1. **Avoiding Duplicates**[1]: In this exercise, you will create a program that reads words from the user until the user enters a blank line. After the user enters a blank line your program should display each word entered by the user exactly once. The words should be displayed in the same order that they were entered. For example, if the user enters:

```
first
second
first
third
second
```

then your program should display:

```
first
second
third
```

> Hint: You can check if a word already exists in the list before appending it.

2. **Formatting a List**[1]: When writing out a list of items in English, one normally separates the items with commas. In addition, the word “and” is normally included before the last item, unless the list only contains one item. Consider the following four lists:

```
apples
apples and oranges
apples, oranges and bananas apples,
oranges, bananas and lemons
```

> Hint: You can concatenate the strings until the last one using the comma as the separator and add the "and" for the last one. Optionally, you can do the concatenation with the `join` string method. `join` is used with the separator as the string that calls the `join method`. For example we can have `", ".join(["apples", "oranges", "banana"])` that creates the string `"apples, oranges, banana"`. More at https://docs.python.org/3/library/stdtypes.html#str.join

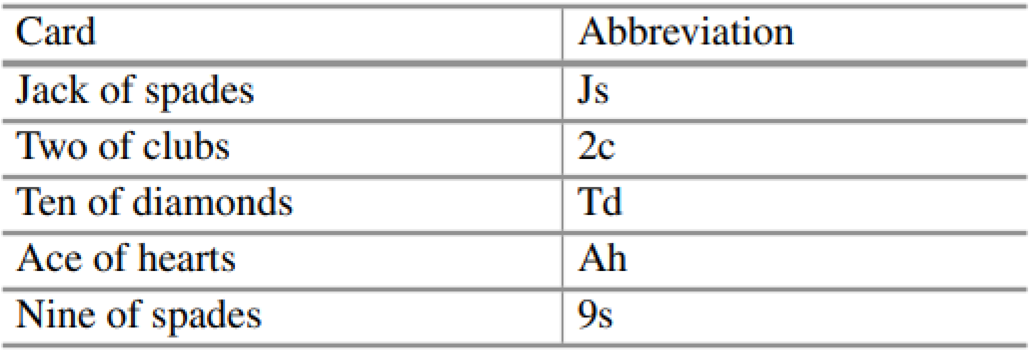
3. **Remove Outliers**[1]: When analyzing data collected as part of a science experiment it may be desirable to remove the most extreme values before performing other calculations. Write a function that takes a list of values and an non-negative integer, n, as its parameters. The function should create a new copy of the list with the n largest elements and the n smallest elements removed. Then it should return the new copy of the list as the function’s only result. The order of the elements in the returned list does not have to match the order of the elements in the original list. Write a main program that demonstrates your function. Your function should read a list of numbers from the user and remove the two largest and two smallest values from it. Display the list with the outliers removed, followed by the original list. Your program should generate an appropriate error message if the user enters less than 4 values

> Hint: You can use the `copy` module to make a copy of the original list. You can also user the `sort` method on the copied list in order to get easier access to the smallest and largest numbers. Since it would be a list with integers, would it be enough to make a shallow copy or would a deep copy be necessary? Try both to see if there are any differences.

4. **Generating URLs**: You are given the following base URL as a string: `BASE_URL = 'https://{environment}.happyapi.com/{endpoint}?access_token={access_token}'`. Given that the `environment` can be any of the `[production, staging, development, sandbox]`, `endpoint` can be any of the `[orders, transactions, balance, store]` and the `access_token` is something specific to a user, create all combinations of URLs by using the supplied `environment` and `endpoint` values. Read the access token from the user in the beginning. The access token can be any random string that is supplied by the user. Use both the "`string.format()`" method and the "f-string" methods to build the strings.

> Hint: Store the supplied values for environments and endpoints as lists and use a "for in for" to iterate all combinations. Don't forget to read from user's input for the access token value.

5. **Shuffling a Deck of Cards**[1]: A standard deck of playing cards contains 52 cards. Each card has one of four suits along with a value. The suits are normally spades, hearts, diamonds and clubs while the values are 2 through 10, Jack, Queen, King and Ace. Each playing card can be represented using two characters. The first character is the value of the card, with the values 2 through 9 being represented directly. The characters “T”, “J”, “Q”, “K” and “A” are used to represent the values 10, Jack, Queen, King and Ace respectively. The second character is used to represent the suit of the card. It is normally a lowercase letter: “s” for spades, “h” for hearts, “d” for diamonds and “c” for clubs. The following table provides several examples of cards and their two-character representations.



Card	Abbreviation
Jack of spades	Js
Two of clubs	2c
Ten of diamonds	Td
Ace of hearts	Ah
Nine of spades	9s

Begin by writing a function named `create_deck`. It will use loops to create a complete deck of cards by storing the two-character abbreviations for all 52 cards into a list. Return the list of cards as the function’s only result. Your function will not take any parameters.

Write a second function named `shuffle` that randomizes the order of the cards in a list. One technique that can be used to shuffle the cards is to make use of Python’s `random.shuffle` function. Use both of the functions described in the previous paragraphs to create a main program that displays a deck of cards before and after it has been shuffled.

> Hint: https://www.w3schools.com/python/ref_random_shuffle.asp

Don't forget to use the `main()` function when building your programs!

---

[1] - *Ben Stephenson, The Python Workbook*