# *Lab Activity #6 - Arrays*

***Exercise #1:***

Write a program that declares an array named *alpha* with 50 components of the type **double**. Initialize the array so that the first 25 components are equal to the square of the counter (or index) variable and the last 25 components are equal to three times the index variable.

Output the array so that exactly ten elements per line are printed.

***Exercise #2:***

Write a program that declares an array named *myArray* with 8 components of the type **int**. Initialize the array to 8 values that the user inputs (remember to prompt the user). Finally, pass the array as a parameter to a new function called *filterEvens*. This new function will display all of the even numbers in the array.

***Exercise #3:***

A car dealership has 10 salespeople. The dealership keeps track of the number of cars sold by each salesperson each month and reports it to management. The management then takes that data and assigns a number 1 – 10 to each salesperson. The following statement declares an array to store the number of cars sold by each salesperson (it means that salesperson #1 sold 7 cars, salesperson #2 sold 3 cars, etc):

```
int cars[10] = {7, 3, 6, 0, 14, 8, 1, 2, 9, 8};
```

Output
1) the total number of cars sold at the entire dealership,
2) which salesperson sold the most cars (Salesperson #1, Salesperson #2, etc.), and
3) how many cars that best salesperson sold.

***Exercise #4:***

DNA is made up from 4 different bases (nucleotides), adenine (A), thymine (T), guanine (G) and cytosine (C). This is true for plants, animals, bacteria, in fact it is true all life forms on earth that contain DNA.

In an incredible molecular feat called transcription, your cells create molecules of messenger RNA that mirror the sequence of nucleotides in your DNA. The RNA then creates proteins that do the work of the cell.

Create a function called dna_to_rna, which should take as input a string which will have DNA nucleotides (capital letter As, Cs, Gs, and Ts). There may be other characters, too; they should be ignored by your transcribe function and disappear from the output. These might be spaces or other characters that are not DNA nucleotides.

Then, dna_to_rna should output the messenger RNA that would be produced from that DNA string. The correct output simply uses replacement:

- As in the input become Us in the output.
- Cs in the input become Gs in the output.
- Gs in the input become Cs in the output.
- Ts in the input become As in the output.
- any other input characters should disappear from the output altogether

Not quite working? One common problem that can arise is that dna_to_rna needs to have an ELSE to capture all of the non-legal characters. All non-nucleotide characters should be dropped.

Here are the tests to check:

ACGTTGCA should be transformed into UGCAACGU
ACG TGCA should be transformed into UGCACGU // note that the space disappears
GATTACA should be transformed into CUAAUGU
A42% should be transformed into U