Arrays

Purpose:

To be able to use and manipulate arrays.

Due Date:

The completed lab assignment is due Friday, November 30, 2012 *at the beginning of lecture*.

An Airplane booking simulation[1]

A small airline has just purchased a computer for its new automated reservations system. You have been asked to program the new system. You are to write a program to assign seats on each flight of the airline's only plane (capacity: 10 seats).

Your program should display the following menu of alternatives:

```
Please type 1 for "Business class"
Please type 2 for "Economy class"
```

If the person types 1, then your program should assign a seat in the business class section (seats 1-5). If the person types 2, then your program should assign a seat in the economy class section (seats 6-10). Your program should then print a boarding pass indicating the person's seat number and whether it is in the smoking or nonsmoking sections of the plane.

Use a singly subscripted array to represent the seating chart of the plane. Initialize all the elements of the array to `false` to indicate that all seats are empty. As each seat is assigned, set the corresponding elements of the array to `true` to indicate that the seat is no longer available.

Your program should, of course, never assign a seat that has already been assigned. When the business class section is full, your program should ask the person if it is acceptable to be placed in the economy (and vice versa). If yes, then make the appropriate seat assignment. If no, then print the message `"Next flight leaves in 3 hours."` Your script file should indicate thorough testing; include all cases such as when alternate seats are offered and when all seats are sold out.

[1] This problem was originally designed by Professor Haque, and slightly updated for this assignment.

Analysing Frequency Components of Sound

Computer sound files use a variety of different formats. One of the simplest consists of a sequence of numbers representing how loud the sound is at a particular time. Each number is a measurement of sound intensity, and typicially 44 000 such samples are collected per second.[2]

One of the questions that often interests us about sound is the strength of various frequency components. Suppose that we have an array `a` of size n that consists of floating-point numbers representing sound intensity.

For instance, a pure tone (sine wave) that goes through one complete cycle every d samples would have $\sin(2\pi k/d)$ in the k-th location (that is `a[k]`). Shifting the sound in time results in something that sounds the same, but where `a`$[k]$==$\cos(2\pi k/d)$. To measure the $2\pi/d$ frequency component in an array as above, we need to compute

$$A = \Sigma_{k=0}^{n-1}(\mathtt{a}[k]\sin(2\pi k/d)) \quad (1)$$

$$B = \Sigma_{k=0}^{n-1}(\mathtt{a}[k]\cos(2\pi k/d)) \quad (2)$$

$$C = \sqrt{A^2 + B^2} \quad (3)$$

These equations are more ferocious-looking than they actually are. For instance to compute `A`, we need code like

```
    double A = 0.0 ;
    for (int k=0;k<a.length;++k)
        {
        A += a[k]*Math.sin(2*Math.PI*k/d) ;
        }
```

⇒ Write a method with signature

```
public static double frequencyComponent(double [] a, double d)
```

that computes the frequency component C for period d as specified above.

Testing your method

To test your method, do the following. First create a *square-wave* array of size 40 000, where the entries in slots 0. . . 9 999 and 20 000. . . 29 999 are +1.0, and the entries in slots 10 000. . . 19 999 and 30 000. . . 39 999 are −1.0.

[2]If you are interested in this, see, for instance, `https://ccrma.stanford.edu/courses/422/projects/WaveFormat/`, `http://www-mmsp.ece.mcgill.ca/Documents/AudioFormats/WAVE/WAVE.html`, and `http://en.wikipedia.org/wiki/WAV`.
The format of `.wav` files for instance is slightly more complicated than assumed in this lab. For one thing, the data are frequently stored as 16-bit integers in little endian format, rather than as a sequence of `doubles`. Further more the first small segment of the file is additional information, and not actual sound data.

Determine the frequency component for various periods d. There should be a large component at period $d = 20000$, and noticeable components at $d = 20000/(2m+1)$ (where $m \geq 1$). Conversely, there should be almost zero values for periods $d = 10000/m$.

⇒ Write a small driver program that computes and prints various frequency components of the square-wave array and neatly displays the results.