## Sheet 7: Intelligence honing problems on arrays

1. Given a 2D array, print it in spiral form. See the following examples.

Input:

1 2 3 4

5 6 7 8

9 10 11 12

13 14 15 16

Output:

1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10

Input:

1 2 3 4 5 6

7 8 9 10 11 12

13 14 15 16 17 18

Output:

1 2 3 4 5 6 12 18 17 16 15 14 13 7 8 9 10 11

1. Given an array of size n, generate and print all possible combinations of r elements in array. For example, if input array is {1, 2, 3, 4} and r is 2, then output should be {1, 2}, {1, 3}, {1, 4}, {2, 3}, {2, 4} and {3, 4}.
2. 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 arrays is {1, 8, 7, 6, 0}, output must be: 8760. And if the arrays is {7, 7, 7, 6}, output must be: “no number can be formed”.
3. The task here is to study lexicographic sorting on a set of n-tuples. This sort is based on the lexicographic order,<, defined between two tuples as: **(x1,x2,x3,. . . . . . ,xn)<=(y1,y2,y3,. . . . . ,yn) iff xi<=yi for all i=1,2,3,. . . . , n**

Use a 1-D array to represent a tuple.

1. Show how a 2-D array can be used to represent a list of n-tuples. Write a function displayListofTuples() to display this list in an appropriate format.
2. Write a function sortTuples() to sort tuples in the order defined above.
3. We now look at a slight modification of this task. Take 4 numbers from the user say a, b, c, d. Allow the tuples to contain elements from these four numbers only. The sequence in which the user enters these numbers is the new order that is now valid ie, if the user enters the numbers 2, 4, 3, 1 in this sequence then assume that 2<4<3<1 for the new scheme. Now modify the function sortTuples() to sort the list of Tuples based on this new order. Display the two lists: unsorted and sorted.

## Sheet 8: Geometry and arrays

1. Given two line segments (p1, q1) and (p2, q2), find if the given line segments intersect with each other.
2. Given three corner points of a triangle, and one more point P. Write a function to check whether P lies within the triangle or not.
3. Given n points on 2-D geometry, find the three points which are co-linear.
4. Given a number, find the next smallest palindrome larger than this number. For example, if the input number is “2 3 5 4 5″, the output should be “2 3 6 3 2″. And if the input number is “9 9 9″, the output should be “1 0 0 1″. The input is assumed to be an array. Every entry in array represents a digit in input number. Let the array be ‘num[]‘ and size of array be ‘n’.
5. Write a function void oddeven(int[] arr, int s) which takes an array and its size as parameters, and creates a two dimensional array with two rows and s columns. The function should put odd values of arr in first row of 2-D array and the even values in the second row. Unused cells of 2-D array should be filled with the zero. The function should display 2-D array.

## Sheet 9: Matrix revisited

1. Given a 2D matrix, print all elements of the given matrix in diagonal order. For example, consider the following 5 X 4 input matrix.

1 2 3 4

5 6 7 8

9 10 11 12

13 14 15 16

17 18 19 20

Diagonal printing of the above matrix is

1

5 2

9 6 3

13 10 7 4

17 14 11 8

18 15 12

19 16

20

1. You are standing at the point (*n*,*m*) of the grid of positive integers and want to go to origin (0,0) by taking steps either to left or down: that is, from each point (*n*,*m*) you are allowed to move either to the point (*n*-1,*m*) or the point (*n*,*m*-1). Write a method *intcountPaths(int n, int m)* that counts the number of different paths from the point (*n*,*m*) to the origin.
2. Given an unsorted integer (positive values only) array of size ‘n’, we can form a group of two or three, the group should be such that the sum of all elements in that group is a multiple of 3. Count all possible number of groups that can be generated in this way.
3. Given a boolean matrix mat[M][N] of size M X N, modify it such that if a matrix cell mat[i][j] is 1 (or true) then make all the cells of ith row and jth column as 1.

## Sheet 10: More and more matrix

1. Write a code which inputs two numbers m and n and creates a matrix of size m x n (m rows and n columns) in which every elements is either X or 0. The Xs and 0s must be filled alternatively, the matrix should have outermost rectangle of Xs, then a rectangle of 0s, then a rectangle of Xs, and so on. Examples:

Input: m = 3, n = 3

Output: Following matrix

X XX

X 0 X

X XX

1. Given an array of integers and a number x, find the smallest subarray with sum greater than the given value.

Examples:

arr[] = {1, 4, 45, 6, 0, 19}

x = 51

Output: 3

Minimum length subarray is {4, 45, 6}

1. Given a string, eliminate all “b” and “ac” in the string, you have to replace them in-place, and you are only allowed to iterate over the string once. (Source [Google Interview Question](http://www.careercup.com/question?id=18460667))

Examples:

acbac ==> ""

aaac ==> aa

ababac ==> aa

bbbbd ==> d

The two conditions are (if possible):  
**1.** Filtering of all ‘b’ and ‘ac’ should be in single pass  
**2.** No extra space allowed.

1. *char \* cyclicLeftShift(String s, int k)* that creates and returns a string in which character of s has been moved left *k* steps, so that the characters that would “fall off” the left end are appended to the right. For example, when called with the parameters (“Hello world”, 3), the method would return “lo worldHel”. Design your method so that it works even if *k* is greater than the length of *s*, or if *k* is negative (in which case the shift would take place to the right, not left). Note that *k* could be even as large as +1000000000, or as small as -1000000000. Make sure your method is efficient even in these cases.

## Sheet 11: Aspiring Mind questions

**1.** Given equations like 2x+3y+4z+.... let x=1 , y=2 ,z=3.... i.e. variables be denoted by int values and non zero coefficients are stored in a linked list perform adding of such two list and return the result

**2.** Given a map of an area in a 2d array grid of n and m  length and width. Let 1 denote the presence of a house in a cell and 0 the absence. If there are adjacent 1's sharing an edge they are part of same house. Develop a code to count the total # of houses and the max house width.

eg. 101

       110

       101

**3.** What operations can be done on floating number and their representation in machines?Write code for adding digits of 729.456.

**4.** Given strings abcdef and abd find which second  string characters occur in 1st string. Do it efficiently.