

# INF111 – TP2

Lucas Pastor

## 1 Exercice 5.2.1

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     /* Les entiers. */
6     int a, b, c;
7
8     /* Le nombre total de bonnes réponses. */
9     int nbReps = 0;
10
11    /* Le caractère contenant la réponse de l'utilisateur. */
12    char rep;
13
14    /* On demande les entiers à l'utilisateur. */
15    printf("Entrez trois entiers :\n");
16
17    printf("a : ");
18    scanf("%d", &a);
19
20    printf("b : ");
21    scanf("%d", &b);
22
23    printf("c : ");
24    scanf("%d", &c);
25
26    /* On lui indique le déroulement de l'exercice. */
27    printf("Répondre par V (vrai) ou F (faux) : \n");
28
29    /******
30    /* Question 1. a < b < c? */
31    /******
32    printf("Q1. %d < %d < %d? : ", a, b, c);
33    scanf(" %c", &rep);
34
35    /* Si les entiers sont bien triés. */
36    if (a < b && b < c)
37    {
38        /* Si il répond vrai, la réponse est correcte. */
39        if (rep == 'V')
40        {
41            printf("réponse correcte, bravo!\n");
42            nbReps = nbReps + 1;
43        }
44        /* Sinon, c'est qu'il a répondu faux, la réponse n'est pas correcte. */
45        else
46        {
47            printf("réponse incorrecte.\n");
48        }
49    }
50    /* Sinon, c'est que les entiers ne sont pas bien triés. */
51    else
52    {
53        /*
54         * Si il a répondu faux, la réponse est correcte
55         * (car les entiers ne sont effectivement pas bien triés).
56         */
57        if (rep == 'F')
```

```

58     {
59         printf("réponse correcte, bravo!\n");
60         nbReps = nbReps + 1;
61     }
62     /*
63         Sinon, c'est qu'il a répondu vrai. Or les entiers ne sont
64         pas bien triés. La réponse est donc incorrecte.
65     */
66     else
67     {
68         printf("réponse incorrecte.\n");
69     }
70 }
71
72 /* Un petit saut de ligne avant de passer à la question suivante. */
73 printf("\n");
74
75 /******
76 /* Question 2. Un seul nombre impair parmi a, b, c? */
77 /******
78
79 /* On prend la réponse de l'utilisateur. */
80 printf("Q2. Un seul nombre impair parmi %d, %d, %d? : ", a, b, c);
81 scanf(" %c", &rep);
82
83 /*
84     Pour vérifier combien il y a de nombre impairs,
85     on utilise une variable qui va nous permettre de compter
86     combien il y en a.
87     On initialise cette variable à 0.
88 */
89 int nbImpairs = 0;
90
91 /* On vérifie si a est impair. */
92 if (a % 2 == 1)
93 {
94     /* Si c'est le cas, on incrémente nbImpairs de 1. */
95     nbImpairs = nbImpairs + 1;
96 }
97
98 /* On vérifie si b est impair. */
99 if (b % 2 == 1)
100 {
101     /* Si c'est le cas, on incrémente nbImpairs de 1. */
102     nbImpairs = nbImpairs + 1;
103 }
104
105 /* On vérifie si c est impair. */
106 if (c % 2 == 1)
107 {
108     /* Si c'est le cas, on incrémente nbImpairs de 1. */
109     nbImpairs = nbImpairs + 1;
110 }
111
112 /*
113     Si l'utilisateur répond vrai et qu'il n'y a qu'un seul nombre impair,
114     il a une bonne réponse.
115 */
116 if (rep == 'V' && nbImpairs == 1)
117 {
118     printf("réponse correcte, bravo!\n");
119     nbReps = nbReps + 1;
120 }
121 /*
122     Sinon, si l'utilisateur répond faux et qu'il n'y a pas qu'un seul nombre impair,
123     il a une bonne réponse.
124 */
125 else if (rep == 'F' && nbImpairs != 1)
126 {
127     printf("réponse correcte, bravo!\n");
128     nbReps = nbReps + 1;

```

