

1. Generate a random integer. Depending on whether the number is even or odd, print out an appropriate message on the screen. To generate a random integer, use `random.randint` or `numpy.random.randint`. *Hint: how does an even / odd number react differently when divided by 2?*

Extras:

If the number is a multiple of 4, print out a different message.

Write the script into a function.

2. Take a list, say for example this one:

```
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
```

and write a program that prints out all the elements of the list that are less than 5.

Extras:

Instead of printing the elements one by one, make a new list that has all the elements less than 5 from this list in it and print out this new list.

Write it into a function.

3. Write a Python program which accepts the radius of a circle from the user and compute the area. Use `input()` to get user inputs.

Sample Output :

```
r = 1.1
```

```
Area = 3.8013271108436504
```

After finish, try to write it into a function.

4. Write a Python script to generate and print a dictionary that contains a number (between 1 and n) in the form (x, x*x).

Sample Dictionary (n = 5) :

Expected Output : {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

After finish, try to write it into a function.

5. Take two lists, say for example these two:

```
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
```

```
b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
```

and write a program that returns a list that contains only the elements that are common between the lists (without duplicates). Make sure your program works on two lists of different sizes. After finish, randomly generate two lists to test this (use `random.sample` or `numpy.random.randint`). Also, write the script into a function.

6. In English there are 5 vowels, i.e. 'a', 'e', 'i', 'o', 'u'. Write a Python program to get all non-repeating 3-vowel sequences. Each vowel can only appear at most once. Write a function to perform similar action, when given a list and an integer k, generate all possible k-length combination of the elements without repetition from the list.

7. Write a Python program to print all even numbers from a given numbers list in the same order and stop the printing if any numbers that come after 237 in the sequence. You may want to use the function you wrote for exercise 1.

Sample list:

```
numbers = [  
    386, 462, 47, 418, 907, 344, 236, 375, 823, 566, 597, 978, 328, 615, 953, 345,  
    399, 162, 758, 219, 918, 237, 412, 566, 826, 248, 866, 950, 626, 949, 687, 217,  
    815, 67, 104, 58, 512, 24, 892, 894, 767, 553, 81, 379, 843, 831, 445, 742, 717,  
    958, 743, 527  
]
```

8. (Hard) Write a Python program to get all strobogrammatic numbers that are of length n.

A strobogrammatic number is a number whose numeral is rotationally symmetric, so that it appears the same when rotated 180 degrees. In other words, the numeral looks the same right-side up and upside down (e.g., 69, 96, 1001).

For example,

Given $n = 2$, return ["11", "69", "88", "96"].

Given $n = 3$, return ['818', '111', '916', '619', '808', '101', '906', '609', '888', '181', '986', '689']

9. Mini project: Rock-Paper-Scissor game, part 2

Write a Rock-Paper-Scissor game program, in which the player plays against the computer.

Below is some starter code to help you get started:

```
import random # the computer will use random to decide how it plays
```

```
print('Welcome to the Rock-Paper_Scissor game!')
```

```
# maybe ask the player if he/she wants to read the rule, and if so, print it out?
```

```
# game starts
```

```
# possible choices
```

```
choices = ['Rock', 'Paper', 'Scissor']
```

```
# get the player's choice; make sure you get meaningful result
```

```
player_choice = input('Rock (r), Paper (p)' or Scissor (s)?')
```

```
# generate computer's choice
```

```
computer_choice = random.choice(choices)
```

```
# compare player's choice to computer's choice to see the result. Probably a function to do this?
```

After you finish, do the following:

- 1) After any game, ask if the player wants to continue or not, and proceed as the player chose.
- 2) Record the player's choice, the computer's choice and the result for each game

3) Calculate the probability of each choice for the player and the computer. Also calculate the winning probability for the player.

4) Write these parameters into a file.

Extras:

5) In the function `random.choice`, each option is chosen with equal probability. Can you write a function in which you can assign different probability for each choice?

6) Now with the function written in 5), you can change the behavior of the computer by modifying the choice probabilities. Write a new version of the game in which you manipulate the choice probabilities as the game progress, and ask your friend(s) to play it. Record the results and check - how much change in the choice probabilities can be noticed by the player(s)? How fast can the player(s) notice that the computer's behavior has changed?