

## PRG1. Labo 02 – Recherche par dichotomie

### Objectif :

- Mettre en œuvre les branchements (if .. else), les boucles, et le calcul numérique.
- Implémenter un mode de test du programme sans duplication de code.
- Utiliser la fonction rand() pour générer des valeurs aléatoires.

### Contexte :

La recherche par dichotomie permet de trouver un nombre dans un intervalle donné grâce à des questions oui-non. Par exemple, pour deviner un nombre qui a été choisi entre 1 et 1000, on peut utiliser une série de questions du type « Le nombre est-il plus petit ou égal à X ? ». Si on choisit correctement les valeurs successives de X pour effectuer une recherche dichotomique, alors on peut trouver le nombre choisi entre 1 et 1000 avec seulement 10 questions.

Voir par exemple [https://fr.wikipedia.org/wiki/Recherche\\_dichotomique](https://fr.wikipedia.org/wiki/Recherche_dichotomique). Attention, les algorithmes qui y figurent doivent être considérablement modifiés pour pouvoir être appliqués à ce labo.

### Travail demandé :

Votre programme implémentera la recherche dichotomique sous la forme d'un jeu avec l'utilisateur, comme indiqué dans l'exemple de déroulement suivant :

```
Entrez la borne inferieure de l'intervalle : 1
Entrez la borne superieure de l'intervalle : 20
Veuillez choisir un nombre dans cet intervalle,
puis repondre aux questions par 'o' ou 'n'.
1. Le nombre est-il plus petit ou egal a 10 ? n
2. Le nombre est-il plus petit ou egal a 15 ? n
3. Le nombre est-il plus petit ou egal a 18 ? o
4. Le nombre est-il plus petit ou egal a 17 ? o
5. Le nombre est-il plus petit ou egal a 16 ? n
Le nombre choisi est 17.
```

Le programme offrira également un mode de test, sélectionné par une question initiale, dans lequel on entre également le nombre choisi, ce qui permet au programme de répondre tout seul aux questions (en affichant les questions et les réponses comme indiqué ci-dessus). L'affichage ci-dessus sera donc complété au début avec les deux lignes suivantes (la seconde en mode test seulement) :

```
Mode jeu ou mode test ? Veuillez repondre par 'j' ou 't' : t Entrez le nombre que
vous avez choisi : 17
```

Pour le choix de la valeur X qui divise l'intervalle vous devez utiliser deux méthodes :

- X divise l'intervalle par 2. E.g. comme l'exemple d'exécution ci-dessus. Ceci est la méthode par défaut.

- Utiliser la fonction `rand()` de la librairie `cstdlib` pour le choix de la valeur `X` et comparer le nombre d'itérations nécessaires avec celui obtenu précédemment (division par moitié). C'est plus ou moins rapide ? <https://cplusplus.com/reference/cstdlib/rand/>  
E.g. pour un intervalle 1 à 20, `X` est une valeur aléatoire entre 1 et 20.  
Pour cette méthode, vous demandez à l'utilisateur s'il souhaite utiliser la méthode `rand` avant de commencer.

## Spécifications techniques :

Le programme respectera les consignes suivantes :

- Vérifier que l'utilisateur saisit des caractères acceptables ('o' ou 'n', 'j' ou 't'), que les limites de l'intervalle sont dans le bon ordre, et (en mode test) que le nombre donné est entre ces limites – et ne pas avancer dans le programme tant que ces données ne sont pas correctes
- Éviter la duplication de code, en faisant des choix appropriés pour le mode test
- Vous êtes libres de choisir une autre forme pour les questions, mais les réponses doivent toujours être binaires ('o' pour 'oui' et 'n' pour 'non')

## Remise du travail :

- Votre programme sera contenu dans un seul fichier appelé `main.cpp`
- Le code devra être rédigé et commenté clairement, selon les indications reçues dans le cours.
- Veuillez soumettre le fichier sur Cyberlearn.