```

129     }
130     /*
131     Sinon, c'est que l'utilisateur a répondu faux et qu'il n'y a qu'un seul
132     nombre impair ou que l'utilisateur a répondu vrai et qu'il n'y a pas qu'un
133     seul nombre impair. C'est donc une mauvaise réponse.
134     */
135     else
136     {
137         printf("réponse incorrecte.\n");
138     }
139
140     /* Un petit saut de ligne avant de passer à la question suivante. */
141     printf("\n");
142
143     /******
144     /* Question 3. a, b, c distincts deux à deux? */
145     /******
146
147     /* On prend la réponse de l'utilisateur. */
148     printf("Q3. %d, %d, %d distincts deux à deux? : ", a, b, c);
149     scanf(" %c", &rep);
150
151     /*
152     Si l'utilisateur répond vrai et que tous les nombres sont différents OU
153     si l'utilisateur répond faux et que tous les nombres ne sont pas différents,
154     il a une bonne réponse.
155     */
156     if ((rep == 'V' && (a != b && b != c && a != c)) ||
157         (rep == 'F' && !(a != b && b != c && a != c)))
158     {
159         printf("réponse correcte, bravo!\n");
160         nbReps = nbReps + 1;
161     }
162     /* Sinon, c'est ni l'un ni l'autre, il a donc une mauvaise réponse. */
163     else
164     {
165         printf("réponse incorrecte.\n");
166     }
167
168     /* Saut de ligne pour faciliter la lecture de l'affichage. */
169     printf("\n");
170     printf("%d bonnes réponses sur 3.\n", nbReps);
171
172     return 0;
173 }

```

## 2 Exercice 5.2.2

```

1 #include <stdio.h>
2
3 int main(void)
4 {
5     /* Les entiers. */
6     int a, b, c;
7
8     /* Le nombre total de bonnes réponses. */
9     int nbReps = 0;
10
11     /* Le caractère contenant la réponse de l'utilisateur. */
12     char rep;
13
14     /* On demande les entiers à l'utilisateur. */
15     printf("Entrez trois entiers :\n");
16
17     printf("a : ");
18     scanf("%d", &a);
19
20     printf("b : ");

```

```

21 scanf("%d", &b);
22
23 printf("c : ");
24 scanf("%d", &c);
25
26 /* On lui indique le déroulement de l'exercice. */
27 printf("Répondre par V (vrai) ou F (faux) : \n");
28
29 /******
30 /* Question 1. a < b < c? */
31 /******
32 printf("Q1. %d < %d < %d? : ", a, b, c);
33 scanf(" %c", &rep);
34
35 /* Si les entiers sont bien triés. */
36 if (a < b && b < c)
37 {
38     /* Si il répond vrai, la réponse est correcte. */
39     if (rep == 'V')
40     {
41         printf("réponse correcte, bravo!\n");
42         nbReps = nbReps + 1;
43     }
44     /* Sinon, c'est qu'il a répondu faux, la réponse n'est pas correcte. */
45     else
46     {
47         printf("réponse incorrecte.\n");
48     }
49 }
50 /* Sinon, c'est que les entiers ne sont pas bien triés. */
51 else
52 {
53     /*
54     Si il a répondu faux, la réponse est correcte
55     (car les entiers ne sont effectivement pas bien triés).
56     */
57     if (rep == 'F')
58     {
59         printf("réponse correcte, bravo!\n");
60         nbReps = nbReps + 1;
61     }
62     /*
63     Sinon, c'est qu'il a répondu vrai. Or les entiers ne sont
64     pas bien triés. La réponse est donc incorrecte.
65     */
66     else
67     {
68         printf("réponse incorrecte.\n");
69     }
70 }
71
72 /* Un petit saut de ligne avant de passer à la question suivante. */
73 printf("\n");
74
75 /******
76 /* Question 2. Un seul nombre impair parmi a, b, c? */
77 /******
78
79 /* On prend la réponse de l'utilisateur. */
80 printf("Q2. Un seul nombre impair parmi %d, %d, %d? : ", a, b, c);
81 scanf(" %c", &rep);
82
83 /*
84 Pour vérifier combien il y a de nombre impairs,
85 on utilise une variable qui va nous permettre de compter
86 combien il y en a.
87 On initialise cette variable à 0.
88 */
89 int nbImpairs = 0;
90
91 /* On vérifie si a est impair. */

```

