1. Create a list called years\_list, starting with the year of your birth, and each year thereafter until the year of your fifth birthday. For example, if you were born in 1980. the list would be years\_list = [1980, 1981, 1982, 1983, 1984, 1985].

ANS: To create a list called `years\_list` with the years from your birth year until your fifth birthday, you can follow these steps. Replace `1980` with your birth year:

birth\_year = 1980

years\_list = [year for year in range(birth\_year, birth\_year + 6)]

print(years\_list)

For example, if your birth year is `1980`, running the above code will output:

[1980, 1981, 1982, 1983, 1984, 1985]

The `years\_list` contains the years from 1980 to 1985, inclusive.

1. In which year in years\_list was your third birthday? Remember, you were 0 years of age for your first year.

ANS: To find the year in `years\_list` when you had your third birthday, you need to consider that you were 0 years old for your first year. Therefore, your third birthday would have occurred when you were 3 years old.

Here's how you can find the year of your third birthday using the `years\_list`:

birth\_year = 1980

years\_list = [year for year in range(birth\_year, birth\_year + 6)]

year\_of\_third\_birthday = years\_list[3]

print(year\_of\_third\_birthday)

If your birth year is `1980`, running the above code will output:

1983

So, your third birthday would have been in the year 1983.

3.In the years list, which year were you the oldest?

ANS: To find the year when you were the oldest in the `years\_list`, you need to determine the maximum year in the list. Since the `years\_list` contains the years from your birth year until your fifth birthday, the maximum year in the list corresponds to the year of your fifth birthday.

Here's how you can find the year when you were the oldest using the `years\_list`:

birth\_year = 1980

years\_list = [year for year in range(birth\_year, birth\_year + 6)]

oldest\_year = max(years\_list)

print(oldest\_year)

If your birth year is `1980`, running the above code will output:

1985

So, in the `years\_list`, you were the oldest in the year 1985, which corresponds to your fifth birthday.

1. Make a list called things with these three strings as elements: "mozzarella", "cinderella", "salmonella".

ANS: List called `things` with the three specified strings as elements:

things = ["mozzarella", "cinderella", "salmonella"]

print(things)

The list `things` contains the following elements:

["mozzarella", "cinderella", "salmonella"]

1. Capitalize the element in things that refers to a person and then print the list. Did it change the element in the list?

ANS: To capitalize the element in the `things` list that refers to a person ("cinderella" in this case), you can use the `capitalize()` method. This method capitalizes the first character of a string while making the rest of the characters lowercase.

Here's how you can do it:

things = ["mozzarella", "cinderella", "salmonella"]

# Capitalize the element that refers to a person

things[1] = things[1].capitalize()

print(things)

Output:

['mozzarella', 'Cinderella', 'salmonella']

Yes, the element in the `things` list that refers to a person ("cinderella") has been changed to "Cinderella" with the first letter capitalized. Lists in Python are mutable, meaning you can modify their elements, and the change is reflected in the list.

1. Make a surprise list with the elements "Groucho," "Chico," and "Harpo."

ANS: Here's a surprise list with the elements "Groucho," "Chico," and "Harpo":

surprise = ["Groucho", "Chico", "Harpo"]

print(surprise)

The `surprise` list contains the following elements:

["Groucho", "Chico", "Harpo"]

1. Lowercase the last element of the surprise list, reverse it, and then capitalize it.

ANS: Sure! Here's how you can perform the requested operations on the last element of the `surprise` list:

surprise = ["Groucho", "Chico", "Harpo"]

# Lowercase the last element

last\_element\_lowercased = surprise[-1].lower()

# Reverse the last element

last\_element\_reversed = last\_element\_lowercased[::-1]

# Capitalize the last element

last\_element\_capitalized = last\_element\_reversed.capitalize()

# Replace the last element in the list

surprise[-1] = last\_element\_capitalized

print(surprise)

['Groucho', 'Chico', 'Oprah']

The last element "Harpo" was first converted to lowercase ("harpo"), then reversed ("oprah"), and finally capitalized ("Oprah"). The modified last element has been updated in the `surprise` list.

1. Make an English-to-French dictionary called e2f and print it. Here are your starter words: dog is chien, cat is chat, and walrus is morse.

ANS: Sure! Here's an English-to-French dictionary called `e2f` with the provided starter words:

e2f = {

"dog": "chien",

"cat": "chat",

"walrus": "morse"

}

print(e2f)

Output:

{'dog': 'chien', 'cat': 'chat', 'walrus': 'morse'}

The `e2f` dictionary contains the English words as keys and their corresponding French translations as values.

1. Write the French word for walrus in your three-word dictionary e2f.

ANS: In the `e2f` dictionary, the French word for "walrus" is "morse." So, you can access it as follows:

e2f = {

"dog": "chien",

"cat": "chat",

"walrus": "morse"

}

french\_word\_for\_walrus = e2f["walrus"]

print(french\_word\_for\_walrus)

Output:

morse

The French word for "walrus" is "morse" in the `e2f` dictionary.

