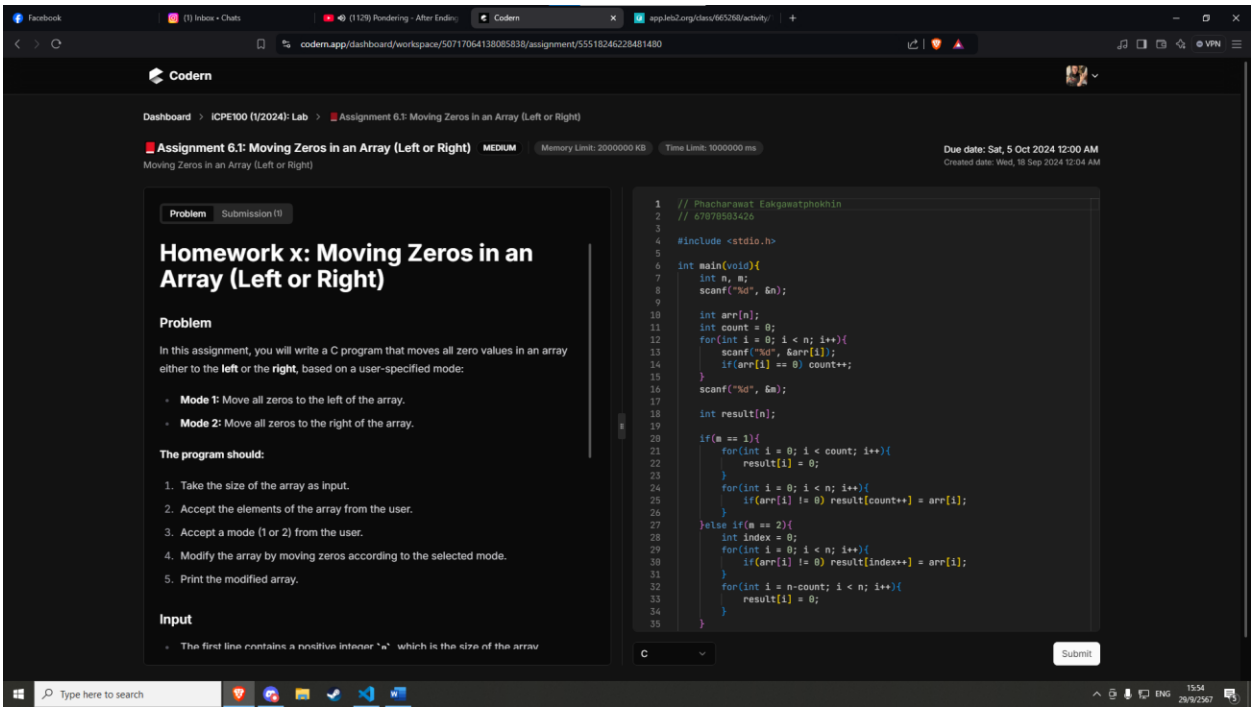
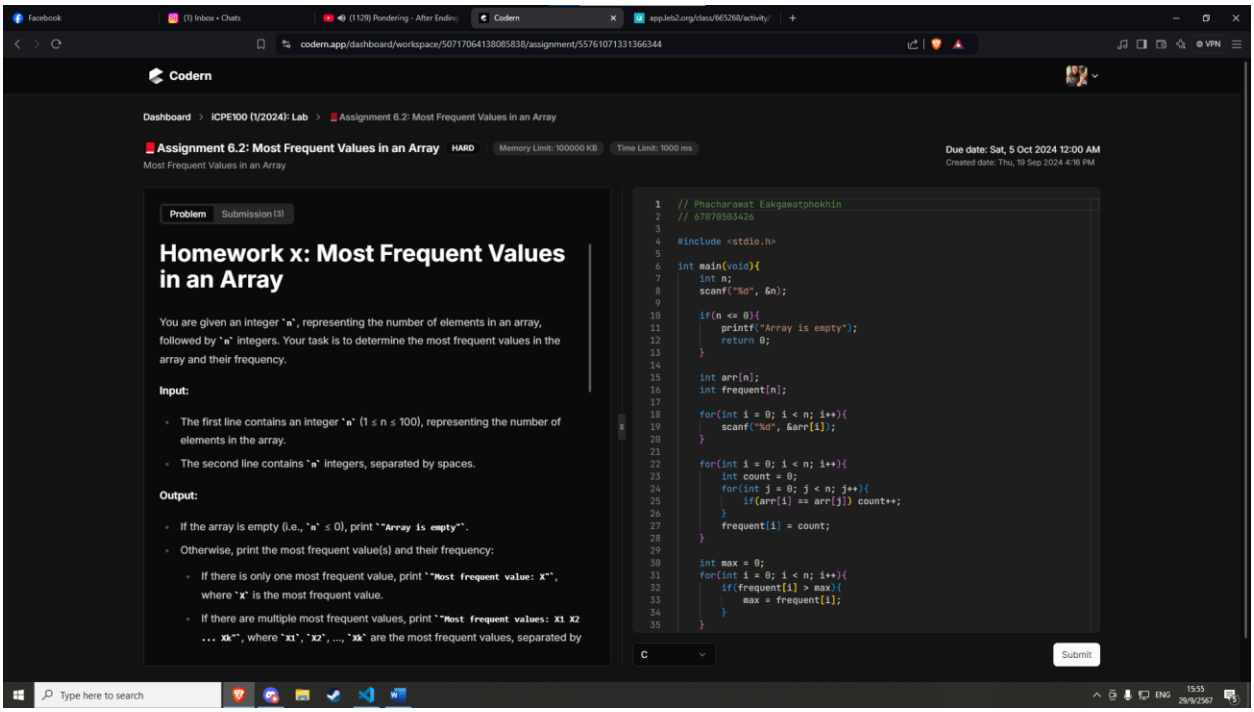
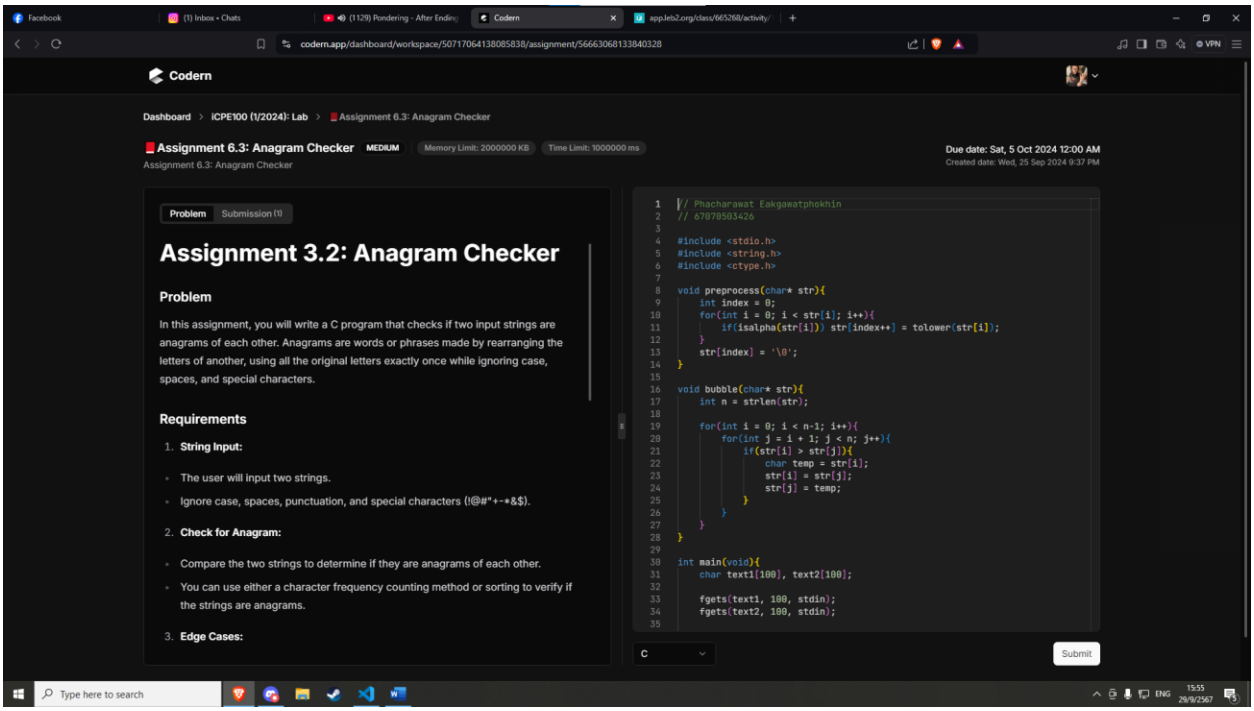

codern.app/dashboard/workspace/5071706413085838/assignment/55518246228481480
Codern
Dashboard > iCPE100 (1/2024): Lab > Assignment 6.1: Moving Zeros in an Array (Left or Right)
Assignment 6.1: Moving Zeros in an Array (Left or Right) MEDIUM
Memory Limit: 2000000 KB
Time Limit: 1000000 ms
Due date: Sat, 5 Oct 2024 12:00 AM
Created date: Wed, 18 Sep 2024 12:04 AM
Moving Zeros in an Array (Left or Right)
Problem
Submission (1)
Homework x: Moving Zeros in an Array (Left or Right)
Problem
In this assignment, you will write a C program that moves all zero values in an array either to the left or the right, based on a user-specified mode:
Mode 1: Move all zeros to the left of the array.
Mode 2: Move all zeros to the right of the array.
