

INF111 – TP4

Lucas Pastor

1 Exercice 5.4.3

```
1 #include <stdio.h>
2 #include <hasard.h>
3
4 /* Nombre maximum d'éléments à trier. */
5 #define NMAX 50
6
7 /* Plus petit entier que peut contenir le tableau. */
8 #define VALMIN -10
9
10 /* Plus grand entier que peut contenir le tableau. */
11 #define VALMAX 400
12
13 int main(int argc, char *argv[])
14 {
15     /* Le tableau. */
16     int tab[NMAX];
17
18     /* Le nombre d'éléments à trier. */
19     int n;
20
21     /* Indices de boucle. */
22     int i;
23     int j;
24
25     /*
26      La variable min contiendra l'indice
27      du plus petit entier qu'on trouvera en parcourant le
28      tableau.
29     */
30     int min;
31
32     /* Variable nécessaire pour l'échange. */
33     int z;
34
35     /* On récupère la donnée de l'utilisateur. */
36     printf("Entrez le nombre d'éléments à trier : ");
37     scanf("%d", &n);
38
39     /* On remplit le tableau. */
40     for (i = 0; i < n; i++)
41     {
42         tab[i] = hasard(VALMIN, VALMAX);
43     }
44
45     /* On affiche le tableau non trié. */
46     for (i = 0; i < n; i++)
47     {
48         printf("%d ", tab[i]);
49     }
50     /* On saute une ligne. */
51     printf("\n");
52
53     for (i = 0; i < n - 1; i++)
54     {
55
56         /*
57          On va rechercher l'indice du min à partir du premier
```

```

58     indice des éléments non triés.
59     */
60
61     /*
62     Par défaut, l'indice du min est le premier indice parmi les
63     éléments non triés.
64     */
65     min = i;
66
67     /*
68     Avec cette boucle, on trouve l'indice du min parmi les
69     éléments non triés du tableau.
70     */
71     for (j = i; j < n; j++)
72     {
73         /*
74         Si on trouve un plus petit élément dans le tableau
75         on change l'indice du min.
76         */
77         if (tab[j] < tab[min])
78         {
79             min = j;
80         }
81     }
82
83     /*
84     Quand on arrive ici, on a l'indice du min parmi
85     les éléments non triés.
86     On va donc échanger les cases d'indice i et min.
87     */
88     z = tab[min];
89     tab[min] = tab[i];
90     tab[i] = z;
91 }
92
93 /* On affiche le tableau trié. */
94 for (i = 0; i < n; i++)
95 {
96     printf("%d ", tab[i]);
97 }
98
99 /* On saute une ligne. */
100 printf("\n");
101
102 return 0;
103 }

```

2 Exercice 5.4.4

```

1 #include <stdio.h>
2 #include <hasard.h>
3 #include <graphsimple.h>
4
5 /* Nombre maximum d'éléments à trier. */
6 #define NMAX 50
7
8 /* Plus petit entier que peut contenir le tableau. */
9 #define VALMIN -10
10
11 /* Plus grand entier que peut contenir le tableau. */
12 #define VALMAX 400
13
14 int main(int argc, char *argv[])
15 {
16     /* Le tableau. */
17     int tab[NMAX];
18
19     /* Le nombre d'éléments à trier. */

```

```

20     int n;
21
22     /* Indices de boucle. */
23     int i;
24     int j;
25
26     /* Variable nécessaire pour l'échange. */
27     int z;
28
29     /*
30      La variable min contiendra l'indice
31      du plus petit entier qu'on trouvera en parcourant le
32      tableau.
33     */
34     int min;
35
36     /*
37      On trace la représentation graphique du tableau non trié.
38      Variables utiles au tracé graphique.
39     */
40
41     /* Distance entre le début du graphique et le haut de la fenêtre. */
42     int e = 30;
43
44     /* Distance entre le début du graphique et le côté gauche de la fenêtre. */
45     int a = 30;
46
47     /* Distance entre le côté gauche du graphique et le premier rectangle. */
48     int b = 5;
49
50     /* Distance entre deux rectangles. */
51     int c = 5;
52
53     /* Largeur d'un rectangle. */
54     int d = 5;
55
56     /* On récupère la donnée de l'utilisateur. */
57     printf("Entrez le nombre d'éléments à trier : ");
58     scanf("%d", &n);
59
60     /* On remplit le tableau. */
61     for (i = 0; i < n; i++)
62     {
63         tab[i] = hasard(VALMIN, VALMAX);
64     }
65
66     /* Création d'une fenêtre 600 x 500. */
67     Initialiser(600, 500);
68
69
70     /* On dessine l'axe des abscisses. */
71     Ligne(a, e, a + b + c * (n - 1) + d * n, e);
72
73     /* On dessine l'axe des ordonnées. */
74     Ligne(a, e, a, e + VALMAX - VALMIN);
75
76     for (i = 0; i < n; i++)
77     {
78         Rectangle(a + b + i * (d + c), e, a + b + i * (d + c) + d, e + tab[i] - VALMIN);
79     }
80
81     /* On attend un clic de souris. */
82     AttendreClic();
83
84     for (i = 0; i < n - 1; i++)
85     {
86
87         /*
88          On va rechercher l'indice du min à partir du premier
89          indice des éléments non triés.
90         */

