

# Lab 03

## 1D Arrays



Department of Software Engineering-FIT-VNU-HCMUS

# 1.

## Content

In this lab, we will practice the following topics:

- Divide the source code into multiple files
- Write functions manipulating in arrays

## 2. Assignments

YY:  $30 + 5 + 5 + 5 + 5 = 50$

W04: YY: 25

1. A: YY: 05
2. H: YY: 25

W05: The remaining, YY: 25

1. A: YY: 05
2. H: YY: 25

Besides the `main()` function, students are asked to write at least one more function in the following assignments. The source code should be located in 3 different files (or more).

- A header file which containing all declarations.
- A source file which containing all function definitions.
- Another source file which containing the `main()` function.

### 2.1 Basic Array Operations

YY: 30

Perform the following operations in integer arrays.

In the `main()` function, create a menu allowing users to call each operation.

1. Write a function to input a one-dimensional array of integers.
2. Write a function to print a one-dimensional array of integers to the console.
3. Write a function to find the maximum value in a one-dimensional array of integers.
4. Write a function to find the first even number in a one-dimensional array of integers. If there is no even number, return -1.
5. Write a function to find the smallest even number in a one-dimensional array of integers. If there is no even number, return -1.
6. Write a function to find the first position of the smallest even number in a one-dimensional array of integers. If there is no even number, return -1.

7. Given a one-dimensional array of integers, find the value of  $x$  such that the interval  $[-x, x]$  contains all the values in the array.
8. Given a one-dimensional array of integers, write a function to find the first value in the array that has the form  $2^k$  (a power of 2). If no such value exists, return 0.
9. List the values in the array that satisfy the condition of being smaller than the absolute value of the next element and greater than the absolute value of the previous element.
10. List the positions of the largest even numbers in a one-dimensional array of integers.
11. Calculate the sum of the values in a one-dimensional array of integers whose first digit is an odd number.
12. Calculate the average of the positive numbers in a one-dimensional array of integers.
13. Count the number of prime numbers in a one-dimensional array of integers.
14. Count the number of distinct values in a one-dimensional array of integers.
15. List the frequency of each distinct value in a one-dimensional array of integers (each value should be listed only once).
16. Given two arrays,  $a$  and  $b$ , list the values that appear in only one of the two arrays.
17. Check if the one-dimensional array of integers is sorted.
18. Determine whether the elements in the array form an arithmetic sequence. If they do, identify the common difference  $d$ .
19. Sort the array in ascending order.
20. Sort the odd numbers in ascending order while keeping all other values in their original positions.
21. Given two arrays,  $a$  and  $b$ , determine whether  $b$  is a permutation of  $a$  (i.e., whether  $b$  contains the same elements as  $a$ , possibly in a different order).
22. Merge two sorted arrays into a single sorted array in ascending order.
23. Insert an element  $x$  into the array at position  $k$ .
24. Insert  $x$  into the sorted array while maintaining the ascending order.
25. Delete the elements at index  $k$  in the array.
26. Delete all elements in the array that are equal to  $x$ .
27. Remove all duplicate elements from the array, keeping only one occurrence of each value.
28. Left rotate the array by  $k$  positions.
29. Right rotate the array by  $k$  positions.
30. Create an array  $b$  that contains only the prime numbers from array  $a$ .

## 2.2 Correct score values

YY: 05

**a. (10 điểm)** Trong quá trình theo dõi điểm học tập, Tuấn đã phát hiện ra một tính chất thú vị rằng “điểm của mình **tăng dần đều** theo từng học kỳ”. Do đó, Tuấn đã thu thập điểm số của  $n$  học kỳ liên tiếp ( $n \geq 3$ ). Tuy nhiên, vì sơ sót, Tuấn đã ghi thiếu điểm của **chỉ một học kỳ duy nhất** và phải dùng giá trị điểm **-1** để thay thế cho học kỳ đó.

Hãy giúp Tuấn viết hàm khôi phục giá trị bị thiếu này, bằng cách thay giá trị **-1** trong mảng điểm của Tuấn thành giá trị điểm số tương ứng với tính chất của dãy điểm số. Nếu có thể khôi phục được giá trị này, hàm trả về **1** và giá trị tương ứng tìm được sẽ được cập nhật vào dãy điểm số của Tuấn. Nếu không thể khôi phục được giá trị này, hàm trả về **0** và dãy điểm số vẫn giữ nguyên như cũ.

**Ví dụ:** các dãy điểm số và kết quả trả về mong đợi từ hàm như sau:

- {30, -1, 70, 90}	→	1 - {30, 50, 70, 90}
- {-1, 20, 50, 80}	→	0 - {-1, 20, 50, 80}

**b. (15 điểm)** Không may rằng có những môn học mà dãy điểm số của Tuấn không thỏa tính chất “tăng dần đều” như trên. Hãy giúp Tuấn viết chương trình tìm ra dãy con (**có ít nhất 3 phần tử**) gồm các điểm số liên tiếp sao cho dãy con này là một dãy điểm số tăng dần đều **dài nhất**. Hàm sẽ trả về vị trí bắt đầu và kết thúc của dãy con tăng dần đều dài nhất này, biết rằng vị trí xét bắt đầu từ 0. Nếu có nhiều hơn một dãy con tăng dần đều dài nhất thì trả về thông tin của dãy con có vị trí bắt đầu lớn nhất.

