Homework Problems

4-1. Pizzas: Think of at least three kinds of your favorit pizza. Store these pizza names in a list, and then use a `for` loop to print the name of each pizza.

- Modify your `for` loop to print a sentence using the name of the pizza instead of printing just the name of the pizza. For each pizza you should have one line of output containing a simple statement like *I like pepperoni pizza*.
- Add a line at the end of your program, outside the `for` loop, that states how much you like pizza. The output should consist of three or more lines about the kinds of pizza you like and then an additional sentence, such as *I really love pizza!*

In [1]:
```
pizzas = ['cheese', 'veggie', 'meat']
for name in pizzas:
    print(f"I like {name} pizza")
print("I really love pizza!")
```

```
I like cheese pizza
I like veggie pizza
I like meat pizza
I really love pizza!
```

4-2. Animals: Think of at least three different animals that have a common characteristic. Store the names of these animals in a list, and then use a `for` loop to print out the name of each animal.

- Modify your program to print a statement about each animal, such as *A dog would make a great pet.*
- Add a line at the end of your program stating what these animals have in common. You could print a sentence such as *Any of these animals would make a great pet!*

In [2]:
```
animals = ['zebra', 'monkey', 'lizard']
for animal in animals:
    print(f"A {animal} would make a great pet")
print("Any of these animals would make a great pet!")
```

```
A zebra would make a great pet
A monkey would make a great pet
A lizard would make a great pet
Any of these animals would make a great pet!
```

4-3 Counting to Twenty: Use a `for` loop to print the numbers from 1 to 20, inclusive

In [3]:
```
for number in range(1,21):
    print(number)
```

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

4-4. One Hundred: Make a list of the numbers from one to one hundred, and then use a `for` loop to print the numbers

In [4]:
```
numbers = list(range(1,101))
for number in numbers:
    print(number)
```

```
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

4-5. Summing a Million: Make a list of the numbers from one to one million, and then use `min()` and `max()` to make sure your list actually starts at one and ends at one million. Also use the `sum()` function to see how quickly Python can add numbers.

In [5]:
```
numbers_one_to_a_million = list(range(1,1000001))
print(min(numbers_one_to_a_million))
print(max(numbers_one_to_a_million))
print(sum(numbers_one_to_a_million))
```

```
1
1000000
500000500000
```

4-6. Odd Numbers: Use the third argument of the `range()` function to make a list of the odd numbers from 10 to 20. Use a `for` loop to print each number.

In [6]:
```
odd_numbers = range(11, 21, 2)
for number in odd_numbers:
    print(number)
```

```
11
13
15
17
19
```

4-7. Threes: Make a list of the multiples of 3 from 3 to 30. Use a `for` loop to print the numbers in your list.

In [7]:
```
multipules_of_3 = list(range(3, 31, 3))
for number in multipules_of_3:
    print(number)
```

```
3
6
9
12
15
18
21
24
27
30
```

4-8. Cubes: A number raised to the third power is called a *cube*. For example, the cube of 2 is written as 2**3 in Python. Make a list of the first 10 cubes (that is, the cube of each integer from 1 through 10), and use a `for` loop to print out the value of each cube.

In [8]:
```
cubes = []
for x in range(1,11):
    cubes.append(x**3)
    #print(cubes[x-1]) this works too
for cube in cubes:
    print(cube)
```

```
1
8
27
64
125
216
343
512
729
1000
```

4-9. Cube Comprehension: Use a list comprehension to generate a list of the first 10 cubes.

In [9]:
```
cubes = [x**3 for x in range(1,11)]
for cube in cubes:
    print(cube)
```

```
1
8
27
64
125
216
343
512
729
1000
```

4-10: Slices: Using one of the programs you wrote above, add several lines to the end of the program that do the following:

- Print the message *The first three items in the list are:*. Then use a slice to print the first three items from that programs's list
- Print the message *Three items from the middle of the list are:*. Use a slice to print three items from the middle of the list.
- Print the message *The last three items in the list are:*. Use a slice to print the last three items in the list

In [10]:
```
print(f"The first three items in the list are: {cubes[:3]}")
middle_index = (len(cubes)/2) - 1
print(f"The items from the middle of the list are: {cubes[int(middle_index-1)
print(f"The last three times in the list are: {cubes[-3:]}")
```

```
The first three items in the list are: [1, 8, 27]
The items from the middle of the list are: [64, 125, 216]
The last three times in the list are: [512, 729, 1000]
```

4-11. My Pizzas, Your Pizzas: Start with your program from **4-1**. Make a copy o fthe list of pizzas, and call it `friend_pizzas` . Then, do the following:

- Add a new pizza to the original list.
- Add a different pizza to the list `friend_pizzas` .
- Prove that you have two separate lists. Print the message *My favorite pizzas are:*, and then use a `for` loop to print the first list. Print the message *My friend's favorite pizzas are:*, and then use a `for` loop to print the second list. Make sure each new pizza is stored in the appropriate list.

In [11]:
```
friend_pizzas = pizzas[:]
pizzas.append('Combo')
friend_pizzas.append('Margherita')
print("My favorite pizzas are:")
for pizza in pizzas:
    print(pizza.title())
print("My friend's favorite pizzas are:")
for pizza in friend_pizzas:
    print(pizza.title())
```

```
My favorite pizzas are:
Cheese
Veggie
Meat
Combo
My friend's favorite pizzas are:
Cheese
Veggie
Meat
Margherita
```

4-12. Index Variables: Make a list of your favorite 5 vacation destinations from highest to lowest. Make another list of how much each vacaction would cost.

- Use the range function to index through your list from least favorite to most favorite destination.
- Use the reverse function to print the same information
- Use the range function to index through both lists to print the cost of each vacation option.

In [12]:
```
destinations = ['Japan', 'Thailand', 'Indonesia', 'Costa Rica', 'Mexico']
cost = ['500', '400', '300', '200', '100']
for x in range(len(destinations)-1, -1, -1):
    print(destinations[x])
destinations.reverse()
for destination in destinations:
    print(destination)
destinations.reverse()
print("The following destinations cost: ")
for x in range(len(destinations)):
    print(f"{destinations[x]}: ${cost[x]}")
```

```
Mexico
Costa Rica
Indonesia
Thailand
Japan
Mexico
Costa Rica
Indonesia
Thailand
Japan
The following destinations cost:
Japan: $500
Thailand: $400
Indonesia: $300
Costa Rica: $200
Mexico: $100
```

4-13: Buffet: A buffet-style restaurant offers only five basic foods. Think of five simple foods and store them in a tuple.

- Use a `for` loop to print each food the restaurant offers
- The restaurant changes its menu, replacing two of the items with different foods. Add a line that rewrites the tuple, and then use a `for` loop to print each of the items on the revised menu.
- Try to modify one of the items without re-writing the tuple, and make sure that Python rejects the change

In [13]:
```
foods = ('rice', 'noodles', 'soup', 'lettuce', 'bread')
for food in foods:
    print(food)
foods = ('rice', 'noodles', 'soup', 'cheese', 'beets')
for food in foods:
    print(food)
foods[1] = 'eggs'
```

```
rice
noodles
soup
lettuce
bread
rice
noodles
soup
cheese
beets

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[13], line 7
      5 for food in foods:
      6     print(food)
----> 7 foods[1] = 'eggs'

TypeError: 'tuple' object does not support item assignment
```

4-14. PEP 8: Look through the original PEP 8 style guide at [https://python.org/dev/peps/pep-0003/] (https://python.org/dev/peps/pep-0003/%5D). You won't use much of it now, but it might be interesting to skim through it.