

INF111 – TP3

Lucas Pastor

1 Exercice 5.3.1

```
1 #include <stdio.h>
2 #include <hasard.h>
3
4 /* Valeur maximum. */
5 #define MAXVAL 1000
6
7 /* Valeur minimum. */
8 #define MINVAL 1
9
10 int main(int argc, char *argv[])
11 {
12     /* Le nombre choisi au hasard. */
13     int n;
14
15     /* Compteur du nombre d'essais. */
16     int nbEssais;
17     nbEssais = 1;
18
19     /* Entier choisi par l'utilisateur. */
20     int x;
21
22     printf("Tirage au sort imminent!\n");
23
24     /* On tire le nombre n au hasard. */
25     n = hasard(MINVAL, MAXVAL);
26
27     /* On demande à l'utilisateur de choisir un nombre. */
28     printf("Choisissez un nombre entre %d et %d : ", MINVAL, MAXVAL);
29     scanf("%d", &x);
30
31     /*
32      * Tant que l'entier choisi par l'utilisateur est différent
33      * de l'entier tiré au hasard, on continue.
34      */
35     while (x != n)
36     {
37         /* Si x est plus petit que n. */
38         if (x < n)
39         {
40             printf("Trop petit.\n");
41         }
42         /* Sinon, c'est que x est plus grand que n. */
43         else
44         {
45             printf("Trop grand.\n");
46         }
47
48         /* On demande à l'utilisateur de choisir un nombre. */
49         printf("Choisissez un nombre entre %d et %d : ", MINVAL, MAXVAL);
50         scanf("%d", &x);
51
52         nbEssais++;
53     }
54
55     /* Si on arrive ici c'est que le joueur a trouvé. */
56     printf("Bravo! Vous avez gagné en %d essais.\n", nbEssais);
57 }
```

```

58
59     return 0;
60 }

```

2 Exercice 5.3.2

```

1 #include <stdio.h>
2 #include <hasard.h>
3
4 /* Valeur maximum. */
5 #define MAXVAL 1000
6
7 /* Valeur minimum. */
8 #define MINVAL 1
9
10 int main(int argc, char *argv[])
11 {
12     /* Le nombre choisi au hasard. */
13     int n;
14
15     /* Compteur du nombre d'essais. */
16     int nbEssais;
17
18     /* Nombre d'essais maximum. */
19     int nbEssaisMax;
20
21     /* Compteur du nombre de parties. */
22     int nbParties = 1;
23
24     /* Permet à l'utilisateur de choisir de continuer ou non. */
25     char c = '0';
26
27     /* Entier choisi par l'utilisateur. */
28     int x;
29
30     while(c == '0')
31     {
32         /* On réinitialise le nombre d'essais. */
33         nbEssais = 1;
34
35         printf("Tirage au sort imminent pour la partie %d!\n", nbParties);
36
37         /* On tire le nombre n au hasard. */
38         n = hasard(MINVAL, MAXVAL);
39
40         printf("Choisissez un nombre d'essais maximum : ");
41         scanf("%d", &nbEssaisMax);
42
43         /* On demande à l'utilisateur de choisir un nombre. */
44         printf("Choisissez un nombre entre %d et %d : ", MINVAL, MAXVAL);
45         scanf("%d", &x);
46
47         /*
48          * Tant que l'entier choisi par l'utilisateur est différent
49          * de l'entier tiré au hasard, on continue ET tant
50          * que le nombre d'essais maximum n'est pas dépassé.
51          */
52         while (x != n && nbEssais < nbEssaisMax)
53         {
54             /* Si x est plus petit que n. */
55             if (x < n)
56             {
57                 printf("Trop petit.\n");
58             }
59             /* Sinon, c'est que x est plus grand que n. */
60             else
61             {
62                 printf("Trop grand.\n");

```

```

63     }
64
65
66     /* On demande à l'utilisateur de choisir un nombre. */
67     printf("Choisissez un nombre entre %d et %d : ", MINVAL, MAXVAL);
68     scanf("%d", &x);
69
70     nbEssais++;
71 }
72
73 /*
74 Si on arrive ici c'est que la boucle while est terminée.
75 Si le nombre x est égal au nombre n, il a gagné.
76 */
77 if (x == n)
78 {
79     printf("Bravo! Vous avez gagné en %d essais.\n", nbEssais);
80 }
81 /* Sinon, c'est que son nombre d'essais est épuisé. */
82 else
83 {
84     printf("Perdu. Vous avez épuisé le nombre d'essais autorisés.\n");
85     printf("Le prix à trouver était : %d\n", n);
86 }
87
88 /* On demande l'utilisateur si il souhaite rejouer. */
89 printf("Voulez-vous rejouer (O/N) ? : ");
90 scanf(" %c", &c);
91
92 /* On incrémente le nombre de parties. */
93 nbParties++;
94 }
95
96 return 0;
97 }

