

# Programmation 1

## TP 2 : Le marin ivre

Après un long séjour en mer, lorsqu'un bateau fait escale dans un port, il arrive que les marins descendant à terre boivent un peu plus que de raison. Dans ce cas là, ils peuvent avoir quelques difficultés à remonter à bord. Pour remonter à bord, depuis le quai, les marins doivent marcher sur une planche. Si un marin ne marche pas très droit sur la planche, il peut tomber à l'eau avant d'avoir regagné le navire.

Un marin non ivre marche tout droit : quand il est sur la planche, à chaque pas il a une probabilité égale à 1 de marcher vers l'avant. Au bout de `long` pas, il est à bord du bateau. Nous supposons qu'un marin ivre a, à chaque pas, une probabilité égale à :

- 0,5 d'aller vers l'avant ;
- 0,2 d'aller vers la droite ;
- 0,2 d'aller vers la gauche ;
- 0,1 d'aller vers l'arrière.

Le sens considéré va du quai vers le bateau.

Pour simplifier, nous supposons que :

- les dimensions de la planche se mesurent en nombre de pas : la planche a une largeur `largeur` impaire (par exemple, 3 pas) et une longueur `longueur` supérieure à `largeur` ;
- un marin commence toujours par un pas le positionnant au milieu de la largeur de la planche. Par exemple, en position 1 pour une planche de largeur 3 (de 0 à 2) ;
- si l'on désigne par  $x$ ,  $y$  la position du marin sur la planche, dans l'exemple ci-dessus (`largeur` = 3), il commence toujours en  $x = 1$  et  $y = 0$  ;
- suite à un certain nombre de pas en arrière, si un marin se retrouve sur le quai ( $y = -1$ ), il est automatiquement repositionné à la position de départ ( $x = 1$  et  $y = 0$  dans l'exemple) ;
- si  $x$  atteint la valeur  $-1$  ou `largeur`, alors le marin tombe à la mer et ne regagnera jamais le bateau ;
- le marin est à bord dès que  $y$  atteint la valeur `longueur`.

Problème : simuler le parcours d'un marin ivre sur une planche... et déterminer la « probabilité empirique » qu'il a de pouvoir remonter à bord du bateau, en fonction des dimensions de la planche.

### 1 Générateur aléatoire

Programmer un générateur de nombres aléatoires qui générera des entiers compris entre 1 et 100. Indications dans `Annee1/prog1/Utile/Complements_Ada_Java.doc` sur Moodle ou utiliser, en Java, la fonction `Ut.randomMinMax(int min, int max)`.

On vérifiera sur un grand nombre de tirages (par exemple 1000) que :

- environ 50% de ces nombres sont compris entre 1 et 50 ;
- environ 20% de ces nombres sont compris entre 51 et 70 ;
- environ 20% de ces nombres sont compris entre 71 et 90 ;
- environ 10% de ces nombres sont compris entre 91 et 100.

## 2 Simulation d'un marin ivre

1. Ecrire une fonction `arivobato` qui, étant données les dimensions de la planche, simule la trajectoire d'un marin ivre sur la planche (sans rien afficher) et retourne `vrai` si le marin parvient à monter sur le bateau et `faux` sinon.
2. Ecrire une fonction `affichePlanche` qui, étant données les dimensions de la planche et la position du marin, affiche la planche (avec des caractères `'|'` et `'-'`) et le marin (représenté par un `'o'` par exemple).  
Suggestion (pour l'implantation en Java) : utiliser la fonction `Ut.clearConsole()` pour effacer la console (le terminal) et la fonction `Ut.pause(int duree)` pour figer le processus pendant `duree` millisecondes.  
Variante : Dans la fonction `arivobato`, définir une structure de données pour mémoriser toutes les positions du marin au cours de sa marche, par exemple une matrice de booléens ayant les dimensions de la planche. Cette structure de données est transmise à la fonction `affichePlanche` pour l'affichage.
3. Modifier la fonction `arivobato` pour qu'elle accepte un paramètre `trace` supplémentaire. Si `trace` est `faux`, la fonction n'affiche rien et se comporte comme la première version. Si `trace` est `vrai`, la fonction fait en plus des affichages, notamment des affichages de la planche et du marin à chaque pas.

## 3 Quelques statistiques

1. Pour estimer la probabilité qu'a un marin de monter à bord, on calcule sur un grand nombre de marins (par exemple 100) combien atteignent le bateau.  
Ecrire une fonction `probaEmpirique` qui, étant données les dimensions de la planche et un nombre `nbMarins` de marins, retourne la proportion (un réel compris entre 0 et 1) de marins parvenus à monter à bord sur `nbMarins` marins.
2. Ecrire une procédure `afficheProbas` qui, étant données la largeur de la planche et 3 entiers positifs `longMin`, `longMax` et `nbMarins`, affiche les probabilités empiriques d'arriver au bateau pour les paramètres donnés et les longueurs de la planche allant de `longMin` à `longMax`.  
Vérifier à l'exécution que (pour une valeur suffisamment grande de `nbMarins`!) plus la planche est longue, plus la probabilité empirique d'arriver au bateau est petite.
3. Ecrire une fonction `largeurMin` qui, étant données la longueur de la planche, un entier positif `nbMarins` et un réel `probaMin` compris entre 0 et 1, retourne la largeur minimum de la planche pour que la probabilité empirique d'arriver au bateau pour les paramètres donnés soit au moins égale à `probaMin`.
4. Ajouter un menu, c'est-à-dire une fonction « principale » `menuMarin`, qui permet à l'utilisateur de choisir de lancer l'une des fonctions écrites, selon un numéro saisi. L'utilisateur saisira 0 pour sortir du menu et quitter le programme.

## 4 Pour aller plus loin

1. Du fait que la probabilité est empirique, deux exécutions de la fonction `largeurMin` avec les mêmes paramètres peuvent retourner des résultats différents.

Ecrire une procédure `testStabilite` qui étant donnés les paramètres de la fonction `largeurMin` et un entier positif `nbCalculs`, effectue `nbCalculs` calculs de la largeur minimum de la planche pour les paramètres donnés et affiche le plus petit et le plus grand des résultats obtenus.

Vérifier à l'exécution que pour une valeur suffisamment grande de `nbMarins` (et suffisamment petite de la longueur de la planche), les 2 résultats affichés sont identiques.

2. On souhaite que les probabilités d'aller vers l'avant, vers la droite et vers la gauche soient choisies par l'utilisateur (celle d'aller vers l'arrière se déduisant des 3 autres). Ces 3 paramètres supplémentaires sont saisis initialement par l'utilisateur dans le programme principal, et peuvent être modifiés à l'aide du menu. Modifier le programme en conséquence.