University of BATNA 2
Faculty: Mathematics and Computer
Department: Common Core in Mathematics and Computer Science
1st Year CC-MCS 2023-2024 academic year

Series of 5th supervised exercises

Exercise 1

To calculate a student's average, write an algorithm using the following steps:

- **a.** Read and save 8 marks from 8 modules in a vector whose coefficients are (2,3,1,2,4,2,1,3).
- **b.** Calculate and display the student's average without taking module coefficients into account.
- **c.** Calculate and display of the student's average taking module coefficients into account.

Exercise 2

Rewrite the algorithm for Exercise 6 of Series 4 (Fibonacci sequence) using an array T.

Exercise 3

Let TAB be an array of N real numbers (N does not exceed 120).

1- Write an algorithm that finds the minimum and maximum values and their positions in the TAB array.

2- Write an algorithm that adds the value 15 to all even-numbered values and subtracts the value 20 from all odd-numbered values.

3- Write an algorithm that calculates the frequency (number of repetitions) of the first number in the array, as well as all its positions.
Example :

1.2	8.3	0	1.2	1.2	4.5	8	0	9	1.2	0.5	10	3	1.2	1.8

First element "1.2", its frequency = 5 (it exists 5 times in the table)
Its positions are : 1, 4, 5, 10, 14.

4- Write an algorithm that allows to copy TAB array into another array TAB2 with deletion duplicated elements.

Example

TAB

4.1	1.7	3	7	3	1.7	7	9.5

TAB2

4.1	1.7	3	7	9.5

5- Write an algorithm that splits TAB into two arrays, TAB_POS and TAB_NEG, containing positive and negative numbers respectively.

Exercise 4

Let V1and V2 be two vectors (one-dimensional arrays) of N integers for each (N≤150).
Write the algorithm that determines the sum and scalar product of V1 and V2.

The sum of V1 and V2 is V3 such that : V3[i]=V1[i]+V2[i]
The scalar product of V1 and V2= $\sum_{i=1}^{N}(V1[i] * V2[i])$.

Home exercises

Exercise 5

```
Algorithm Algo_1 ;
  Vari,N : integer ;
  Nb : array of [1..50] ofinteger;
Begin
  Repeat
    Read(N);
  Until ((N>0) et (N≤50)) ;

  For (ifrom 1 to  N ) do
    Nb[i]←i * i ;
  Endfor

  For (i from 1 to  N ) do
    Write(Nb[i]) ;
  Endfor
End.
```

```
Algorithm Algo_2 ;
  Vari,N,S : integer;
  Nb : array of [1..50] ofinteger;
Begin
  Repeat
    Read (N);
  Until ((N>0) et (N≤50)) ;

  S ← 0 ;
  For (i from 1 to  N ) do
    Nb[i]←i * i ;
    S←S +Nb[i] ;
  Endfor

  Write(S) ;
End.
```

```
Algorithm Algo_3 ;
  Vari,N : integer;
  T : array of [1..80] ofinteger;
Begin
  Repeat
    Read (N);
  Until ((N>0) et (N≤80)) ;
  T[1]←100 ;
  For (i from 2 to  N ) do
    T[i] ← T[i-1] + 2 ;
  Endfor

  For (i from 1 to  N ) do
     Write(T[i]) ;
  Endfor
End.
```

```
Algorithm Algo_4 ;
  Vari,N : integer;
  T : array of [1..150] of integer;
Begin
  Repeat
    Read (N);
  Until ((N>0) et (N≤150)) ;

  T[1]←8 ;
  For (i from 2 to  N ) do
    T[i] ← T[i-1] *3 ;
  Endfor

  For (i from 1 to  N ) do
     Write(T[i]) ;
  Endfor

End.
```

```
Algorithm Algo_5 ;
  Vari,N,S,P : integer;
  T : array of [1..70] ofinteger;
Begin
  Repeat
    Read (N);
  Until ((N>0) et (N≤70)) ;

  For (i from 1 to  N ) do
    Read (T[i]) ;
  Endfor
  S←0 ;
  P ←1 ;
  For (i from 1 to  N ) do
     If(T[i] mod 2 =0 )Then
        S ← S + T[i] ;
     Else
        P ← P*T[i] ;
     Endif
  Endfor
     Write(S,P) ;
End.
```

```
Algorithm Algo_6 ;
  Vari,N,j,k,X : integer;
  T1 : array of [1..40] ofinteger;
  T2 : array of [1..40] ofinteger;
Begin
  Repeat
    Read (N);
  Until ((N>0) et (N≤40)) ;
  j← 0 ;k ← 0 ;
  For (i from 1 to  N ) do
    Read (X) ;
     If(X mod 5 =0 ) Then
        j← j+1 ;
        T1[j]←X ;
     Else
        k← k+1 ;
        T2[k]←X ;
     Endif
  Endfor

  For (i from 1 to  j ) do
     Write(T1[i]) ;
  Endfor
  For (i from 1 to  k ) do
     Write(T2[i]) ;
  Endfor
End.
```

Consider the preceding algorithms:

1. Trace and say what each of the above algorithms does.
2. Choose one of the previous algorithms and rewrite its **Repeat** loop with the **while** loop.
3. Choose one of the above algorithms and rewrite its **For** loops as **while** loops.

Exercise 6

Let TAB be an array of N real numbers (N does not exceed 120).

1- Write an algorithm to check whether the array TAB is sorted or not.

2- Write an algorithm to sort the array TAB in descending order.

3- Write an algorithm to invert the elements of an array.

Example

TAB	5	1	8	2	2	3

Its inverse is:

3	2	2	8	1	5

4- Write an algorithm to insert a VAL value at the k^{th} position in the TAB array. Note: in this question (N<120).

5- Assume that the array TAB is sorted in ascending order and (N<120). Write an algorithm that inserts a real number X into this array while maintaining the sorted array.

Exercise 7

Let V1and V2 be two vectors (one-dimensional arrays) of N integers for each (N≤150).

1- Write an algorithm that constructs, from the previous vectors, two vectors MAX and MIN defined as follows:

- MAX[i] = maximum between V1[i] and V2[i].
- MIN[i] = minimum between V1[i] and V2[i].

2- Assume that V1 and V2 are sorted in ascending order:

- Write an algorithm that fuses these two vectors into a vector V3 sorted in descending order.
- Modify the previous algorithm to remove any duplicates.

Exercise 8

Let T be an array of characters containing a sentence ending in a point. Write a C program that determines :

- *The number of words in the sentence*
- *The longest word in the sentence*

Exercise 9

Let Tab be an array of N characters containing a single word.

1- Write a C program that determines the mirror word of a given word in the Tab array.

> *The "mirror" word of a given word is obtained by reading the word from the end to the beginning.*

Example :

Tab	***B***	***a***	***T***	***n***	***A***

Its mirror is:

A	***n***	***T***	***a***	***B***

2- *Write a C program that checks whether the word given in the Tab array is a palindrome.*

> *A word is said to be palindrome if it is identical to its mirror word.*