```

3 Exercice 5.3.3

```

1 #include <stdio.h>
2 #include <hasard.h>
3
4 /* Valeur maximum. */
5 #define MAXVAL 1000
6
7 /* Valeur minimum. */
8 #define MINVAL 1
9
10 int main(int argc, char *argv[])
11 {
12     /* Le nombre choisi au hasard. */
13     int n;
14
15     /* Compteur du nombre d'essais. */
16     int nbEssais;
17
18     /* Compteur du nombre de parties. */
19     int nbParties = 1;
20
21     /* Nombre de parties à effectuer. */
22     int nbPartiesMax = 2000;
23
24     /* Nombre total d'essais. */
25     int nbTotalEssais = 0;
26
27     /* Entier choisi par l'IA (Intelligence Artificielle). */
28     int x;
29
30     /* Sauvegarde de l'entier x choisi précédemment. */

```

```

31     int xSave;
32
33     /* Borne inférieure de l'intervalle (fermé). */
34     int infInterval = 1;
35
36     /* Borne supérieure de l'intervalle (fermé). */
37     int supInterval = 1000;
38
39     while(nbParties <= nbPartiesMax)
40     {
41         /* On réinitialise le nombre d'essais. */
42         nbEssais = 1;
43
44         printf("Tirage au sort imminent pour la partie %d!\n", nbParties);
45
46         /* On tire le nombre n au hasard. */
47         n = hasard(MINVAL, MAXVAL);
48
49         /*
50          * On remet infInterval et supInterval à
51          * leur valeur par défaut.
52          */
53         infInterval = 1;
54         supInterval = 1000;
55
56         /*
57          * Ici c'est l'IA qui va choisir un nombre.
58          * Ce nombre est la moyenne entre infInterval et supInterval.
59          */
60         x = (infInterval + supInterval) / 2;
61         xSave = x;
62
63         /*
64          * On affiche le nombre choisi.
65          */
66         printf("Le nombre choisi est %d.\n", x);
67
68         /*
69          * Tant que l'entier choisi par l'IA est différent
70          * de l'entier tiré au hasard, on continue.
71          */
72         while (x != n)
73         {
74             /* Si x est plus petit que n. */
75             if (x < n)
76             {
77                 /*printf("Trop petit.\n"); */
78
79                 /*
80                  * Si le nombre est trop petit,
81                  * la borne inférieure est maintenant égale
82                  * à notre nombre.
83                  */
84                 infInterval = x;
85             }
86             /* Sinon, c'est que x est plus grand que n. */
87             else
88             {
89                 /* printf("Trop grand.\n"); */
90
91                 /*
92                  * Si le nombre est trop grand,
93                  * la borne supérieure est maintenant égale
94                  * à notre nombre.
95                  */
96                 supInterval = x;
97             }
98
99             /*
100            * Ici c'est l'IA qui va choisir un nombre.
101            * Ce nombre est la moyenne entre infInterval et supInterval.

```

```

102     */
103     x = (infInterval + supInterval) / 2;
104
105     /*
106     Si l'entier x choisi est le même que précédemment,
107     on est bloqué. On incrémente alors x de 1 car la division entière
108     arrondi à l'inférieur.
109     */
110     if (x == xSave)
111     {
112         x++;
113     }
114
115     /* On sauvegarde l'entier x. */
116     xSave = x;
117
118     /*
119     On affiche le nombre choisi.
120     printf("Le nombre choisi est %d.\n", x);
121     printf("Le nombre à trouver est %d.\n", n);
122     */
123
124     nbEssais++;
125 }
126
127 /* printf("Bravo! Vous avez gagné en %d essais.\n", nbEssais); */
128
129 /* On met à jour le compteur de nombre d'essais au total. */
130 nbTotalEssais = nbTotalEssais + nbEssais;
131
132 /* On incrémente le nombre de parties. */
133 nbParties++;
134 }
135
136 /* On affiche la moyenne. */
137 printf("En moyenne, l'IA trouve en %d essais.\n", nbTotalEssais/nbPartiesMax);
138
139 return 0;
140 }

```