1 Enoncé 1

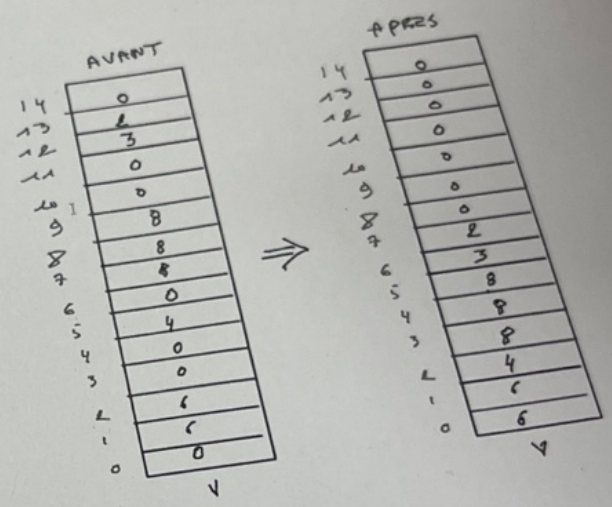
Créez une fonction qui devra compresser les blocs contenus dans le vecteur et retourner le nombre de cases libres.

Exemple :

Un bloc est représenté par des nombres identiques (>0) qui se suivent dans le vecteur. Une case vide contiendra 0.

Vous devrez faire descendre les blocs pour combler les trous (cases à 0).

Retour : 7 (nombre de cases à 0)



Fonction : **int compresse(int[] v)**

Hypothèse de départ :

- #v > 0 ,
- une case vide est à 0
- un bloc correspond à 1..x cases consécutives contenant le même nombre

Variable d'entrée de la fonction

- v est un vecteur d'entiers (entrée et sortie)

Variable de sortie de votre fonction :

- nbCaseVide un entier qui indiquera le nombre de cases libres (à 0).

Variables locales (nom, type et description)

inferieur a 32.

2 Réalisez une fonction « boolean findCode(int n, int code) »

Cette fonction va devoir renvoyer « true » lorsque le code se retrouve dans le nombre n. Sachant que code représente un nombre <32 (5 bits)

Si la variable « n » = 011011101110111_2 et code = 11011_2 → retour vrai

Si la variable « n » = 101011100101111_2 et code = 11101_2 → retour faux

Remarques :

Pour obtenir la taille de la variable en pseudo-code, vous pouvez écrire ceci :

nbr ← nombre de bits de n

ainsi en Java si n est de type « int », la variable nbr contiendra 32.

Hypothèse de départ :

« code » représente un nombre < 32

Variables d'entrée de la fonction :

- « n » : un entier
- « code » est un entier inférieur à 32

Variable de sortie de la fonction :

- « trouve » un booléen qui indiquera si le code a été trouvé dans « n » ou non.

Variables locales (nom, type et description)

3 Créez une fonction qui devra indiquer la taille du bateau situé dans une position

Considérons une grille de n*n représentant un espace d'eau dans lequel se trouve des bateaux.

Un bateau occupe 3 où 5 cases consécutives (horizontalement ou verticalement)

Aucun bateau ne se touche.

Ecrivez la fonction : **int quiEstLa(int[][] m, int x, int y)**

Exemples :

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	0	1	1	1	1	1	0	-1
-1	0	0	0	0	0	0	0	-1
-1	1	1	1	0	0	0	0	-1
-1	0	0	0	1	0	0	0	-1
-1	0	0	0	1	0	1	0	-1
-1	0	0	0	1	0	1	0	-1
-1	0	0	0	0	0	1	0	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1

- **quiEstLa(m,3,2) → 3**
- **quiEstLa(m,3,4) → 0**
- **quiEstLa(m,1,3) → 5**

Hypothèses de départ :

« m » : une matrice carrée d'entiers avec une bordure tel que :

- -1 : désigne la bordure
- 0 une case vide
- 1 une case occupée par un bateau

Aucun bateau ne se touche.

Variables d'entrée de la fonction (nom, type et description)

m : matrice carrée d'entiers avec une bordure comme indiqué ci-dessus

x et **y** une position dans la grille (1.. taille-2)

Variable de sortie de votre fonction : (nom, type et description)

Un entier désignant la taille du bateau ou 0 s'il la case en (x,y) est vide.

Variables locales utilisées et description :