Tutorial 6: C continue

Question 1: Pointer

1. Here are two ways we could store a collection of words in C.

   ```
   char hello[] = "Hello";
   char  cat[]      = "Cat";
   char world[] = "World";
   char* dict[] = {hello, cat, world};
   ```

   What do each of the following values describe? If possible, also give the value itself. If the value is undefined, explain why.

   (a) hello
   (b) *hello
   (c) &hello
   (d) hello+1
   (e) dict
   (f) *dict
   (g) **dict
   (h) dict[2][3]
   (i) world[10]
   (j) cat[0][0]

2. If you had a pointer to a char array, (like hello in the above example), and we passed this pointer to a function to compute the length of the array, how would the function known when to stop?

3. What about if we have an array of integers?

   ```
   int  arrayptr[] = {4,2,7,2,9,8};
   ```

   Given arrayptr, is it possible to compute how many numbers in the array that arrayptr points to?

4. Consider the following two functions.

   ```
   int addone(int x)
   {
       return x + 1;
   }

   void addone2(int* x)
   {
       *x = *x + 1;
   }
   ```

   We call the first style of function *pass-by-value* and the second *pass-by-reference*.[1]

   What do each of these functions do? Identify the differences.

[1] I'm lying here, pass by reference requires the ability to be able to modify the input to a function. C can't actually do this, but merely *emulates* pass-by-reference by passing a non-mutable pointer to a value, which can then be modified by writing data back to the memory address specified by the pointer.

5. Suppose you wanted to write a function to swap the value of two integers.

```
int x = 10;
int y = 20;
// call the swap function on x and y
// now x = 20, y = 10
```

Write such a function. Should you write it as a pass-by-value, or as a pass-by-reference[1]?

Question 2: Pointer and array

Write a program that creates an array of type float. How you populate this array is up to you. Print out the content of the array and the corresponding memory addresses for each entry. (you may want to try with both equivalent ways of referencing an array to make sure your syntax is up to speed).

Take note of the numeric difference between successive address entries. What does that suggest regarding the way the array is stored to memory? What is the size of a float entry?

Question 3: Structure

Implement, using structures and functions as appropriate, a program which requires you to enter multiple points in 3 dimensions. The points will have a name (one alphanumeric character) and three coordinates x, y, and z. Find and implement a suitable way to enter the points. The program, through an appropriate function, should calculate the centre of gravity of the point cloud as a new point (i.e., the average of each coordinate for all points entered)

The output should show a list of the points entered and the centre of gravity as a new point. (i.e., name, x, y, and z)