```

92  if (a % 2 == 1)
93  {
94      /* Si c'est le cas, on incrémente nbImpairs de 1. */
95      nbImpairs = nbImpairs + 1;
96  }
97
98  /* On vérifie si b est impair. */
99  if (b % 2 == 1)
100 {
101     /* Si c'est le cas, on incrémente nbImpairs de 1. */
102     nbImpairs = nbImpairs + 1;
103 }
104
105 /* On vérifie si c est impair. */
106 if (c % 2 == 1)
107 {
108     /* Si c'est le cas, on incrémente nbImpairs de 1. */
109     nbImpairs = nbImpairs + 1;
110 }
111
112 /*
113  Si l'utilisateur répond vrai et qu'il n'y a qu'un seul nombre impair,
114  il a une bonne réponse.
115  */
116 if (rep == 'V' && nbImpairs == 1)
117 {
118     printf("réponse correcte, bravo!\n");
119     nbReps = nbReps + 1;
120 }
121 /*
122  Sinon, si l'utilisateur répond faux et qu'il n'y a pas qu'un seul nombre impair,
123  il a une bonne réponse.
124  */
125 else if (rep == 'F' && nbImpairs != 1)
126 {
127     printf("réponse correcte, bravo!\n");
128     nbReps = nbReps + 1;
129 }
130 /*
131  Sinon, c'est que l'utilisateur a répondu faux et qu'il n'y a qu'un seul
132  nombre impair ou que l'utilisateur a répondu vrai et qu'il n'y a pas qu'un
133  seul nombre impair. C'est donc une mauvaise réponse.
134  */
135 else
136 {
137     printf("réponse incorrecte.\n");
138 }
139
140 /* Un petit saut de ligne avant de passer à la question suivante. */
141 printf("\n");
142
143 /******
144 /* Question 3. a, b, c distincts deux à deux? */
145 /******
146
147 /* On prend la réponse de l'utilisateur. */
148 printf("Q3. %d, %d, %d distincts deux à deux? : ", a, b, c);
149 scanf(" %c", &rep);
150
151 /*
152  Si l'utilisateur répond vrai et que tous les nombres sont différents OU
153  si l'utilisateur répond faux et que tous les nombres ne sont pas différents,
154  il a une bonne réponse.
155  */
156 if ((rep == 'V' && (a != b && b != c && a != c)) ||
157     (rep == 'F' && !(a != b && b != c && a != c)))
158 {
159     printf("réponse correcte, bravo!\n");
160     nbReps = nbReps + 1;
161 }
162 /* Sinon, c'est ni l'un ni l'autre, il a donc une mauvaise réponse. */

```

