Exercice Array 1D

Exercise 1:

Write a program that calculates the scalar product of two integer vectors U and V (of the same dimension N, where N<50). Example: if the elements of U are 2,5,-1 and the elements of V are 3,0,10 then the result is equal to: 2*3+5*0+(-1)*10=-4

Exercise 2:

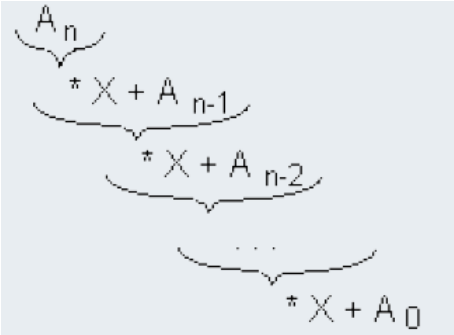
For a given float value X, the numeric value of a polynomial of degree n is given by:

$$P(X) = A_n X^n + A_{n-1} X^{n-1} + \cdots + A_1 X + A_0$$

Write a program that enter the values of *n and* the coefficients $A_n, A_{n-1}, \ldots, A_1, A_0$ which shall be stored in a float array A of dimension $n + 1$.

The program then reads the value of X and calculates the value of P(X) using the two methods:

1. Use the pow() function in the library math.h
2. Use the Horner scheme that avoids the exponentiation operations:



Exercise 3:

Write a program that determines the largest and the smallest value in an array of integers A[20]. Next, display the value, the number of appearance and the position of the maximum and minimum values. If the array contains several maxima or minima, the program will retain the position of the first maximum or minimum encountered.

Exercise 4:

Write a program that search if the value VAL entered on the keyboard is contained in an array A of 20 integers. The program will display the position of VAL if it is in the array A, otherwise the program will display the message "NOT FOUND". Hint: use a variable POS to memorize the position of the element equal to VAL and POS will have the value -1 as long as VAL is not in the array A.

Exercise 5:

i. Write a program that reads the dimension N of an integer array T (maximum dimension: 50 components), fills the array with values entered on the keyboard and displays the array.

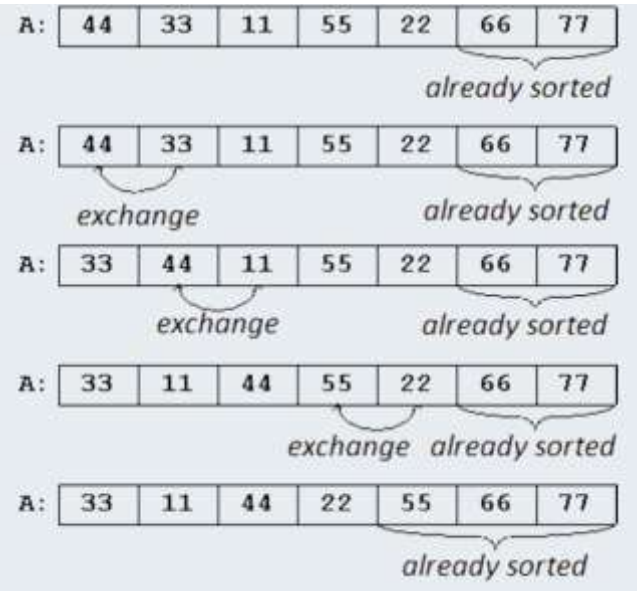
Then sort the elements of the array in ascending order. Finally, display the array after sorting it.

Using the following method:

Starting again each time at the beginning of the table, we carry out several times (size of the array) the following treatment: We propagate, by successive permutations, the largest element of the array towards the end of the array (like a bubble that goes back to the surface of a liquid).

Example: in the first time, we have the following changes.

We compare A[0] with A[1], then A[1] with A[2], then A[2] with A[3], ….A[N-2] with A[N]



Then we repeat N times, we get at the end a sorted array.

ii. Read a new value VAL, insert the value of VAL read from the keyboard in the array A in such a way the array remain sorted in ascending order.