```

```

91
92     /*
93     Par défaut, l'indice du min est le premier indice parmi les
94     éléments non triés.
95     */
96     min = i;
97
98     /*
99     Avec cette boucle, on trouve l'indice du min parmi les
100    éléments non triés du tableau.
101    */
102    for (j = i; j < n; j++)
103    {
104        /*
105        Si on trouve un plus petit élément dans le tableau
106        on change l'indice du min.
107        */
108        if (tab[j] < tab[min])
109        {
110            min = j;
111        }
112    }
113
114    /*
115    Quand on arrive ici, on a l'indice du min parmi
116    les éléments non triés.
117    On va donc échanger les cases d'indice i et min.
118    */
119
120    z = tab[min];
121    tab[min] = tab[i];
122    tab[i] = z;
123 }
124
125 /* On efface les dessins précédents. */
126 Effacer();
127
128
129 /* On dessine l'axe des abscisses. */
130 Ligne(a, e, a + b + c * (n - 1) + d * n, e);
131
132 /* On dessine l'axe des ordonnées. */
133 Ligne(a, e, a, e + VALMAX - VALMIN);
134
135 for (i = 0; i < n; i++)
136 {
137     Rectangle(a + b + i * (d + c), e, a + b + i * (d + c) + d, e + tab[i] - VALMIN);
138 }
139
140 AttendreClic();
141
142 /* Fermeture de la fenêtre. */
143 Clore();
144
145 return 0;
146 }

```

```

1 #include <stdio.h>
2 #include <hasard.h>
3 #include <graphsimple.h>
4
5 /* Nombre maximum d'éléments à trier. */
6 #define NMAX 50
7
8 /* Plus petit entier que peut contenir le tableau. */
9 #define VALMIN -10
10
11 /* Plus grand entier que peut contenir le tableau. */
12 #define VALMAX 400
13
14 int main(int argc, char *argv[])

```

```

15 {
16     /* Le tableau. */
17     int tab[NMAX];
18
19     /* Le nombre d'éléments à trier. */
20     int n;
21
22     /* Indices de boucle. */
23     int i;
24     int j;
25
26     /* Variable nécessaire pour l'échange. */
27     int z;
28
29     /*
30      La variable min contiendra l'indice
31      du plus petit entier qu'on trouvera en parcourant le
32      tableau.
33     */
34     int min;
35
36     /*
37      On trace la représentation graphique du tableau non trié.
38      Variables utiles au tracé graphique.
39     */
40
41     /* Distance entre le début du graphique et le haut de la fenêtre. */
42     int e = 30;
43
44     /* Distance entre le début du graphique et le côté gauche de la fenêtre. */
45     int a = 30;
46
47     /* Distance entre le côté gauche du graphique et le premier rectangle. */
48     int b = 5;
49
50     /* Distance entre deux rectangles. */
51     int c = 5;
52
53     /* Largeur d'un rectangle. */
54     int d = 5;
55
56     /* On récupère la donnée de l'utilisateur. */
57     printf("Entrez le nombre d'éléments à trier : ");
58     scanf("%d", &n);
59
60     /* On remplit le tableau. */
61     for (i = 0; i < n; i++)
62     {
63         tab[i] = hasard(VALMIN, VALMAX);
64     }
65
66     /* Création d'une fenêtre 600 x 500. */
67     Initialiser(600, 500);
68
69
70     /* On dessine l'axe des abscisses. */
71     Ligne(a, e, a + b + c * (n - 1) + d * n, e);
72
73     /* On dessine l'axe des ordonnées. */
74     Ligne(a, e, a, e + VALMAX - VALMIN);
75
76     /* On va dessiner le tableau non trié. */
77     for (i = 0; i < n; i++)
78     {
79         Rectangle(a + b + i * (d + c), e, a + b + i * (d + c) + d, e + tab[i] - VALMIN);
80     }
81
82     /* On attend un clic de souris. */
83     AttendreClic();
84
85     for (i = 0; i < n - 1; i++)

```

```

86  {
87
88      /*
89      On va rechercher l'indice du min dans à partir du premier
90      indice des éléments non triés.
91      */
92
93      /*
94      Par défaut, l'indice du min est le premier indice parmi les
95      éléments non triés.
96      */
97      min = i;
98
99      /*
100     Avec cette boucle, on trouve l'indice du min parmi les
101     éléments non triés du tableau.
102     */
103     for (j = i; j < n; j++)
104     {
105         /*
106         Si on trouve un plus petit élément dans le tableau
107         on change l'indice du min.
108         */
109         if (tab[j] < tab[min])
110         {
111             min = j;
112         }
113     }
114
115     /*
116     Quand on arrive ici, on a l'indice du min parmi
117     les éléments non triés.
118     */
119
120     /* On supprime le rectangle du min trouvé. */
121     SuppRectangle(a + b + min * (d + c), e, a + b + min * (d + c) + d, e + tab[min] - VALMIN);
122
123     /* On supprime le rectangle d'indice i. */
124     SuppRectangle(a + b + i * (d + c), e, a + b + i * (d + c) + d, e + tab[i] - VALMIN);
125
126     /* Puis on créer le rectangle du min à la place du rectangle i et inversement. */
127     Rectangle(a + b + i * (d + c), e, a + b + i * (d + c) + d, e + tab[min] - VALMIN);
128     Rectangle(a + b + min * (d + c), e, a + b + min * (d + c) + d, e + tab[i] - VALMIN);
129
130     /* On va maintenant échanger les cases d'indice i et min. */
131     z = tab[min];
132     tab[min] = tab[i];
133     tab[i] = z;
134
135     /* On attend un peu. */
136     AttendreDelai(100);
137 }
138
139 AttendreClic();
140
141 /* Fermeture de la fenêtre. */
142 Clore();
143
144 return 0;
145 }

```