```

163     else
164     {
165         printf("réponse incorrecte.\n");
166     }
167
168     /* Un petit saut de ligne avant de passer à la question suivante. */
169     printf("\n");
170
171     /******
172     /* Question 4. Quelle est la valeur de a * b + b * c + c * a? */
173     /******
174
175     /* Entier servant à récupérer le résultat de l'utilisateur. */
176     int resultat;
177
178     /* On affiche la question et on prend la réponse de l'utilisateur. */
179     printf("Q4. Quelle est la valeur de %d * %d + %d * %d + %d * %d?\n",
180         a, b, b, c, c, a);
181
182     scanf("%d", &resultat);
183
184     /* Si le résultat est correcte, bonne réponse. */
185     if (resultat == a * b + b * c + c * a)
186     {
187         printf("réponse correcte, bravo!\n");
188         nbReps = nbReps + 1;
189     }
190     /* Sinon, mauvaise réponse. */
191     else
192     {
193         printf("réponse incorrecte.\n");
194     }
195
196     /* Un petit saut de ligne avant de passer à la question suivante. */
197     printf("\n");
198
199     /******
200     /* Question 5. Existe-t-il un triangle dont les côtés sont a, b, c? */
201     /******
202
203     /*
204     Pour qu'un triangle de côtés a, b, c existe il faut que
205     la somme de deux côtés soit plus grande ou égale au troisième côté.
206     Donc on doit avoir: a + b >= c, a + c >= b, b + c >= a.
207     */
208
209     /* Affichage de la question et obtention de la réponse de l'utilisateur. */
210     printf("Q5. Existe-t-il un triangle dont les côtés sont %d, %d, %d?\n", a, b, c);
211     scanf(" %c", &rep);
212
213     /*
214     Si l'utilisateur répond vrai et qu'on respecte les inéquations des côtés,
215     c'est une bonne réponse. Ou si l'utilisateur répond faux et qu'on ne respecte
216     pas les inéquations des côtés, c'est une bonne réponse.
217     */
218     if ((rep == 'V' && (a + b >= c && a + c >= b && b + c >= a)) ||
219         (rep == 'F' && !(a + b >= c && a + c >= b && b + c >= a)))
220     {
221         printf("réponse correcte, bravo!\n");
222         nbReps = nbReps + 1;
223     }
224     /* Sinon, c'est ni l'un ni l'autre, c'est donc une mauvaise réponse. */
225     else
226     {
227         printf("réponse incorrecte.\n");
228     }
229
230
231     /* Un petit saut de ligne avant de passer à la question suivante. */
232     printf("\n");
233

```

```

234 /*****
235 /* Question 6. Donner les valeurs de a, b, c dans l'ordre croissant. */
236 /*****
237
238 /* Affichage de la question. */
239 printf("Q6. Donner les valeurs de %d, %d, %d dans l'ordre croissant.\n", a, b, c);
240
241 /* Pour la réponse de l'utilisateur on a besoin de 3 entiers. */
242 int x, y, z;
243
244 /* On prend sa réponse. */
245 printf("Premier nombre : ");
246 scanf("%d", &x);
247
248 printf("Deuxième nombre : ");
249 scanf("%d", &y);
250
251 printf("Troisième nombre : ");
252 scanf("%d", &z);
253
254 /*
255     Enfin, on regarde si l'ordre qu'il a choisi est le bon.
256     Si c'est le cas c'est une bonne réponse.
257 */
258 if (x <= y && y <= z)
259 {
260     printf("réponse correcte, bravo!\n");
261     nbReps = nbReps + 1;
262 }
263 /* Sinon, c'est une mauvaise réponse. */
264 else
265 {
266     printf("réponse incorrecte.\n");
267 }
268
269 /* Un petit saut de ligne avant de passer à la question suivante. */
270 printf("\n");
271
272 /*****
273 /* Question 7. Deux des nombres a, b, c sont-ils égaux entre eux? */
274 /*****
275
276 /* Affichage de la question et réponse de l'utilisateur. */
277 printf("Q7. Deux des nombres %d, %d, %d sont-ils égaux entre eux?\n", a, b, c);
278 scanf(" %c", &rep);
279
280 /*
281     Si l'utilisateur répond vrai et qu'il y a, a == b ou a == c ou
282     b == c, c'est une bonne réponse car on a au moins deux nombres égaux.
283     Ou si l'utilisateur répond faux et qu'on a pas a == b ou a == c ou
284     b == c, c'est qu'aucun des nombres ne sont égaux.
285     Ceux sont donc des bonnes réponses.
286 */
287 if ((rep == 'V' && (a == b || a == c || b == c)) ||
288     (rep == 'F') && !(a == b || a == c || b == c))
289 {
290     printf("réponse correcte, bravo!\n");
291     nbReps = nbReps + 1;
292 }
293 else
294 {
295     printf("réponse incorrecte.\n");
296 }
297
298
299 /* Saut de ligne pour faciliter la lecture de l'affichage. */
300 printf("\n");
301 printf("%d bonnes réponses sur 7.\n", nbReps);
302
303 return 0;
304 }

```

### 3 Exercice 5.2.3

