

TD N°3

Boucles et tableaux

1. Afficher les nombres impairs

Ecrivez un programme qui lit au clavier un entier $n > 0$ et qui affiche à l'écran les n premiers entiers impairs. On utilisera les procédures suivantes :

```
Procédure Lire(S x : Entier)
Procédure Ecrire(E x : Entier)
```

Corrections :

```
Programme nombresImpairs ()
Déclaration
    i : Entier
    n : Entier
Début
    n ← 0
    Lire(n)
    Pour i ← 1 a 2*n-1 par pas de 2 faire
        Ecrire(i)
    FinPour
Fin
```

2. Moyenne d'une suite

Ecrivez un programme qui lit au clavier une suite x_0, x_1, x_2, \dots des nombres entiers positifs ou nuls et qui en affiche la moyenne. La frappe d'un nombre négatif indique la fin de la série. On utilisera les procédures de lecture et d'écriture de l'exercice 1.

Corrections :

```
Programme moyenneSuite()  
Déclaration  
    x : Entier  
    n : Entier  
    acc : Réel  
    moyenne : Réel  
Début  
    n ← 0  
    Lire(x)  
    Tant que (x ≥ 0) faire  
        acc ← acc + x  
        n ← n + 1  
        Lire(x)  
    FinTantQue  
  
    Si n > 0 alors  
        moyenne ← acc / n  
        Ecrire(moyenne)  
    FinSi  
Fin
```

3. Suite inverse

Ecrivez un programme qui lit au clavier une suite x_0, x_1, x_2, \dots de nombres entiers positifs ou nuls et qui les affiche dans l'ordre inverse de leur lecture. La frappe d'un nombre négatif indique la fin de la série. Nous avons des raisons de penser qu'il n'y aura pas plus de 100 nombres... On utilisera les procédures de lecture et d'écriture de l'exercice 1.

Corrections :

```

Programme afficheSuiteInverse ()
Déclaration
    suite : Tableau de 100 Entiers
    x : Entier
    n : Entier
    i : Entier
Début
    n ← 1
    Lire(x)
    Tantque (x ≥ 0) et (n ≤ 100) faire
        suite[n] ← x
        n ← n+1
        lire(x)
    FinTantQue

    Si (n > 100 ou (n = 1 et x < 0)) alors
        Ecrire(« erreur ! »)
    Sinon
        Pour i ← n jusqu'à 1 par pas de -1
            Ecrire(suite[i])
        FinPour
    FinSi
Fin

```

4. Recherche dans un tableau

Ecrivez un programme qui cherche et imprime la valeur et le rang du plus petit élément d'un tableau T de N nombres entiers. On utilisera les procédures de lecture et d'écriture de l'exercice 1.

```

Programme rechercheMin(E T : Tableau de N Entiers)
Déclaration
    i : Entier
    min : Entier
    imin : Entier

Début
    min ← T[1]
    imin ← 1
    Pour i ← 2 jusqu'à N par pas de 1
        Si T[i] < min alors
            min ← T[i]
            imin ← i
        FinSi
    FinPour
    Ecrire(min)
    Ecrire(i)
Fin

```

```
#include <stdio.h>
#define N 4

void rechercheMin(int* tab_pi){
    int i = 0;
    int min_i = 0;
    int imin_i = 0;

    min_i = tab_pi[0];
    imin_i = 0;

    for(i=1;i<N;i++){
        if(tab_pi[i] < min_i){
            min_i = tab_pi[i];
            imin_i = i;
        }
    }
    printf("La valeur du min est %d de rang %d\n",min_i, imin_i);
}

int main(){

    int tab_pi[N] = {3,2,1,4};
    rechercheMin(tab_pi);

    return 0;
}
```

5. Tassage de tableau

Etant donné un tableau T de N nombres entiers positifs ou nuls, écrivez le programme qui le tasse, c'est à dire qui détecte les éléments nuls du tableau et qui récupère leur place en décalant vers le début du tableau tous les autres éléments.

```
fonction tassage(E/S Tab : Tableau de N Entiers) : Entier
Déclaration
    Tab2 : Tableau de N Entiers
    i : Entier
    j : Entier
Début
    j ← 0
    Pour i ← 1 jusqu'à N par pas de 1
        Si Tab[i] ≠ 0 alors
            j ← j+1
            Tab2[j] ← Tab[i]
        FinSi
    FinPour

    Pour i ← 1 jusqu'à j par pas de 1
        Tab[i] ← Tab2[i]
    FinPour

    Retourner j
Fin
```