

Travaux Dirigés n°2

MAIN3

VOYLES Evan

October 13, 2021

Algorithm 1 Renvoi l'indice du plus petit élément d'un tableau d'entiers

Données d'entrée : T (tableau d'entiers), N (nombre entier, taille de T)

Données de sortie : indice_min (nombre entier)

```
1: indice_min ← -1
2: min ←  $T_1$ 
3: for  $i \leftarrow 1, N$  do
4:   if  $T_i < min$  then
5:      $min \leftarrow T_i$ 
6:      $indice\_min \leftarrow i$ 
7:   end if
8: end for
9: return indice_min
```

Algorithm 2 Nombre d'occurrences

Données d'entrée : T (tableau d'entiers), n (nombre entier), P (nombre entier, taille de T)

Données de sortie : nombre_fois (nombre entier)

```
1: nombre_fois ← 0
2: for  $i \leftarrow 1, P$  do
3:   if  $T_i = n$  then
4:      $nombre\_fois \leftarrow nombre\_fois + 1$ 
5:   end if
6: end for
7: return nombre_fois
```

Algorithm 3 Moyenne et ecart-type

Données d'entrée : T (tableau de doubles), N (nombre entier, taille de T)

Données de sortie : moyenne (double), ecart_type (double)

```
1: moyenne ← 0
2: ecart_type ← 0
3: sum ← 0
4: ecart_carre ← 0
5: for  $i \leftarrow 1, N$  do
6:    $sum \leftarrow sum + T_i$ 
7: end for
8:  $moyenne \leftarrow sum / N$ 
9: for  $i \leftarrow 1, N$  do
10:   $ecart\_carre \leftarrow (T_i - moyenne)^2$ 
11: end for
12:  $ecart\_type \leftarrow \sqrt{ecart\_carre / (N - 1)}$ 
13: return moyenne, ecart_type
```

Algorithm 4 Palindrome

Données d'entrée : T (tableau d'entiers), N (nombre entier, taille de T)**Données de sortie :** est_palindrome (boolean)

```
1: for  $i \leftarrow 1, \lfloor N/2 \rfloor$  do
2:   if  $T_i \neq T_{N-i+1}$  then
3:     return false
4:   end if
5: end for
6: return true
```

Algorithm 5 Remplacement des doublons

Données d'entrée : T (tableau d'entiers > 0), N (nombre entier, taille de T)**Données de sortie :** T avec les doublons remplacés par 0

```
1:  $old\_value \leftarrow T_1$ 
2: for  $i \leftarrow 2, N$  do
3:   if  $T_i = old\_value$  then
4:      $T_i \leftarrow 0$ 
5:   else
6:      $old\_value \leftarrow T_i$ 
7:   end if
8: end for
9: return T
```

Algorithm 7 Crible d'Ératosthène

Données d'entrée : T (tableau d'entiers > 0), N (nombre entier)**Données de sortie :** T avec les éléments non-primés remplacés par -1

```
1:  $T \leftarrow 2 : N$ 
2: for  $i \leftarrow 2, \lfloor \sqrt{N} \rfloor$  do
3:   for  $j \leftarrow 1, N$  do
4:     if  $T_j \bmod i \equiv 0$  then
5:        $T_j \leftarrow -1$ 
6:     end if
7:   end for
8: end for
9: return T
```

Travaux Dirigés n°3

Algorithm 1 Crible d'Ératosthène

Données d'entrée : T (tableau d'entiers > 0), N (nombre entier)

Données de sortie : T avec les éléments non-primés remplacés par -1

```

1:  $T \leftarrow 2 : N$ 
2: for  $i \leftarrow 2, \lfloor \sqrt{N} \rfloor$  do
3:   for  $j \leftarrow 1, N$  do
4:     if  $T_j \bmod i \equiv 0$  then
5:        $T_j \leftarrow -1$ 
6:     end if
7:   end for
8: end for
9: return T

```

Travaux Dirigés n°4

Exercice 3. On considère le programme suivant:

```

1 #include <stdio.h>
2 int a = 27;
3 int chose(int a);
4 int machin();
5
6 int chose(int a)
7 {
8     return a+17+machin(); // a est une variable locale
9 }
10 int machin()
11 {
12     return a; // a est une variable globale
13 }
14 int main()
15 {
16     int a=1; // nouvelle declaration de a cache le 'a' de ligne 2
17     a=chose(a); // a <- (1 + 17 + 27)
18     printf("%d\n", a);
19 }

```

Affiche 45 à l'écran.

Exercice 4. Définissez l'affichage produit par l'exécution du programme. Expliquez.

```

1 #include <stdio.h>
2
3 int nb = 3;
4 //=====
5
6 void mal_ecrit_1 (int nb) // nb est une variable locale
7 {
8     printf ("nb mal_ecrit_1 = %d\n",nb++); // nb++ renvoie nb et s'incrémente
9     APRES
10     {
11         int nb = 14; // cette exo est deguelasse
12         printf ("nb mal_ecrit_1 = %d\n",nb); // nb = 14
13     }
14     printf ("nb mal_ecrit_1 = %d\n",nb); // nb locale
15 }
16 //=====
17 int mal_ecrit_2 (int x, int y) // nb dans cette scope est globale
18 {
19     printf ("nb mal_ecrit_2 = %d\n",x + nb);
20     printf ("nb mal_ecrit_2 = %d\n",y + nb);
21     return (nb *= 0); // nb <- 0
22 }
23 //=====

```

```
24
25 int main(void)
26 {
27     int x = 3;
28     int y = 6;
29     printf ("nb main = %d\n",nb); // nb globale est initialise a 3
30     mal_ecrit_1(nb); // ne modifie pas du tout le nb globale
31     printf ("nb main = %d\n",nb);
32     printf("resultat de mal_ecrit_2 = %d\n",mal_ecrit_2(y,x)); // utilise nb = 3,
        renvoie nb <- 0
33     // Attention, on a appele la fonction mal_ecrit_2 avec l'ordre de y et x
        inverse
34     printf ("nb = %d\n",nb);
35     return 0;
36 }
```

```
29: nb main = 3
30: nb mal_ecrit_1 = 3
    nb mal_ecrit_1 = 14
    nb mal_ecrit_1 = 4
31: nb main = 3
32: nb mal_ecrit_2 = 9
    nb mal_ecrit_2 = 6
    resultat de mal_ecrit_2 = 0
34: nb = 0
```