

CSE 4304-Data Structures Lab. Winter 2024-25

Date: 12 November 2025

Target Group: All groups

Topic: Basic tasks, Arrays, recursion, logic-based problems.

Instructions:

- Task naming format: fullID_T01L01_1A.c/cpp
- If you find any issues in problem description/test cases, comment in the google classroom.
- If you find any test case that is tricky that I didn't include but others might forget to handle, please comment! I'll be happy to add.
- Modified sections will be marked with **BLUE** color.

Group	Tasks
1A/1B/2A/2B	1 4 8 9

Task-01:

Assume your program will take a set of positive integers(1~9) as input and store them in an array. It will stop taking input for any negative number or zero. Now, the program will show how many times each of the numbers have occurred. (Don't show anything for a number that didn't occur.)

Input	Output
3 4 5 7 8 9 2 3 3 4 4 5 0	3 occurs 3 times 4 occurs 3 times 5 occurs 2 times 7 occurs 1 time 8 occurs 1 time 9 occurs 1 time 2 occurs 1 time
5 2 2 3 3 3 1 1 9 -5	3 occurs 3 times 2 occurs 2 times 1 occurs 2 times 5 occurs 1 time 9 occurs 1 time

Note:

- Output is arranged based on the occurrence of each number. If the frequency of two numbers is the same, it's not mandatory to sort based on the relative ordering.
- **STL libraries are not allowed for this task. Implement the sort function manually (if needed). Better if you dont use sort!**

Task-02:

Write a program that takes a positive integer number(n) as input and calculates the factorial(n!) value of the number.

$$n! = n \times (n - 1) \times (n - 2) \times \dots \times 1$$

The factorial value should be calculated in two different ways using the following function:

- Factorial_using_iteration (int n): returns int
- Factorial_using_recursion (int n): returns int

Input	Output
5	120 (using iteration) 120 (using recursion)
10	3628800 (using iteration) 3628800 (using recursion)
0	1 (using iteration) 1 (using recursion)

Task-03

‘Binary-search’ is an algorithm that is based on compare and split mechanism. This algorithm is also known as ‘half-interval search’, ‘logarithmic search’, or ‘binary chop’.

The objective is to search for the position of the target value in a sorted array. It compares the target value with the middle element of the array. If the element is equal to the target element, then the algorithm returns the index of the found element. Otherwise, the searching algorithm uses a half section of that array, based on the comparison value, it uses either the first half (when the value is less than the middle) or the second half (when the value is greater than the middle). Then, it does the same for the next half.

Your task is to implement this algorithm in two ways using the following functions:

- Iterative_binary_search(int x, int array[]): returns index
- Recursive_binary_search(int x, int array[]): returns index

Input	Output
0 2 6 11 12 18 34 45 55 99 -1 55	8 (using iteration) 8 (using recursion)
25 33 37 43 55 60 -1 42	-1 (using iteration) -1 (using recursion)
2 3 7 13 15 20 25 33 -1 -1	-1 (using iteration) -1 (using recursion)

[Note:

- The input array is given in a sorted sequence.
- **Using iterations** means using a loop inside the function that checks for the equality of the middle element.
- **Using recursion** means the function calls itself again and again.
- Array indexing starts from 0.
- It will stop taking input for any negative number.]

Task 4:

IUT hired a group of workers to build a new dormitory. The energy level of each worker varies. Each day, they have new tasks where they have to nominate two workers to work in a pair. The task has a defined energy level (let's say, 'n'). So two workers with energy levels n_1 , and n_2 have to be selected such that $n=n_1+n_2$.

Your task is to find the number of possible pairs of workers that can be nominated to solve the task.

Notes:

- Assuming the serial of worker starts from 1
- Stop taking input when you get 0
- The energy level of workers can be positive/negative (except zero).
- In the output, you have to show the number of pairs and the workers who form the pairs. (worker1,worker-2) means the same pair as (worker-2,worker-1) so don't consider both of them!

Input	Output
1 5 7 -1 0 6	2 Pairs: (worker-1, worker-2) (worker-3, worker-4)
1 5 7 -1 5 0 6	3 Pairs: (worker-1, worker-2) (worker-3, worker-4) (worker-1, worker-5)
1 1 1 1 0 2	6 pairs: (worker-1, worker-2) (worker-1, worker-3) (worker-1, worker-4) (worker-2, worker-3) (worker-2, worker-4) (worker-3, worker-4)
10 12 15 -1 7 6 5 4 2 1 1 1 0 11	6 pairs (worker-1, worker-10) (worker-1, worker-11) (worker-1, worker-12) (worker-2, worker-4) (worker-5, worker-8) (worker-6, worker-7)
1 5 3 7 -1 0 6	2 Pairs: (worker-1, worker-2) (worker-4, worker-5)
1 5 7 -1 0 5	No pair found!

