



## Tutorial N° 05 : Arrays and strings

1. Declare an array of 9 reals and initialize it with the value 0.
2. Write the algorithm that fills an array of 9 (real) notes with values requested from the user. Then display the 9 values contained in the table.
3. Write the algorithm that calculates the sum and the average of the values in an array.
4. Write the algorithm that allows you to check that two tables are identical (the elements of the two tables will be entered by the user).
5. Write the algorithm that allows you to display the number of occurrences of a given number X in an array T of N elements.
6. Write an algorithm that calculates the smurf of two arrays. To calculate the smurf, you must multiply each element of array 1 by each element of array 2, and add everything.

For example if we have: array 1:

4	8	3	7
---	---	---	---

Array 2: 3 6

5	17
---	----

The Smurf will be:  $(4*5)+(5*8)+(3*5)+(7*5)+(4*17)+(8*17)+(3*17)+(7*17) = 484$ .

7. Consider the algorithm below which allows you to manipulate an array containing 10 integers.

### Algorithm Exo7

**Var T : arrayof [10] integer ; i : integer ;**

**begin**

**for i <-- 0 to 9 do**

**read(T[i]) ;**

**if T[i] = 0 then write (i) ;**

**endif.**

**End for**

**end**

1. What does this algorithm do?

2. Using the following example:  $T = \{2, 0, -5, 3, 0, 4, -1, 0, 0, 15\}$ , what displayed on the screen?

3. Give the equivalent C program.

### 8. Fill in the empty fields:

#### Algorithm calcule

**Var** .. : array [100] of real ;

..... : real ;

..... : integer ;

**begin**

write ('this algorithm calculates the average of even numbers') ;

write ('Please fill in the array with real numbers) ;

for (i= 0 to i= 99 step 1 )do

read (T[i]) ;

end do

write (' test the parity of numbers and calculate their average') ;

S←0 ;

```

n←0 ;
for (i= 0 to 99 step 1 )do
  if (... mod 2 = 0 ) then
    S ← S+ ... ;
    n← n+1 ;
  endif ;
end for
moy ← ... / ...;
write (' the average of ‘’,....., ‘even numbers is :’, ...);
end .

```

9. Let M be a square matrix of size 5x5 containing integers. Write the declaration and initialization to 0 of such a data structure.

10. Write an algorithm that asks the user to fill a matrix M of size 5x5.

11. Write the algorithms which allow on a 2D matrix of size 5\*5:

- To calculate the sum of the elements of a line (the line number being passed as a parameter).

12. Write an algorithm that asks the user for a **sentence** and displays on the screen the **number of vowels** contained in that sentence.

### Additional Exercises

13. Write an algorithm that asks the user for a positive number less than 100. In addition, fills an array with all even numbers less than or equal this number.

**For example** if the user enter 15 , the algorithm fills the array with 0.2.4.6.8.10.12.14.

14. Let M be a square matrix of size 5x5 containing integers. Write the declaration and initialization to 0 of such a data structure.

- To calculate the sum of the elements of a column (the column number being passed as a parameter).

- To calculate the sum of the elements of the diagonal (provided the matrix is square).

15. Write an algorithm that asks the user for a **word** and displays the **number of letters** in that word on the screen.

16. What does the following code display if we enter the two variables chaine1 and chaine2 ?

```

#include <stdio.h>
#include <string.h>
main()
{ char chaine1[]="my name is ";
  char chaine2 [ ]="Mohamed Amine" ;
  gets(chaine1) ;
  scanf("%s",chaine2) ;
  printf("%s\n",strncat(chaine1,chaine2,3)) ;
  printf(strcpy(chaine1,chaine2)) ;
}

```

1. My name is Mohamed amine

Mohamed amine

2. My name is Mohamed

Mohamed amine

3. My name isMoh

Mohamed

4. My name isMoh

Mohamed amine

**Examples of solution:**

Exercise 02: #include <stdio.h>

```
int main() {
    float tab[9]; // array of 9 real (float) numbers
    int i;

    printf("Give 9 real numbers:\n");
    // Input loop
    for (i = 0; i < 9; i++) {
        scanf("%f", &tab[i]);
    }
    // Output loop
    printf("You entered:\n");
    for (i = 0; i < 9; i++) {
        printf("%.2f ", tab[i]);
    }
    printf("\n");
    return 0;
}
```

Exercise 04: #include <stdio.h>

```
int main() {
    int n, i;
    int same = 1; // flag variable: assume tables are identical

    printf("Enter the number of elements in the array: ");
    scanf("%d", &n);

    int tab1[n], tab2[n];

    // Read first array
    printf("Enter %d elements for the first array:\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &tab1[i]);
    }

    // Read second array
    printf("Enter %d elements for the second array:\n", n);
    for (i = 0; i < n; i++) {
        scanf("%d", &tab2[i]);
    }

    // Compare element by element
    for (i = 0; i < n; i++) {
        if (tab1[i] != tab2[i]) {
            same = 0; // found a difference
            break;
        }
    }
}
```

```

    }
}

// Display result
if (same == 1)
    printf("The two array are identical.\n");
else
    printf("The two array are NOT identical.\n");

return 0;
}

```

Exercise 12:

```

#include <stdio.h>
#include <string.h>

int main() {
    char sentence[200];
    int i, count = 0;

    printf("Enter a sentence:\n");
    fgets(sentence, sizeof(sentence), stdin); // read the whole line including spaces

    for (i = 0; sentence[i] != '\0'; i++) {
        char ch = sentence[i];
        // Check for vowels (both uppercase and lowercase)
        if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||
            ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U') {
            count++;
        }
    }

    printf("The number of vowels in the sentence is: %d\n", count);

    return 0;
}

```