Arrays

For this exercise, you will be provided with several different source files. You will need to edit these files to clean them up.

Review:

Exercise 1: Initialize an Array

Within main(), create an array of integers, of size 5, like this:

```
int myNumbers[5];
```

Assign values for each element of the array.
Note that, for an array of size 5, the indices range from 0 to 4. You access specific elements via their index, such as:

```
myNumbers[0] = 20;
cout << myNumbers[0] << endl;
```

After you've assigned values to each element of the array, use a **for loop** to iterate from `0 <= index <= 4` and display the value of every element of the array. (array at the "`index`" of the for-loop)

Example for-loop:

```
for ( int index = 0; index <= 4; index++ )
{
     …
}
```

Exercise 2: User input into an array

Within main(), create an array of strings, of size 5. Its name is flavors.

Create a for loop that will iterate from `0 <= index <= 4`, and for each element of the array, ask the user to enter a value for that element. Example output:

```
Please enter the flavor for icecream #0:
```

Use **`cin`** to get the user's input, and store it in the appropriate element at that index.

After the for loop is done and all the ice cream flavors have been entered, create another for-loop that will display the index number and then the ice cream flavor. Example output:

```
0. Chocolate
1. Vanilla
2. Strawberry
3. Pistachio
4. GreenTea
```

Exercise 3: Cats

Within main(), create the following variables:

- MAXIMUM_CATS, a **const** integer, initialized to 5.
- catNames, a string array of size MAXIMUM_CATS
- catAges, an integer array of size MAXIMUM_CATS
- amountOfCats, an integer, initialized to 0

Note that you create a const variable in the following way:

```
const MAXIMUM_CATS = 5;
```

Create a for loop that iterates from `0 <= index < MAXIMUM_CATS`. Within the for loop, have the user enter the cat name and cat age, and store those in the two arrays.

Afterwards, add 1 year to each of the cat's ages manually, by accessing each `catAge` by index, one at a time. Display the age of each cat.

Ask the user to input an amount of years to go through. Store this value in a variable called **yearCount**.

Use either a while loop or a for loop to iterate through that amount of years. For each year, add 1 year to the age of each cat. You can add years to the cat ages manually, or through a for loop.

> ***Hint 1:*** *If you're using a for-loop to increment the cats' ages, you will have the cat age for loop inside of the yearCount loop.*

Once you're done adding ages to all the cats, display all the cats and their ages again.

> ***Hint 2:*** *Any time you're iterating through all of the cats, you will use a for loop iterating from* `0 <= index < MAXIMUM_CATS,` *and access a specific cat with [index].*

> ***Hint 3:*** *To iterate through all the years, you can either use a for-loop, like* `0 <= years < yearCount` *, or a while loop that subtracts 1 from yearCount each time through the loop.B*