```
1 #include <stdio.h>
2
3 int main(void)
4 {
5     /* Les entiers. */
6     int a, b, c;
7
8     /* Le nombre total de bonnes réponses. */
9     int nbReps;
10
11    /* Le caractère contenant la réponse de l'utilisateur. */
12    char rep;
13
14    /* Indice de la boucle while. */
15    int i = 1;
16
17    /* Les entiers de la partie précédente. */
18    int a1, b1, c1;
19
20    while (i <= 3)
21    {
22        /* On réinitialise le nombre de bonnes réponses. */
23        nbReps = 0;
24
25        /* Affichage du numéro de partie. */
26        printf("Partie %d\n", i);
27        printf("=====\n");
28
29        /* On demande les entiers à l'utilisateur. */
30        printf("Entrez trois entiers :\n");
31
32        printf("a : ");
33        scanf("%d", &a);
34
35        printf("b : ");
36        scanf("%d", &b);
37
38        printf("c : ");
39        scanf("%d", &c);
40
41        while (i >= 2 && a == a1 && b == b1 && c == c1)
42        {
43            printf("Choisissez trois nombres différents de"
44                " ceux choisi dans la partie %d !\n", i - 1);
45
46            printf("a : ");
47            scanf("%d", &a);
48
49            printf("b : ");
50            scanf("%d", &b);
51
52            printf("c : ");
53            scanf("%d", &c);
54        }
55
56        /* On enregistre les entiers saisis. */
57        a1 = a;
58        b1 = b;
59        c1 = c;
60
61        /* On lui indique le déroulement de l'exercice. */
62        printf("Répondre par V (vrai) ou F (faux) : \n");
63
64        /******
65        /* Question 1. a < b < c? */
66        /******
67        printf("Q1. %d < %d < %d? : ", a, b, c);
68        scanf(" %c", &rep);
```

```

69
70 /* Si les entiers sont bien triés. */
71 if (a < b && b < c)
72 {
73     /* Si il répond vrai, la réponse est correcte. */
74     if (rep == 'V')
75     {
76         printf("réponse correcte, bravo!\n");
77         nbReps = nbReps + 1;
78     }
79     /* Sinon, c'est qu'il a répondu faux, la réponse n'est pas correcte. */
80     else
81     {
82         printf("réponse incorrecte.\n");
83     }
84 }
85 /* Sinon, c'est que les entiers ne sont pas bien triés. */
86 else
87 {
88     /*
89     Si il a répondu faux, la réponse est correcte
90     (car les entiers ne sont effectivement pas bien triés).
91     */
92     if (rep == 'F')
93     {
94         printf("réponse correcte, bravo!\n");
95         nbReps = nbReps + 1;
96     }
97     /*
98     Sinon, c'est qu'il a répondu vrai. Or les entiers ne sont
99     pas bien triés. La réponse est donc incorrecte.
100    */
101    else
102    {
103        printf("réponse incorrecte.\n");
104    }
105 }
106
107 /* Un petit saut de ligne avant de passer à la question suivante. */
108 printf("\n");
109
110 /******
111 /* Question 2. Un seul nombre impair parmi a, b, c? */
112 /******
113
114 /* On prend la réponse de l'utilisateur. */
115 printf("Q2. Un seul nombre impair parmi %d, %d, %d? : ", a, b, c);
116 scanf(" %c", &rep);
117
118 /*
119 Pour vérifier combien il y a de nombre impairs,
120 on utilise une variable qui va nous permettre de compter
121 combien il y en a.
122 On initialise cette variable à 0.
123 */
124 int nbImpairs = 0;
125
126 /* On vérifie si a est impair. */
127 if (a % 2 == 1)
128 {
129     /* Si c'est le cas, on incrémente nbImpairs de 1. */
130     nbImpairs = nbImpairs + 1;
131 }
132
133 /* On vérifie si b est impair. */
134 if (b % 2 == 1)
135 {
136     /* Si c'est le cas, on incrémente nbImpairs de 1. */
137     nbImpairs = nbImpairs + 1;
138 }
139

```

