

```
1  //////////////////////////////////////
2  ////////////////////////////////////// TESTS POUR RETIRER //////////////////////////////////////
3  //////////////////////////////////////
4  void testRetirer1()
5  {
6      printTitre("TEST 4 : RETIRER");
7      // Initialisation de la collection
8      printGras("Collection initiale :");
9      int tabBase[] = {1,42,1,42,8,4,42,42,42,1,8,9,42,42,8,7,3,42,8};
10
11      Collection collectionTest(19,tabBase);
12      collectionTest.Afficher();
13
14      printGras("On retire 1 élément 42");
15      collectionTest.Retirer(42,1);
16      collectionTest.Afficher();
17  }
18
19  void testRetirer2()
20  {
21      printGras("Retirer plusieurs éléments");
22      // Retirer plus d'un seul élément
23      printGras("Collection initiale :");
24      int tabBase[] = {1,42,1,42,8,4,42,42,42,1,8,9,42,42,8,7,3,42,8};
25
26      Collection collectionTest(19,tabBase);
27      collectionTest.Afficher();
28
29      // Le test à proprement parler
30      printGras("On retire 4 éléments 42");
31      collectionTest.Retirer(42,4);
32      collectionTest.Afficher();
33  }
34
35  void testRetirer3()
36  { // Retirer tous les éléments d'une valeur donnée
37      printGras("On retire tous les éléments 42");
38      printGras("Collection initiale :");
39      int tabBase[] = {1,42,1,42,8,4,42,42,42,1,8,9,42,42,8,7,3,42,8};
40
41      Collection collectionTest(19,tabBase);
42      collectionTest.Afficher();
43
44      // Le test à proprement parler
45
46      printGras("Collection après Retirer(42,-1)");
47      collectionTest.Retirer(42,-1);
48      collectionTest.Afficher();
49  }
50
51  void testRetirer4()
52  {
53      // Suppression d'une valeur non comprise dans la Collection
54      printGras("On retire une valeur non présente dans la Collection (100)");
55      printGras("Collection initiale :");
56      int tabBase[] = {1,42,1,42,8,4,42,42,42,1,8,9,42,42,8,7,3,42,8};
57  }
```

```
58     Collection collectionTest(19,tabBase);
59     collectionTest.Afficher();
60
61     printGras("Collection après Retirer(100,-1)");
62
63     collectionTest.Retirer(100,-1);
64     collectionTest.Afficher();
65 }
66
67 void testRetirer5()
68 {
69     // Supression de toutes les valeurs :
70     int tabBase[] = {1,9,1,9,8,4,9,9,9,1,8,9,9,9,8,7,3,9,8};
71     printGras("On retire toutes les valeurs du tableau");
72     printGras("Collection initiale :");
73     Collection collectionTest(19,tabBase);
74     collectionTest.Afficher();
75
76     printGras("On retire tous les éléments de la Collection");
77     for (int i = 0; i < 10; ++i)
78     {
79         collectionTest.Retirer(i,-1);
80     }
81     collectionTest.Afficher();
82 }
83
84 //////////////////////////////////////
85 ////////////////////////////////////// TEST POUR AJUSTER //////////////////////////////////////
86 //////////////////////////////////////
87 void testAjuster1()
88 {
89     printTitre("TEST 3 : AJUSTER");
90     // TESTER SI LE REDIMENTIONNAGE AVEC UNE TAILLE TROP FAIBLE NE MARCHE PAS
91
92     printGras("Création d'une collection de taille 10 et population avec 5 éléments");
93     Collection collectionTest(10);
94     for (int i = 0; i < 5; ++i)
95     { collectionTest.Ajouter(i);
96     }
97     collectionTest.Afficher();
98     printGras("Test de réajustement dans le cas où nbElement > nouvelleTaille");
99     printGras("(de 10 à 4 de taille alouée avec 5 éléments)");
100    if(collectionTest.Ajuster(4))
101    { cout << "(ERREUR !) Le réajustement a réussi alors qu'il aurait du échouer\n";
102    }
103    else
104    { cout << "Le réajustement a bien échoué\n";
105    }
106    collectionTest.Afficher();
107 }
108
109 void testAjuster2()
110 {
111     printGras("Test de réajustement dans avec la nouvelle taille plus grande que l'ancienne ");
```

```
112     printGras("(De 10 à 20 de taille allouée)");
113
114     printGras("Création d'une collection de taille 10 et population avec 5 éléments");
115     Collection collectionTest(10);
116     for (int i = 0; i < 5; ++i)
117     {   collectionTest.Ajouter(i);
118     }
119     collectionTest.Afficher();
120
121     if(collectionTest.Ajuster(20))
122     {   cout << "Réajustement réussi\n";
123     }
124     else
125     {   cout << "(ERREUR !) Le réajustement a échoué\n";
126     }
127     collectionTest.Afficher();
128 }
129
130 void testAjuster3()
131 {
132     // Redimensionnement de la collection dans le cas où nouvelleTaille < alloue
133     printGras("Test de réajustement avec la nouvelle taille plus petite que l'ancienne ");
134     printGras("(De 20 à 10 de taille allouée)");
135     printGras("Création d'une Collection de taille 20 et population avec 5 éléments");
136     Collection collectionTest(20);
137     for (int i = 0; i < 5; ++i)
138     {   collectionTest.Ajouter(i);
139     }
140     collectionTest.Afficher();
141     if(collectionTest.Ajuster(10))
142     {   cout << "Réajustement réussi\n";
143     }
144     else
145     {   cout << "(ERREUR !) Le réajustement a échoué\n";
146     }
147     collectionTest.Afficher();
148 }
149
```