

University of Mila
Faculty of Science and Technology
Department of Process Engineering

Practical Work 04

Course: Introduction to Programming

Level: 1st Year ST - ENG & LMD

Semester 02

By:

Dr. Farouk KECITA

Academic Year: 2025/2026

Exercise 01: Reading and Printing Array Elements

Question: Write a C program that reads n elements from two 1D arrays, $\text{arr_1}[n]$ and $\text{arr_2}[n]$, entered by the user. Then, print the elements of both arrays separately as $\text{arr_1}[i] = \text{value}$ and $\text{arr_2}[i] = \text{value}$ using a for loop.

Exercise 02: Merging Two Arrays

Question: Write a C program to read elements into two 1D arrays and merge them into a third array.

Steps to merge two 1D arrays in a C program:

1. Read the size of the first array and store it in the variable `size1`. Read the elements and store them in the first array `arr1[]` using `scanf()` and a for loop: `for(i = 0; i < size1; i++)`.
2. Read the size of the second array and store it in the variable `size2`. Read the elements and store them in the second array `arr2[]` using `scanf()` and a for loop: `for(j = 0; j < size2; j++)`.
3. Merge `arr1[]` and `arr2[]` into `arr3[]` using a for loop from `i = 0` to `i < size3`, where `size3 = size1 + size2`.
 - 1.A If `i < size1`, then `arr3[i] = arr1[i]`; otherwise, `arr3[i] = arr2[i - size1]`.
 - 2.B After the loop completes, the two arrays are merged into `arr3[]` with size `size3`.
4. Print `arr1`, `arr2`, and the merged array `arr3`.

Important note: To merge arrays in ascending or descending order, both arrays must already be sorted in the same order.

Exercise 03: Split Array into Even and Odd

Question: Write a C program to split a 1D array into two separate arrays based on even and odd values.

Steps to split a 1D array into even and odd arrays:

- Read the size of the original array and store it in the variable `n`.
- Declare three arrays: `arr` (original), `evenArr` (for even numbers), and `oddArr` (for odd numbers).
- Read the elements of the original array `arr` from the user.
- If `arr[i]` is even (`arr[i] % 2 == 0`), store it in `evenArr`.
- Otherwise, store it in `oddArr`.
- Display both `evenArr` and `oddArr`.

Exercise 04(Homework): Frequency of Elements

Question: Write a C program to count the frequency of each element in a 1D array.

Steps to find the frequency of each element in a 1D array:

1. Read the array size and store it in the variable `size`.
2. Read the elements and store them in the array `arr[]` using `scanf("%d", &arr[i])` and a for loop: `for(i = 0; i < size; i++)`.
3. Use a for loop from `i = 0` to `i < size`:
 - 3.A** If `arr[i] != -1`:
 - 3.A.1** Compare `arr[i]` with the remaining elements using a for loop from `j = i + 1` to `j < size`.
 - 3.A.2** If `arr[i] == arr[j]`, increment the count and set `arr[j] = -1` (mark as counted).
 - 3.B** Store the count in `freq[i]`. The array `freq` contains the frequency of each element.
4. Print each element along with its frequency using a for loop from `i = 0` to `i < size`:
`printf("Frequency of %d is %d\n", arr[i], freq[i]);`

Exercise 05: Array Sum and Average

Question: Write a C program to store integers entered by the user in a one-dimensional array, then compute the sum and average of the elements.

Steps to calculate the average of all elements in a 1D array:

- Read the array size and store it in the variable `S`.
- Declare a 1D array `arr` of size `S` and read its elements from the user.
- Print all elements as: `arr[i] = value`.
- Calculate the sum of all elements.
- Divide the sum by the number of elements to get the average.
- Store the result in the variable `avg` and print it.

Exercise 06: Dot Product of Two Vectors

Question: Write a C program to compute the dot product of two 1D arrays (vectors).

Steps to calculate the dot product of two vectors:

- Declare two 1D arrays, `vec1` and `vec2`, and initialize a variable `dotProduct` to 0.
- Verify that the lengths of `vec1` and `vec2` are equal.
- Use a for loop to compute the sum of products: `dotProduct += vec1[i] * vec2[i]`.
- Print the result stored in `dotProduct`.