```

140 /* On vérifie si c est impair. */
141 if (c % 2 == 1)
142 {
143     /* Si c'est le cas, on incrémente nbImpairs de 1. */
144     nbImpairs = nbImpairs + 1;
145 }
146
147 /*
148 Si l'utilisateur répond vrai et qu'il n'y a qu'un seul nombre impair,
149 il a une bonne réponse.
150 */
151 if (rep == 'V' && nbImpairs == 1)
152 {
153     printf("réponse correcte, bravo!\n");
154     nbReps = nbReps + 1;
155 }
156 /*
157 Sinon, si l'utilisateur répond faux et qu'il n'y a pas
158 qu'un seul nombre impair, il a une bonne réponse.
159 */
160 else if (rep == 'F' && nbImpairs != 1)
161 {
162     printf("réponse correcte, bravo!\n");
163     nbReps = nbReps + 1;
164 }
165 /*
166 Sinon, c'est que l'utilisateur a répondu faux et qu'il n'y a qu'un seul
167 nombre impair ou que l'utilisateur a répondu vrai et qu'il n'y a pas qu'un
168 seul nombre impair. C'est donc une mauvaise réponse.
169 */
170 else
171 {
172     printf("réponse incorrecte.\n");
173 }
174
175 /* Un petit saut de ligne avant de passer à la question suivante. */
176 printf("\n");
177
178 /******
179 /* Question 3. a, b, c distincts deux à deux? */
180 /******
181
182 /* On prend la réponse de l'utilisateur. */
183 printf("Q3. %d, %d, %d distincts deux à deux? : ", a, b, c);
184 scanf(" %c", &rep);
185
186 /*
187 Si l'utilisateur répond vrai et que tous les nombres sont différents OU
188 si l'utilisateur répond faux et que tous les nombres ne sont pas différents,
189 il a une bonne réponse.
190 */
191 if ((rep == 'V' && (a != b && b != c && a != c)) ||
192     (rep == 'F' && !(a != b && b != c && a != c)))
193 {
194     printf("réponse correcte, bravo!\n");
195     nbReps = nbReps + 1;
196 }
197 /* Sinon, c'est ni l'un ni l'autre, il a donc une mauvaise réponse. */
198 else
199 {
200     printf("réponse incorrecte.\n");
201 }
202
203 /* Un petit saut de ligne avant de passer à la question suivante. */
204 printf("\n");
205
206 /******
207 /* Question 4. Quelle est la valeur de a * b + b * c + c * a? */
208 /******
209
210 /* Entier servant à récupérer le résultat de l'utilisateur. */

```

