**Exercises 2**

1. 5 pts. -> Write a function that takes a list and returns a new list that contains all the elements of the first list minus all the duplicates.
2. 3 pts. -> Write a function to remove characters from a string starting from zero up to n and return a new string. For example:

remove\_chars("pynative", 4) so output must be tive. Here, we need to remove the first four characters from a string.

1. ~~5 pts. -> Write a program (using functions!) that asks the user for a long string containing multiple words. Print back to the user the same string, except with the words in backwards order. For example, say I type the string:~~

*~~My name is Emir~~*

~~Then I would see the string:~~

*~~Emir is name My~~*

1. 5 pts. -> Write a function that will take a String and count the number of times that the letters **a**, **e**, **i**, **o** and **u** appear in the string. ***(You are not allowed to create a variable for each of the letters and count like that***)
2. 8 pts -> Create a program that will play the “cows and bulls” game with the user. The game works like this:

Randomly generate a 4-digit number. Ask the user to guess a 4-digit number. For every digit that the user guessed correctly *in the correct place*, they have a “cow”. For every digit the user guessed correctly *in the wrong place* is a “bull.” Every time the user makes a guess, tell them how many “cows” and “bulls” they have. Once the user guesses the correct number, the game is over. Keep track of the number of guesses the user makes throughout teh game and tell the user at the end.

1. 8 pts. -> Write a password generator in Python. Be creative with how you generate passwords - strong passwords have a mix of lowercase letters, uppercase letters, numbers, and symbols. The passwords should be random, generating a new password every time the user asks for a new password.
2. 8 pts -> Write a python program that guesses a number between 1 and 100,000 that you thought of. You think of a number and the program guesses. Each time the program guesses, you tell it whether your number is higher or lower than the guess. This keeps happening until the program guesses your number
3. 15 pts -> Write a Python program to calculate the number of days between two given dates.(pretend leap years don’t exist, extra week of homework if you count leap years as well)
4. 5 pts. -> Two words are anagrams, if they are made from exactly the same letters. E,g, **dusty & study**. Write a function that takes two words and prints whether they are anagrams or not.
5. 8 pts -> Write a function ***drawFace(center, size, win)*** *(center is a Point, size is an int, and win is a GraphWin).* Your function draws a smiley face of specified size and location of center in the given window. Then write a main function that lets the user click in the window, and have a smileyface drawn in the place where they clicked.
6. 5 tps. -> A year is a leap year if it is divisible by 4, unless it is a century year that is not divisible by 400. (1800 and 1900 are not leap years while 1600 and 2000 are.) Write a program that calculates whether a year is a leap year.
7. 8 pts -> Write a program that draws an archery target (see Programming Exercise 2 from Chapter 4) and allows the user to click five times to represent arrows shot at the target. Using five-band scoring, a bulls-eye (yellow) is worth 9 points and each successive ring is worth 2 fewer points down to 1 for white. The program should output a score for each click and keep track of a running sum for the entire series.
8. 8 pts -> The Fibonacci sequence starts 1, 1, 2, 3, 5, 8, . . .. Each number in the sequence (after the first two) is the sum of the previous two. Write a program that computes and outputs the nth Fibonacci number, where n is a value entered by the user