Practical worksheet N°5. Arrays and Strings

1) Objectives

The objective of this lab sheet is to become familiar with the use of arrays and strings of characters in the C language. Arrays are data structures that allow to store multiple values of the same type in a single variable. The different elements in an array are accessible by their indices. The type “String” in the C language is an array of characters ending with the special character "**\0**".

At the end of this lab, the student should be able to declare, initialize, and manipulate strings and arrays with one or two dimensions in the C language.

2) Example 1: Arrays

Consider the following program:

```
#include <stdio.h>
int main() {
    #define n 4
    int T[n]; int i;
    printf("Please fill the array\n");
    for (i = 0; i < n; i++) {
        printf("T[%d] = ", i);
        scanf("%d", &T[i]);
    }
    printf("The array T contains the values :\n");
    for (i = 0; i < n; i++) {
        printf("%d\n", T[i]);
    }
    return 0;
}
```

1. Create a new project and type the above code.
2. Compile and run, what does this program do?
3. What is the nature of the variable **T**?
4. What is the difference between a variable of type **int** and an array of the same type?
5. How can you access each element in an array?
6. What will happen if we try to access an index beyond the size of the array?
7. Replace the expression « **i < n** » in both loops with « **i < 10** » and execute the code. Is there a problem? Explain what happens.
8. Change the program to calculate the sum of the elements in the array.

9. Make the necessary modifications for the program to declare a matrix with 4 rows and 3 columns (a two-dimensional array), read its elements, and calculate their sum.

3) Example 2 : Strings

Consider the following program:

```
#include <stdio.h>
int main() {
    #define n 10
    char s[n]; int i;
    printf("Enter a string of characters : ");
    gets(s);
    i = 0;
    while (s[i] != '\0') {
        i = i + 1;
    }
    printf("The string \"%s\" contains %d characters", s, i);
    return 0;
}
```

1. Create a new project and type the above code.
2. Compile and run, what does this program do?
3. What does the format « **`%s`** » mean in the « **`printf`** » statement?
4. Considering the previous code:
    a. What is the maximum length of the string that can be entered without encountering problems?
    b. Test the code with a string of length 10. Is there an issue? Explain.
    c. Test the code with a string length greater than 10. What do you observe?
5. Modify the code to count the number of spaces in the string entered by the user.

4) Application exercises

1. Write a program that allows the user to enter the elements of an array of size *n*, and then calculate the minimum and maximum of its elements. (**The algorithm was covered in lecture**).
2. Write a program that asks the user to fill an array *T* of size *n*, and then provide a value *x*. The program should then indicate whether *x* belongs to *T*. (**The algorithm was covered in lecture**).

    Modify the program to display the number of occurrences of *x*, and the index of its first and last occurrence if it belongs to *T*.
3. Write a program to read the elements of two vectors *U* and *V* of size *n*, and then calculate their dot product (scalar product) using the following formula:

$$U \cdot V = \sum_{i=0}^{n-1} U_i V_i$$

4. Write a program that reads the elements of an integer array of size *n*, then cyclically shifts its elements to the left by one position and displays the result.

   Example:
   - Array before shifting: {2, 7, 1, 6, 5}.
   - Array after left shift: {7, 1, 6, 5, 2}.

5. The infinity norm of a matrix is defined as the maximum of the sum of the absolute values of each row. Formally, we can define the infinity norm of a matrix *A* with *n* rows and *m* columns using this formula:

$$\|A\|_\infty = \max_{i \in [0..n-1]} \sum_{j=0}^{m-1} |A_{i,j}|$$

   Write a program to fill a matrix *A* with *n* rows and *m* columns, then calculate and display its infinity norm. (**The algorithm was covered in TW**).

6. Write a program that asks the user to enter the elements of an $n \times m$ matrix. Then, calculate and display the minimum and maximum values of the matrix, along with their positions in the matrix (row and column indices).
7. Write a program that reads a matrix with 3 rows and 2 columns and calculates and displays its transpose.

   The transpose of a matrix *A* with *n* rows and *m* columns, denoted as A^T, is a matrix with *m* rows and *n* columns, obtained by exchanging the rows and columns of *A*.
8. A word is called a "palindrome" if it reads the same from left to right or from right to left. Write a program that asks the user to provide a word and then determines whether the word is a palindrome.

5) Additional exercises

9. Given an array *T* of integers containing binary values (*T[i]* ∈ {0, 1}). Write a program that returns the position *i* such that *T[i]* is the start of the longest consecutive chain of zeros.

   Write a program that calculates the largest difference between elements in an array of real numbers, where the difference between two elements is defined by the absolute value of their subtraction.

10. A matrix of size (9×9) is called a "valid Sudoku" if:
    - All its rows contain all the digits from 1 to 9 without repetition.
    - All its columns contain all the digits from 1 to 9 without repetition.

- Each of its 9 sub-matrices of size (3×3) as shown in the image below contains all the digits from 1 to 9 without repetition.

Write a program that reads the elements of a (9×9) matrix and determines if it is a valid Sudoku or not.

Example of a valid Sudoku matrix :

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

11. Ecrire un programme qui permet de saisir deux chaînes de caractères S1 et S2, puis dit si S2 est une sous-chaîne de S1.
12. Write a program that asks the user to enter a string of characters, then displays the index of the first non-repeated character in the string.
13. Write a program that calculates the largest difference between elements in an array of real numbers, where the difference between two elements is defined by the absolute value of their subtraction.
14. Write a program that reads the elements of an array of integers, then moves all the zeros to the end of the array.

    Modify the program to keep the relative order of the elements after moving the zeros.
15. Write a program that asks the user to provide a sentence, then calculates and displays the length of the last word in the sentence.

    Note: We assume that the sentence contains only letters and spaces, no numbers, no special characters, and it ends with a letter.

    Execution example:

    ```
    Provide a sentence: Hello world

    The length of the last word is 5.
    ```
16. Write a program that allows the user to enter two strings of characters, *S1* and *S2*, then determines if *S2* is a substring of *S1*.