Licence S.T.A. - semestre 2
Informatique - A.P.I.1
TP sur les recherches dans un tableau

Exercice 1 : *Les unités U_ELEMENT et U_TABLEAU*

Récupérez les fichiers `U_ELEMENT.pas` et `U_TABLEAU_pour_etudiant.pas` sur le semainier.
Ces deux fichiers contiennent

- une unité *U_ELEMENT* pour la manipulation des éléments qui seront dans les cases des tableaux (lecture, écriture et choix au hasard d'un élément)

  Comme vous pouvez le constater sur cet exemple très simple, une unité contient

  - une partie *interface* où on met les choses que le programme pourra utiliser (ici le type `ELEMENT` et trois sous-programmes pour lesquels on ne met que les entêtes)
  - une partie *implementation* où on met l'écriture complète des sous-programmes déclarés dans l'interface plus éventuellement des choses nécessaires à cette écriture mais pas utiles pour le programme utilisant l'unité.
  - et parfois comme ici une partie *initialization* qui s'exécute dès que le programme qui utilise l'unité fait appel à celle-ci (ici c'est l'initialisation du générateur aléatoire)
  - il peut y avoir aussi une partie *finalization* mais je n'en parle pas.

  Remarquez qu'ici, cette unité n'est qu'un simple rennomage des outils sur les `CARDINAL`s et ceci pour pouvoir éventuellement remplacer `CARDINAL` par un autre type (`CHAR` par exemple)

- une unité *U_TABLEAU_pour_etudiant* (que vous renommerez en *U_TABLEAU* quand vous l'aurez complétée, attention le nom de l'unité et celui du fichier qui la contient doivent être le même).

  Cette unité contient les déclarations des types `INDICE` et `TABLEAU` (tableaux dynamiques car lors du TP sur l'évaluation des tris on choisira des "petits" tableaux puis de très "grands" tableaux).

  Ensuite on a les procédures d'entrées-sorties, puis des fonctions de générations de tableaux, pour un tableau quelconque, pour un tableau trié trivial, pour un tableau trié non trivial (vous devez compléter son écriture dans la partie *implementation*), pour un tableau super-trié (même chose), pour des tableaux anti-triés, et surtout une fonction pour créer une "vraie" copie d'un tableau car une affectation de tableau dynamique ne crée pas un second tableau mais donne un second nom au même tableau (voir le programme *tester_U_TABLEAU*), et pour finir on a mis aussi le calcul du milieu d'un intervalle d'`INDICE`s (ces deux derniers sous-programmes seront très utiles pour la recherche dichotomique et le tri fusion).

Pour tester l'unité *U_TABLEAU* et surtout ce que vous allez compléter vous pouvez récupérer le programme *tester_U_TABLEAU* dans le fichier `tester_U_Tableau_pour_etudiant.pas`.
Remarque : ce programme compile et fonctionne mais ne fera pas ce qu'il faut tant que vous n'aurez pas complété l'unité *U_TABLEAU*. Vous avez en-dessous du programme des résultats analogues à ceux que vous devez obtenir (ce ne sont jamais exactement les mêmes puisque les tableaux sont tirés au hasard)

Exercice 2 : *Les algorithmes de recherche*

Récupérez le fichier `recherches_pour_etudiant.pas`.
Vous devez compléter les fonctions et procédures. Le programme principal est écrit, compile et fonctionne mais ne sera "efficace" que lorsque vous aurez écrit les sous-programmes (vous pouvez tester au fur et à mesure si ce que vous avez écrit est correct).
Ici aussi il y a les résultats d'une exécution sous le programme.