The program should:
1. Take the size of the array as input.
2. Accept the elements of the array from the user.
3. Accept a mode (1 or 2) from the user.
4. Modify the array by moving zeros according to the selected mode.
5. Print the modified array.
Input
The first line contains a positive integer `n`, which is the size of the array
// Phacharawat Eakgawatphokhin
// 67070503426
#include <stdio.h>
int main(void){
int n, m;
scanf("%d", &n);
int arr[n];
int count = 0;
for(int i = 0; i < n; i++){
scanf("%d", &arr[i]);
if(arr[i] == 0) count++;
}
scanf("%d", &m);
int result[n];
if(m == 1){
for(int i = 0; i < count; i++){
result[i] = 0;
}
for(int i = 0; i < n; i++){
if(arr[i] != 0) result[count++] = arr[i];
}
}else if(m == 2){
int index = 0;
for(int i = 0; i < n; i++){
if(arr[i] != 0) result[index++] = arr[i];
}
for(int i = n-count; i < n; i++){
result[i] = 0;
}
}
C
Submit
Type here to search
ENG
15:54
29/9/2567


codern.app/dashboard/workspace/5071706413085838/assignment/55761071331366344
Codern
Dashboard > iCPE100 (1/2024): Lab > Assignment 6.2: Most Frequent Values in an Array
Assignment 6.2: Most Frequent Values in an Array HARD
Memory Limit: 100000 KB
Time Limit: 1000 ms
Due date: Sat, 5 Oct 2024 12:00 AM
Created date: Thu, 19 Sep 2024 4:16 PM
Most Frequent Values in an Array
Problem
Submission (3)
Homework x: Most Frequent Values in an Array
You are given an integer `n`, representing the number of elements in an array, followed by `n` integers. Your task is to determine the most frequent values in the array and their frequency.
Input:
The first line contains an integer `n` (1 ≤ n ≤ 100), representing the number of elements in the array.
The second line contains `n` integers, separated by spaces.
Output:
If the array is empty (i.e., `n` ≤ 0), print `"Array is empty"`.
Otherwise, print the most frequent value(s) and their frequency:
If there is only one most frequent value, print `"Most frequent value: X"`, where `X` is the most frequent value.
If there are multiple most frequent values, print `"Most frequent values: X1 X2 ... Xk"`, where `X1`, `X2`, ..., `Xk` are the most frequent values, separated by
// Phacharawat Eakgawatphokhin
// 67070503426
#include <stdio.h>
int main(void){
int n;
scanf("%d", &n);
if(n <= 0){
printf("Array is empty");
return 0;
}
int arr[n];
int frequent[n];
for(int i = 0; i < n; i++){
scanf("%d", &arr[i]);
}
for(int i = 0; i < n; i++){
int count = 0;
for(int j = 0; j < n; j++){
if(arr[i] == arr[j]) count++;
}
frequent[i] = count;
}
int max = 0;
for(int i = 0; i < n; i++){
if(frequent[i] > max){
max = frequent[i];
}
}
C
Submit
Type here to search
ENG
15:55
29/9/2567


Facebook
(1) Inbox • Chats
(1129) Pondering - After Ending
Codern
app.leb2.org/class/665268/activity/
codern.app/dashboard/workspace/5071706413085838/assignment/56663068133840328
Codern
Dashboard > iCPE100 (1/2024): Lab > Assignment 6.3: Anagram Checker
Assignment 6.3: Anagram Checker
MEDIUM
Memory Limit: 2000000 KB
Time Limit: 1000000 ms
Due date: Sat, 5 Oct 2024 12:00 AM
Created date: Wed, 25 Sep 2024 9:37 PM
Assignment 6.3: Anagram Checker
Problem
Submission (1)
Assignment 3.2: Anagram Checker
Problem
In this assignment, you will write a C program that checks if two input strings are anagrams of each other. Anagrams are words or phrases made by rearranging the letters of another, using all the original letters exactly once while ignoring case, spaces, and special characters.
Requirements
1. String Input:
The user will input two strings.
Ignore case, spaces, punctuation, and special characters (!@#"+-*&$).
2. Check for Anagram:
Compare the two strings to determine if they are anagrams of each other.
You can use either a character frequency counting method or sorting to verify if the strings are anagrams.
3. Edge Cases:
// Phacharawat Eakgawatphokhin
// 6707050426
#include <stdio.h>
#include <string.h>
#include <ctype.h>
void preprocess(char* str){
int index = 0;
for(int i = 0; i < str[i]; i++){
if(isalpha(str[i])) str[index++] = tolower(str[i]);
}
str[index] = '\0';
}
void bubble(char* str){
int n = strlen(str);
for(int i = 0; i < n-1; i++){
for(int j = i + 1; j < n; j++){
if(str[i] > str[j]){
char temp = str[i];
str[i] = str[j];
str[j] = temp;
}
}
}
}
int main(void){
char text1[100], text2[100];
fgets(text1, 100, stdin);
fgets(text2, 100, stdin);
C
Submit
Type here to search
ENG
15:55
29/9/2567