```

211     int resultat;
212
213     /* On affiche la question et on prend la réponse de l'utilisateur. */
214     printf("Q4. Quelle est la valeur de %d * %d + %d * %d + %d * %d?\n",
215           a, b, b, c, c, a);
216
217     scanf("%d", &resultat);
218
219     /* Si le résultat est correcte, bonne réponse. */
220     if (resultat == a * b + b * c + c * a)
221     {
222         printf("réponse correcte, bravo!\n");
223         nbReps = nbReps + 1;
224     }
225     /* Sinon, mauvaise réponse. */
226     else
227     {
228         printf("réponse incorrecte.\n");
229     }
230
231     /* Un petit saut de ligne avant de passer à la question suivante. */
232     printf("\n");
233
234     /******
235     /* Question 5. Existe-t-il un triangle dont les côtés sont a, b, c? */
236     /******
237
238     /*
239     Pour qu'un triangle de côtés a, b, c existe il faut que
240     la somme de deux côtés soit plus grande ou égale au troisième côté.
241     Donc on doit avoir: a + b >= c, a + c >= b, b + c >= a.
242     */
243
244     /* Affichage de la question et obtention de la réponse de l'utilisateur. */
245     printf("Q5. Existe-t-il un triangle dont les côtés sont %d, %d, %d?\n",
246           a, b, c);
247
248     scanf(" %c", &rep);
249
250     /*
251     Si l'utilisateur répond vrai et qu'on respecte les inéquations des côtés,
252     c'est une bonne réponse. Ou si l'utilisateur répond faux et qu'on ne respecte
253     pas les inéquations des côtés, c'est une bonne réponse.
254     */
255     if ((rep == 'V' && (a + b >= c && a + c >= b && b + c >= a)) ||
256         (rep == 'F' && !(a + b >= c && a + c >= b && b + c >= a)))
257     {
258         printf("réponse correcte, bravo!\n");
259         nbReps = nbReps + 1;
260     }
261     /* Sinon, c'est ni l'un ni l'autre, c'est donc une mauvaise réponse. */
262     else
263     {
264         printf("réponse incorrecte.\n");
265     }
266
267
268     /* Un petit saut de ligne avant de passer à la question suivante. */
269     printf("\n");
270
271     /******
272     /* Question 6. Donner les valeurs de a, b, c dans l'ordre croissant. */
273     /******
274
275     /* Affichage de la question. */
276     printf("Q6. Donner les valeurs de %d, %d, %d dans l'ordre croissant.\n",
277           a, b, c);
278
279     /* Pour la réponse de l'utilisateur on a besoin de 3 entiers. */
280     int x, y, z;
281

```

```

282     /* On prend sa réponse. */
283     printf("Premier nombre : ");
284     scanf("%d", &x);
285
286     printf("Deuxième nombre : ");
287     scanf("%d", &y);
288
289     printf("Troisième nombre : ");
290     scanf("%d", &z);
291
292     /*
293     Enfin, on regarde si l'ordre qu'il a choisi est le bon.
294     Si c'est le cas c'est une bonne réponse.
295     */
296     if (x <= y && y <= z)
297     {
298         printf("réponse correcte, bravo!\n");
299         nbReps = nbReps + 1;
300     }
301     /* Sinon, c'est une mauvaise réponse. */
302     else
303     {
304         printf("réponse incorrecte.\n");
305     }
306
307     /* Un petit saut de ligne avant de passer à la question suivante. */
308     printf("\n");
309
310     /******
311     /* Question 7. Deux des nombres a, b, c sont-ils égaux entre eux? */
312     /******
313
314     /* Affichage de la question et réponse de l'utilisateur. */
315     printf("Q7. Deux des nombres %d, %d, %d sont-ils égaux entre eux?\n", a, b, c);
316     scanf(" %c", &rep);
317
318     /*
319     Si l'utilisateur répond vrai et qu'il y a, a == b ou a == c ou
320     b == c, c'est une bonne réponse car on a au moins deux nombres égaux.
321     Ou si l'utilisateur répond faux et qu'on a pas a == b ou a == c ou
322     b == c, c'est qu'aucun des nombres ne sont égaux.
323     Ceux sont donc des bonnes réponses.
324     */
325     if ((rep == 'V' && (a == b || a == c || b == c)) ||
326         (rep == 'F') && !(a == b || a == c || b == c))
327     {
328         printf("réponse correcte, bravo!\n");
329         nbReps = nbReps + 1;
330     }
331     else
332     {
333         printf("réponse incorrecte.\n");
334     }
335
336
337     /* Saut de ligne pour faciliter la lecture de l'affichage. */
338     printf("\n");
339     printf("%d bonnes réponses sur 7.\n", nbReps);
340
341     /* Enfin, on incrémente l'indice de boucle. */
342     i = i + 1;
343 }
344
345 return 0;
346 }

```