Revision Coding assignments

1. Let P and Q be two points in the three-dimensional plane. We say that P dominates Q if all the conditions x(P) ≥ x(Q), y(P) ≥ y(Q) and z(P) ≥ z(Q) are satisfied. A point P in a collection is called a champion if P dominates all other points in the collection. Not all collections contain champions (for example, consider the three points (1,1,1), (3,3,3), (2,2,4)). But if a champion exists in a collection, it is unique. In this case, the champion has both the largest x-coordinate, the largest y-coordinate, and the largest z-coordinate in the collection.

   Write a program that enters a loop. Each iteration of the loop body first reads the x-, y- and z-coordinates of a new point (call them x, y and z). Assume that x, y and z are integer values. If any one of x, y or z is negative, then the program terminates. Otherwise, the champion among all the points read so far is printed, provided that the champion exists. Maintain the largest x-coordinate and the largest y-coordinate read so far. Suppose that just before reading the current point (x, y, z), a champion existed. If the new point (x', y', z') dominates the old champion, then the new point now becomes the current champion. If the new point is dominated by the old champion, then the old champion continues to remain the current champion. Otherwise, the current champion ceases to exist.

2. Write a program that takes as input an integer **n**, an array A of **n** non-zero integers (positive / negative) and separates the positive integers from the negative ones.

3. Write a program that takes as input an integer n, an array A of n non-zero integers (positive / negative). It is assumed that all the negative elements of **A** appear before all the positive elements of **A**. Your task is to find out whether there exist indices **i** and **j** (with i < j) in **A** such that **A[i] + A[i+1] + ... + A[j] = 0.**

4. Write a program that takes as input an integer **n**, an array **A** of **n** integers in the range 0 to n-1, an integer **p** and returns a stroll starting from index **p** on the array A. A stroll on the array A is defined in the following way. You start from some index i in the range 0 to n – 1. You then visit the cells A[i], A[A[i]], A[A[A[i]]], ... in that sequence. The cells visited are i, A[i], A[A[i]], A[A[A[i]]], ...A walk starting from s will be eventually periodic. Indeed, there is an initial part of the walk (may be empty). However, after the first duplication of a visited index, the walk will be cyclic (that is, the same sequence of indices will be visited again and again). Print the initial part of the walk (before the cycle begins), and the indices in the cycle in the sequence as they are visited. Print also the length of the cycle.

n = 60

The array A is:
44 47 19 15 23 6 49 7 11 40
49 54 25 12 22 29 2 24 15 34
9 4 46 12 12 2 9 41 30 18
9 6 57 20 21 20 26 10 20 29
43 9 23 0 21 38 21 15 2 36
49 11 41 35 23 53 29 32 27 51

p = 31

Initial part:
31 6 49 36 26 9 40 43 0 44

Cyclic part:
21 4 23 12 25 2 19 34

Length of the cycle = 8

5. Write a program that takes as input an integer **n**, an array **A** of **n** integers in the range 0 to n-1, two integers **p** and **q** and returns an **s-t** common index stroll point on the array A. Consider the two strolls starting from s and t. You are required to find out whether the strolls share a common index. If not, print that and return. If there is a common index **r**, the function should print the walk from **s** to **r** and the walk from **t** to **r**. The index r should be so chosen that the two walks printed must not have any common index except the last one (r).

n = 60

The array A is:
33 33 56 35 14 9 53 31 44 23
13 9 35 58 50 22 31 27 40 52
21 35 15 29 8 17 31 54 38 3
32 12 36 20 47 43 30 32 14 14
56 20 16 23 18 6 45 50 33 17
34 46 45 41 15 53 58 38 48 37

s = 5
t = 8

+++ Path from Cell 5:
5 9 23

+++ Path from Cell 8:
8 44 18 40 56 58 48 33 20 21 35 43 23

6. Write a program that takes input an integer **n**, an array **A** of **n** integers and sorts it using Selection Sort.

7. Write a program that takes input an integer **n**, an array **A** of **n** integers and sorts it using Insertion Sort.

8. Write a program that takes in an array **A** of **n** digits (0 to 9) and prints the largest multiple of 3 that can be formed from the array elements. If the input array is {8, 1, 9}, the output should be 981, and if the input array is {8, 1, 7, 6, 0}, output should be 8760.

9. An array of size **n** is given. The array contains digits from 0 to 9. Generate the largest number using the digits in the array such that the number is divisible by 2, 3 and 5. For example, if the array is {1, 8, 7, 6, 0}, output must be: 8760. If the array is {7, 7, 7, 6}, output must be: “not possible”.

10. Write a program that takes as input an integer **n**, an array **A** of **n** integers and prints the index of any one of its peak elements. An element is called a peak element if its value is not smaller than the value of its adjacent elements (if they exist).