Note: STL libraries may be used for this task

Task 05:

Given two integer arrays, num1, and num2, return an array of their intersection. Each element in the result must appear as many times as it shows in both arrays, and you may return the result in any order.
(keep input for an array until you get -1)

Input	Output	Clarification
1 2 2 1 -1 2 2 -1	2 2	
4 9 5 -1 9 8 9 8 4 -1	4 9	(9 4) is also acceptable.

Task 06:

The Fibonacci sequence is a series where the next term is the sum of the previous two terms. The first two terms of the Fibonacci series are 0 followed by 1.

The Fibonacci sequence: 0, 1, 1, 2, 3, 5, 8, 13, 21 ...

Given a number 'n', your task is to write a function for calculating the n-th Fibonacci number of the series.

Solve the problem in both **iterative** and **recursive** approaches.

Input	Output
10	34
20	4181

Task 07:

Xenia the beginner mathematician is a third-year student at elementary school. She is now learning the addition operation.

The teacher has written down the sum of multiple numbers. Pupils should calculate the sum. To make the calculation easier, the sum only contains digits. Still, that isn't enough for Xenia. She is only beginning to count, so she can calculate a sum only if the summands follow in non-decreasing order. For example, she can't calculate the sum $1+3+2+1$ but she can calculate sums $1+1+2$ and $3+3$.

You've got the sum that was written on the board. Rearrange the terms and print in such a way that Xenia can calculate.

Input	Output
$3+2+1$	$1+2+3$
$1+1+3+1+3$	$1+1+1+3+3$
2	2
$5+2+2+4$	$2+2+4+5$

Task 08:

One measure of ‘undersortedness’ in a sequence is the number of pairs or entries that are out of order with respect to each other.

For instance, in the letter sequence ‘DAABEC’, the measure is 5, since D is greater than four letters to its right and E is greater than one letter to its right. This measure is called the number of inversions in the sequence. The sequence ‘AACEDGG’ has only one inversion (E and D) and it is nearly sorted. On the other hand, the sequence ‘ZWQM’ has 6 inversions; it is as unsorted as can be, exactly in the reverse order.

You are responsible for cataloging a sequence of DNA strings (sequences containing only the four letters A, C, G, and T). However, you want to catalog them not in alphabetical order, but in the order of ‘sortedness’- from the ‘most sorted’ to ‘least sorted’. All the strings are of the same length.

The first line of input will contain two numbers ‘length’ and ‘n’ representing the length of each sequence and the total number of strings.

Your task is to arrange the string from ‘most sorted’ to ‘least sorted’ order. On the right of each string, show the degree of unsortedness.

Input	Output
10 6 AACATGAAGG TTTGGCCAA TTTGGCCAA GATCAGATT CCCGGGGGGA ATCGATGCAT	CCCGGGGGGA 9 AACATGAAGG 10 GATCAGATT 11 ATCGATGCAT 17 TTTGGCCAA 36 TTTGGCCAA 37

Task-09:

Alexander wants to fight for the Coding Club. In each round, there will be S soldiers with various powers. There will be R rounds to fight and in each round, his power will change. With power P , he can kill all the soldiers whose power is less than or equal. The sum of powers of the soldiers will be considered as the total point for that round. After each round, all the soldiers who are dead in the previous round will be reborn. So in each round, there will be N soldiers to fight.

Your job is to count the number of soldiers that he can kill in each round and provide the total points that Alexander has achieved.

Input will start with N & R followed by N power levels of the soldiers. The next lines will contain the R power levels of Alexander for the different rounds.

Input	Output	Explanation
7 3 2 3 1 6 5 7 4 3 10 2	3 6 7 28 2 3	7 3: 7 soldiers, 3 rounds Then the energy levels are given - For input 3: we can beat 3 soldiers with energy levels less than 7 ($2+3+1=6$). Hence output is 3 6. - For input 10: He can beat all soldiers. Hence answer is 7 28 - For Energy level=2: only soldiers with energy 1 & 2 can be beaten. So the answer is 2 3.
10 8 42 12 25 13 5 55 43 41 17 35 2 39 13 22 73 29 0 35	0 0 6 107 3 30 4 47 10 288 5 72 0 0 6 107	

Note:

- Try to optimize the time-complexity of your code as much as possible.
- Do not use any built in function.

