--  
----e-d-c-d-e----  
-----e-d-e-----  
-----e-----
```

#size 10

```
-----j-----
-----j-i-j-----
-----j-i-h-i-j-----
-----j-i-h-g-h-i-j-----
-----j-i-h-g-f-g-h-i-j-----
-----j-i-h-g-f-e-f-g-h-i-j-----
-----j-i-h-g-f-e-d-e-f-g-h-i-j-----
----j-i-h-g-f-e-d-c-d-e-f-g-h-i-j----
--j-i-h-g-f-e-d-c-b-c-d-e-f-g-h-i-j--
j-i-h-g-f-e-d-c-b-a-b-c-d-e-f-g-h-i-j
--j-i-h-g-f-e-d-c-b-c-d-e-f-g-h-i-j--
----j-i-h-g-f-e-d-c-d-e-f-g-h-i-j----
-----j-i-h-g-f-e-d-e-f-g-h-i-j-----
-----j-i-h-g-f-e-f-g-h-i-j-----
-----j-i-h-g-f-g-h-i-j-----
-----j-i-h-g-h-i-j-----
-----j-i-h-i-j-----
-----j-i-j-----
-----j-----
```

The center of the rangoli has the first alphabet letter a , and the boundary has the alphabet letter (in alphabetical order).

Input Format

Only one line of input containing the size of the rangoli.

Sample Input

5

Sample Output

-----e-----
-----e-d-e-----
----e-d-c-d-e----
--e-d-c-b-c-d-e--
e-d-c-b-a-b-c-d-e
--e-d-c-b-c-d-e--
----e-d-c-d-e----
-----e-d-e-----
-----e-----

