**Arrays:**

**1.** Write a C program that, given an array A[] of n numbers and another number x, determines whether or not there exist two elements in S whose sum is exactly x.

2. Write an efficient C program to find the sum of contiguous subarray within a one-dimensional array of numbers which has the largest sum.

3. You are given a list of n-1 integers and these integers are in the range of 1 to n. There are no duplicates in list. One of the integers is missing in the list. Write an efficient code to find the missing integer.

Example:

I/P [1, 2, 4 ,6, 3, 7, 8]

O/P 5

4. Write a function rotate(ar[], d, n) that rotates arr[] of size n by d elements.

5. Print the elements of an array in the decreasing frequency if 2 numbers have same frequency then print the one which came 1st  
E.g. 2 5 2 8 5 6 8 8 output: 8 8 8 2 2 5 5 6

6. Inversion Countfor an array indicates – how far (or close) the array is from being sorted. If array is already sorted then inversion count is 0. If array is sorted in reverse order that inversion count is the maximum.   
Formally speaking, two elements a[i] and a[j] form an inversion if a[i] > a[j] and i < j

7. **Write a C function to return minimum and maximum in an array. You program should make minimum number of comparisons.**

**8.** Write an efficient program for printing k largest elements in an array. Elements in array can be in any order. For example, if given array is [1, 23, 12, 9, 30, 2, 50] and you are asked for the largest 3 elements i.e., k = 3 then your program should print 50, 30 and 23.

9. Given an array arr[] of integers, find out the difference between any two elements **such that larger element appears after the smaller number** in arr[]. Examples: If array is [2, 3, 10, 6, 4, 8, 1] then returned value should be 8 (Diff between 10 and 2). If array is [ 7, 9, 5, 6, 3, 2 ] then returned value should be 2 (Diff between 7 and 9)

10. Given an array arr[] of n integers, construct a Product Array prod[] (of same size) such that prod[i] is equal to the product of all the elements of arr[] except arr[i]. Solve it **without division operator and in O(n)**.

11. Given an array of n elements which contains elements from 0 to n-1, with any of these numbers appearing any number of times. Find these repeating numbers in O(n) and using only constant memory space. For example, let n be 7 and array be {1, 2, 3, 1, 3, 6, 6}, the answer should be 1, 3 and 6.

12. Given an unsorted array arr[0..n-1] of size n, find the minimum length subarray arr[s..e] such that sorting this subarray makes the whole array sorted. **Examples:**  
 1) If the input array is [10, 12, 20, 30, 25, 40, 32, 31, 35, 50, 60], your program should be able to find that the subarray lies between the indexes 3 and 8.

2) If the input array is [0, 1, 15, 25, 6, 7, 30, 40, 50], your program should be able to find that the subarray lies between the indexes 2 and 5.

**13.** Given an array and a value, find if there is a triplet in array whose sum is equal to the given value. If there is such a triplet present in array, then print the triplet and return true. Else return false. For example, if the given array is {12, 3, 4, 1, 6, 9} and given sum is 24, then there is a triplet (12, 3 and 9) present in array whose sum is 24.

14. Tug of War

Given a set of n integers, divide the set in two subsets of n/2 sizes each such that the difference of the sum of two subsets is as minimum as possible. If n is even, then sizes of two subsets must be strictly n/2 and if n is odd, then size of one subset must be (n-1)/2 and size of other subset must be (n+1)/2.

For example, let given set be {3, 4, 5, -3, 100, 1, 89, 54, 23, 20}, the size of set is 10. Output for this set should be {4, 100, 1, 23, 20} and {3, 5, -3, 89, 54}. Both output subsets are of size 5 and sum of elements in both subsets is same (148 and 148).  
Let us consider another example where n is odd. Let given set be {23, 45, -34, 12, 0, 98, -99, 4, 189, -1, 4}. The output subsets should be {45, -34, 12, 98, -1} and {23, 0, -99, 4, 189, 4}. The sums of elements in two subsets are 120 and 121 respectively.

**15.** Given an array of random numbers, Push all the zero’s of a given array to the end of the array. For example, if the given arrays is {1, 9, 8, 4, 0, 0, 2, 7, 0, 6, 0}, it should be changed to {1, 9, 8, 4, 2, 7, 6, 0, 0, 0, 0}. The order of all other elements should be same. Expected time complexity is O(n) and extra space is O(1).

16. Given two arrays A1[] and A2[], sort A1 in such a way that the relative order among the elements will be same as those are in A2. For the elements not present in A2, append them at last in sorted order.