Task-10:

Suppose, you are playing the ‘Bulls and Cows’ game with your friend. You write down a secret number and ask your friend to guess what the number is. When your friend makes a guess, you provide a hint with the following info:

- The number of ‘bulls’: which are digits in the guess that are in the correct position.
- The number of ‘cows’: which are digits in the guess that is in your secret number but are located in the wrong position. Specifically, the non-bull digits in the guess that could be rearranged such that they become bulls.

Given the secret number `secret` and your friend's guess, return *the hint for your friend's guess*. The hint should be formatted as "`xByC`", where `x` is the number of bulls and `y` is the number of cows. Note that both `secret` and `guess` may contain duplicate digits.

The first line of input will contain the secret and the following line will be the guess. The secret and guess consist of only digits and have equal lengths.

Input	Output	Explanation (Bulls are connected with a ' ', and cows are underlined)
1807 7810	1B3C	1807 <u>7810</u>
1123 0111	1B1C	1123 <u>0111</u> or <u>0111</u> (Note that only one of the two unmatched 1s is counted as a cow since the non-bull digits can only be rearranged to allow one 1 to be a bull)
1123 0145	1B0C	1123 0145
2441139 1234522	0B4C	2441139 <u>1234322</u> or <u>1234322</u> or <u>1234322</u>
2441139 4444111	3B1C	2441139 <u>4444111</u> or <u>4444111</u> (There is only one '1' in the secret, don't confuse with the two '1's in the guess)

Bulls and Cows wiki: https://en.wikipedia.org/wiki/Bulls_and_Cows

Note: Bonus marks if you can bring an O(N) solution.

Task-11:

Timmy starts his new job as a DJ in dubstep nation. In dubstep nations, the DJs are supposed to create a new type of music mix made with the word ‘dub’. The process is like, every original song must start with the word “dub” and after every word of the original lyric the word “dub” should be added. But Timmy does not like this music. So he decided to get the original song lyrics from the mixed dubstep music. Your task is to help Timmy get the original song lyrics from dubstep.

The first line of input contains the total length of the dubstep song. The second line contains the dubstep song lyric.

Input	Output
37 dubneverdubgonnadubgivedubyoudubup	never gonna give you up
47 dubhisdubpalmsdubaredubsweatydubkneesd ubweakdub	his palms are sweaty knees weak

Task-12:

Suppose, you are a professional robber planning to rob houses along a street. Each house has a certain amount of money stashed, the only constraint stopping you from robbing each of them is that adjacent houses have security systems connected and it will automatically contact the police if two adjacent houses were broken into on the same night.

(Consecutive houses have a unique amount of money)

Given an integer array nums representing the amount of money of each house, return the maximum amount of money you can rob tonight without alerting the police. (Take input until -1 is given)

Input	Output	Clarification (Picked houses)
1 2 3 1 -1	4	1 3
2 7 9 3 1 -1	12	2 9 1
2 1 1 2 -1	4	2 2
1 2 7 2 1 9 -1	17	1 7 9

Task-13:

Two arrays are given, one containing the positions and another containing the values. Your job is to rearrange the values based on the positions.

The first line contains the position values. Keep taking input in the first line until -1 is given. The following line will contain the values corresponding to those positions.

Input	Output
3 1 2 -1 32 54.7 -2	54.7 -2 32
5 1 6 4 3 2 -1 3.75 3.88 3.75 3.92 3.99 3.84	3.88 3.84 3.99 3.92 3.75 3.75

Note: Bonus marks for O(N) solution.

Task-14:

Given your total obtained marks in the final exam (out of 100), your task is to return the obtained grade.

The first line of input will contain two numbers ‘level’ and ‘trials’. Then each line will contain a number and the associated grade level.

After the levels are inserted, the trials will be inserted. Each trial will contain an integer value.

For each trial, you have to print the corresponding grade.

Input	Output
J	F D B+ A+ B
7 9	F
39 F	F
40 D	F
50 C	A*
60 B	A+
70 A-	A-
80 A+	B
90 A*	B
39	C
35	
0	
100	
89	
77	
60	
61	
59	

Note: Try to find the most efficient solution.