**Tâches à effectuer**

Vous devez concevoir un programme du tirage du loto qui pour chaque séquence tire 5 boules dont la valeur est comprise entre 1 et 45. Les séquences devront être générées par un générateur aléatoire uniforme sans remise. La graine aléatoire sera fixée directement dans le code afin de pouvoir obtenir les mêmes séquences quelque soit l'ordinateur utilisé. Dans ce projet une liste de tâches relatives aux probabilités et aux algorithmes sont données ci-dessous. La réalisation de ces tâches devront figurées dans le rapport final et les programmes développés devront être déposés.

**Partie probabilités**

**Exercice du TD**

* Donner les solutions de l'exercice 3.
* Donner les solutions de l'exercice 4.

**Probabilités relatives au loto**

* Donner le nombre de combinaisons possibles pour ce jeux du loto quand l'ordre est pris en compte.
* Donner le nombre de combinaisons possibles pour ce jeux du loto quand l'ordre n'est pas pris en compte.
* Donner la probabilité d'avoir les 5 bon numéros (quelque soit l'ordre).
* Donner la probabilité d'avoir les 5 bon numéros (en respectant l'ordre du tirage).

**Partie algorithme et programmation**

**Algorithmes et programmes relatifs au tri**

Les algorithmes de tri pourront être utilisés sur une séquence relative à un tirage particulier et sur l'ensemble des séquences relatives aux différents tirages.

* Présenter l'algorithme du tri cocktail (itératif) et écrire l'algorithme.
* Implémenter l'algorithme du tri cocktail en Python.
* Présenter l'algorithme du tri par insertion (itératif) et écrire l'algorithme.
* Implémenter l'algorithme de tri par insertion en Python.
* Présenter l'algorithme du tri fusion récursif et écrire l'algorithme.
* Implémenter l'algorithme de tri fusion en Python.

**Recherche dichotomique**

* Présenter l'algorithme de recherche dichotomique.
* Écrire l'algorithme itératif.
* Écrire l'algorithme récursif.
* Implémenter les deux algorithmes en Python.

**Sauvegarde et chargement des données**

Les méthodes de sauvegarde et de chargement devront pouvoir s'appliquer sur l'ensemble des séquences tirées avec la possibilité d'identifier les numéros de chaque séquence. Vous pouvez utiliser la bibliothèque Python de votre choix pour la sauvegarde et le chargement des données.

* Effectuer une recherche bibliographique sur la différence entre un format binaire et lisible humainement et présenter deux formats pour ce dernier.
* Implémenter une méthode relative à la sauvegarde de données binaires et une autre relative au chargement de données binaires.
* Implémenter deux méthodes de sauvegarde et de chargement de données lisibles humainement, chacune pour un format différent.

**Visualisation de la distribution des numéros tirés**

* Présenter ce qu'est un histogramme.

Il existe un type de graphique qui est utilisé pour représenter les données numériques. Il se compose d'une série de boîtes horizontales disposées côte à côte. Chacune de ces boîtes représente un intervalle de valeurs et sa hauteur est liée au nombre de valeurs dans cet intervalle. Il est important de noter que les données utilisées pour ce type de graphique doivent être discrètes et les intervalles doivent avoir la même largeur. Il existe plusieurs façons de créer ce genre de graphique, différents logiciels peuvent être utilisés pour cela. Ils permettent de visualiser les tendances générales dans les données de manière simple à comprendre.

* Écrire l'algorithme du calcul de l'histogramme des numéros sortis.
* Implémenter l'algorithme en Python.
* Tracer l'histogramme des valeurs tirées avec un digramme en bâton grâce à la bibliothèque Matplotlib.
* Quel observation faites-vous lorsque le nombre de tirages est élevé ?

On remarque que sur un nombre élever de tirage, le histogramme commence a devenir plat ce qui signifie que les nombres tirés apparaissent quasiment a la même fréquence, ce qui est en théorie exacte si il s’agit bien d’un tirage aléatoire, et donc prouve que la probabilité d’obtenir une nombre est égale pour chacun soit 1/45.