Input: A1[] = {2, 1, 2, 5, 7, 1, 9, 3, 6, 8, 8}

A2[] = {2, 1, 8, 3}

Output: A1[] = {2, 2, 1, 1, 8, 8, 3, 5, 6, 7, 9}

The code should handle all cases like number of elements in A2[] may be more or less compared to A1[]. A2[] may have some elements which may not be there in A1[] and vice versa is also possible.

**17.** 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}.

18. Given an integer array and a positive integer k, count all distinct pairs with difference equal to k.

Examples:

Input: arr[] = {1, 5, 3, 4, 2}, k = 3

Output: 2

There are 2 pairs with difference 3, the pairs are {1, 4} and {5, 2}

Input: arr[] = {8, 12, 16, 4, 0, 20}, k = 4

Output: 5

There are 5 pairs with difference 4, the pairs are {0, 4}, {4, 8},

{8, 12}, {12, 16} and {16, 20}

**Matrix:**

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

2. rotate matrix

3. print matrix diagonally

**Strings:**

1. Check if a given string is a rotation of a palindrome

Given a string, check if it is a rotation of a palindrome. For example your function should return true for “aab” as it is a rotation of “aba”.

2. Given two strings, find if first string is a subsequence of second

Given two strings str1 and str2, find if str1 is a subsequence of str2. A subsequence is a sequence that can be derived from another sequence by deleting some elements without changing the order of the remaining elements. Expected time complexity is linear.

Examples:

Input: str1 = "AXY", str2 = "ADXCPY"

Output: True (str1 is a subsequence of str2)

3. Given a sorted dictionary (array of words) of an alien language, find order of characters in the language.

**Examples:**

Input: words[] = {"baa", "abcd", "abca", "cab", "cad"}

Output: Order of characters is 'b', 'd', 'a', 'c'

Note that words are sorted and in the given language "baa"

comes before "abcd", therefore 'b' is before 'a' in output.

Similarly we can find other orders.

Input: words[] = {"caa", "aaa", "aab"} Output: Order of characters is 'c', 'a', 'b'

4. Anagram Substring Search (Or Search for all permutations)

Given a text txt[0..n-1] and a pattern pat[0..m-1], write a function search(char pat[], char txt[]) that prints all occurrences of pat[] and its permutations (or anagrams) in txt[]. You may assume that n > m.   
Expected time complexity is O(n)

Examples:

1) Input: txt[] = "BACDGABCDA" pat[] = "ABCD"

Output: Found at Index 0

Found at Index 5

Found at Index 6

5. Recursively remove all adjacent duplicates

Given a string, recursively remove adjacent duplicate characters from string. The output string should not have any adjacent duplicates. See following examples.

Input: azxxzy

Output: ay

First "azxxzy" is reduced to "azzy". The string "azzy" contains duplicates,

so it is further reduced to "ay".

Input: geeksforgeeg

Output: gksfor

First "geeksforgeeg" is reduced to "gksforgg". The string "gksforgg" contains

duplicates, so it is further reduced to "gksfor".

Input: caaabbbaacdddd

Output: Empty String

Input: acaaabbbacdddd

Output: acac

6. 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:  
**1.** Filtering of all ‘b’ and ‘ac’ should be in single pass  
**2.** No extra space allowed.

7. Print all permutations in sorted (lexicographic) order

Given a string, print all permutations of it in sorted order. For example, if the input string is “ABC”, then output should be “ABC, ACB, BAC, BCA, CAB, CBA”.

8. Write a function to check whether two given strings are [anagram](http://en.wikipedia.org/wiki/Anagram)of each other or not. An anagram of a string is another string that contains same characters, only the order of characters can be different. For example, “abcd” and “dabc” are anagram of each other.

9. Print list items containing all characters of a given word

There is a list of items. Given a specific word, e.g., “sun”, print out all the items in list which contain all the characters of “sum”

For example if the given word is “sun” and the items are “sunday”, “geeksforgeeks”, “utensils”, “”just” and “sss”, then the program should print “sunday” and “utensils”.

10. Given a string, find its first non-repeating character

Given a string, find the first non-repeating character in it. For example, if the input string is “GeeksforGeeks”, then output should be ‘f’ and if input string is “GeeksQuiz”, then output should be ‘G’.

11. Write a program to print N equal parts of a given string.

12. Write a C program to print all permutations of a given string

A permutation, also called an “arrangement number” or “order,” is a rearrangement of the elements of an ordered list S into a one-to-one correspondence with S itself. A string of length n has n! permutation.

13. Write a recursive C function to print reverse of a given string.