**Ví dụ:** các dãy điểm số và kết quả trả về mong đợi từ hàm như sau:

▪ {0, 10, 20, 40, 60, 100} → start = 2, end = 4

**Giải thích:** Có 2 dãy con liên tiếp tăng dần dài nhất gồm {0, 10, 20} và {20, 40, 60}, nên giá trị start và end cần tìm là 2 và 4.

## 2.3 Assignment 3

YY: 05

**BÀI 1** Cho một mảng một chiều gồm  $n$  phần tử số nguyên, hãy viết các hàm thực hiện các yêu cầu sau:

- (a) (10 điểm) Kiểm tra xem mảng có chứa 3 giá trị chẵn hoặc 3 giá trị lẻ nằm cạnh nhau hay không. Hàm trả về giá trị 1 nếu thỏa điều kiện và trả về giá trị 0 nếu không thỏa.

Ví dụ:

[2, 1, 3, 5]: trả về 1

[2, 1, 2, 5]: trả về 0

[2, 4, 2, 5]: trả về 1

- (b) (10 điểm) Tính tổng của các số trong mảng, ngoại trừ các số nằm trong đoạn bắt đầu bởi số 6 và đến số 7 tiếp theo. Mảng đầu vào đảm bảo nếu có số 6 sẽ có ít nhất một số 7 theo sau.

Ví dụ:

[1, 2, 2]: trả về 5

[1, 2, 2, 6, 99, 99, 7]: trả về 5

[1, 1, 6, 7, 2]: trả về 4

- (c) (10 điểm) Kiểm tra xem có thể phân chia mảng thành hai phần sao cho tổng các số ở một bên bằng tổng các số ở bên còn lại hay không. Hàm trả về giá trị 1 nếu có thể thực hiện được việc phân chia, và trả về giá trị 0 nếu không thể. Ví dụ:

[1, 1, 1, 2, 1]: trả về 1 (do có thể tách ra thành: [1, 1, 1] và [2, 1])

[2, 1, 1, 2, 1]: trả về 0 (do không thể tách thành hai bên có tổng giá trị bằng nhau)

[10, 10]: trả về 1 do có thể tách ra thành: [10] và [10])

## 2.4 Assignment 4

YY: 05

**BÀI 2** Các sinh viên ngành Toán học trường K trong một lần học thực hành toán đã phát hiện ra dãy số thú vị T. Dãy là dãy số có một trong các tính chất sau đây:

- *Tính chất 1.* Dãy tăng nghiêm ngặt
- *Tính chất 2.* Dãy giảm nghiêm ngặt
- *Tính chất 3.* Dãy tăng nghiêm ngặt rồi giảm nghiêm ngặt

Ví dụ [1, 2, 3, 1] là dãy tăng nghiêm ngặt rồi giảm nghiêm ngặt (thỏa tính chất 3), còn dãy [1, 2, 2, 1] không phải là dãy tăng nghiêm ngặt rồi giảm nghiêm ngặt (không thỏa tính chất 3).

Bạn hãy hỗ trợ các sinh viên trên tìm dãy T dài nhất có trong một dãy số dài (gồm khá nhiều phần tử).

- (a) (10 điểm) Đề xuất thuật toán tìm dãy số T dài nhất trong một dãy  $a$  gồm có  $n$  phần tử nguyên dương. Thuật toán được thể hiện dạng ngôn ngữ tự nhiên, lưu đồ hay mã giả.
- (b) (10 điểm) Cài đặt hàm thực hiện yêu cầu trên dựa trên thuật toán đề xuất ở câu (a).

Ví dụ,

Dãy  $a$  gồm 6 phần tử {1, 2, 2, 3, 4, 1} có dãy T dài nhất gồm 4 phần tử là {2, 3, 4, 1}.

## 2.5 Assignment 5

YY: 05

**Problem 2.** Given an array **a** of **n** integer numbers, you are asked to implement the following functions:

(3.0pts)

- a. Print out all the numbers which are equal to the sum of all the numbers sitting before it in the array.

**void printNumsEqualSum(int a[], int n);**

E.g.: 12 7 6 25 48 98 -16 180 → Output: 25 98 180

- b. Remove all the prime numbers from the array:

**void removePrimeNums(int a[], int& n);**

- c. You are allowed to rotate the digits on the number, e.g. 123 could be rotated to be 231 or 312. Return the biggest sum of n numbers.

**long long biggestSumRotated(int a[], int n);**

E.g: Array: 215 -83 476

Biggest sum: 1247 = (521+ (-38) + 764)