1. Make a French-to-English dictionary called f2e from e2f. Use the items method.

ANS: You can create a French-to-English dictionary called `f2e` from the existing `e2f` dictionary using the `items()` method. The `items()` method returns a view object that displays a list of dictionary's key-value pairs, which you can then use to create the new dictionary.

Here's how you can do it:

e2f = {

"dog": "chien",

"cat": "chat",

"walrus": "morse"

}

# Create the French-to-English dictionary (f2e) from e2f using items()

f2e = {french: english for english, french in e2f.items()}

print(f2e)

Output:

{'chien': 'dog', 'chat': 'cat', 'morse': 'walrus'}

The `f2e` dictionary now contains the French words as keys and their corresponding English translations as values, reversed from the `e2f` dictionary.

1. Print the English version of the French word chien using f2e.

ANS: To print the English version of the French word "chien" using the `f2e` dictionary, you can simply access the value corresponding to the key "chien" in the `f2e` dictionary.

Here's how you can do it:f2e = {

'chien': 'dog',

'chat': 'cat',

'morse': 'walrus'

}

# Get the English version of "chien" using f2e

english\_word\_for\_chien = f2e['chien']

print(english\_word\_for\_chien)

Output:

dog

The English version of the French word "chien" is "dog" in the `f2e` dictionary.

1. Make and print a set of English words from the keys in e2f.

ANS: To create and print a set of English words from the keys in the `e2f` dictionary, you can use the `keys()` method to get a view object containing all the keys and then convert it to a set.

Here's how you can do it:

e2f = {

"dog": "chien",

"cat": "chat",

"walrus": "morse"

}

# Create a set of English words from the keys in e2f

english\_words\_set = set(e2f.keys())

print(english\_words\_set)

Output:

{'dog', 'cat', 'walrus'}

The `english\_words\_set` now contains a set of English words extracted from the keys of the `e2f` dictionary.

1. Make a multilevel dictionary called life. Use these strings for the topmost keys: 'animals', 'plants', and 'other'. Make the 'animals' key refer to another dictionary with the keys 'cats', 'octopi', and 'emus'. Make the 'cats' key refer to a list of strings with the values 'Henri', 'Grumpy', and 'Lucy'. Make all the other keys refer to empty dictionaries.

ANS: Sure! Here's a multilevel dictionary called `life` with the specified structure:

life = {

'animals': {

'cats': ['Henri', 'Grumpy', 'Lucy'],

'octopi': {},

'emus': {}

},

'plants': {},

'other': {}

}

print(life)

The `life` dictionary has the following structure:

{

'animals': {

'cats': ['Henri', 'Grumpy', 'Lucy'],

'octopi': {},

'emus': {}

},

'plants': {},

'other': {}

}

The 'animals' key refers to another dictionary with keys 'cats', 'octopi', and 'emus', and the 'cats' key refers to a list of strings with values 'Henri', 'Grumpy', and 'Lucy'. All the other keys ('plants' and 'other') refer to empty dictionaries.

1. Print the top-level keys of life.

ANS: To print the top-level keys of the `life` dictionary, you can use the `keys()` method on the dictionary. Here's how you can do it:

life = {

'animals': {

'cats': ['Henri', 'Grumpy', 'Lucy'],

'octopi': {},

'emus': {}

},

'plants': {},

'other': {}

}

# Print the top-level keys of the life dictionary

top\_level\_keys = life.keys()

print(top\_level\_keys)

Output:

dict\_keys(['animals', 'plants', 'other'])

The `top\_level\_keys` variable now contains the top-level keys of the `life` dictionary, which are 'animals', 'plants', and 'other'.

1. Print the keys for life['animals'].

ANS: To print the keys for the 'animals' dictionary, which is a sub-dictionary inside the `life` dictionary, you can use the `keys()` method on the 'animals' dictionary.

Here's how you can do it:

life = {

'animals': {

'cats': ['Henri', 'Grumpy', 'Lucy'],

'octopi': {},

'emus': {}

},

'plants': {},

'other': {}

}

# Print the keys for life['animals']

animal\_keys = life['animals'].keys()

print(animal\_keys)

Output:

dict\_keys(['cats', 'octopi', 'emus'])

The `animal\_keys` variable now contains the keys of the 'animals' dictionary, which are 'cats', 'octopi', and 'emus'.

1. Print the values for life['animals']['cats']

ANS: To print the values for the 'cats' key, which is a list inside the 'animals' dictionary, you can directly access the nested list using `life['animals']['cats']`.

Here's how you can do it:

life = {

'animals': {

'cats': ['Henri', 'Grumpy', 'Lucy'],

'octopi': {},

'emus': {}

},

'plants': {},

'other': {}

}

# Print the values for life['animals']['cats']

cat\_values = life['animals']['cats']

print(cat\_values)

Output:

['Henri', 'Grumpy', 'Lucy']

The `cat\_values` variable now contains the list of cat names ['Henri', 'Grumpy', 'Lucy'] from the 'cats' key inside the 'animals' dictionary.