Département : Mathématiques et Informatique	**EXAMEN 3**
Année Universitaire : 2013/2014	
Matière : Informatique (1ère Année MP, PC, PT)	**Date : 26/05/2014**
Nombre des pages : 4	**Durée :** 2 heures

Exercice 1 : *« Le tri cocktail»*

On considère le type suivant :

CONST MAX = 10000

TYPE TAB = tableau [1..MAX] de booléen

Le but de cet exercice est d'écrire le programme qui effectue le tri dit *double bulle*, ou *tri cocktail* d'un tableau. L'algorithme double bulle s'inspire de l'algorithme de tri par bulles.
Chaque itération de l'algorithme est divisée en deux étapes.
Etape 1 : La première étape consiste à parcourir le tableau jusqu'à la fin en permutant les deux à deux les éléments, à la manière du tri par bulle, pour placer le max des éléments à la dernière position.
Etape 2 : La deuxième étape consiste à retraverser le tableau en sens inverse (en commençant à la position du max-1), en permutant les éléments deux à deux lorsque c'est nécessaire, pour placer le min des éléments à la première position.
Ces itérations sont répétées tant qu'il y a des permutations faites.
Exemple d'exécution :

Tableau initial : **9 6 5 0 1 2 10**

Etape 1
6 5 0 1 2 9 10
Etape 2
0 6 5 1 2 9 10
} **Itération 1**

Etape 1
0 5 1 2 6 9 10
Etape 2
0 1 5 2 6 9 10
} **Itération 2**

Etape 1
0 1 2 5 6 9 10
Etape 2
0 1 2 5 6 9 10
} **Itération 3**

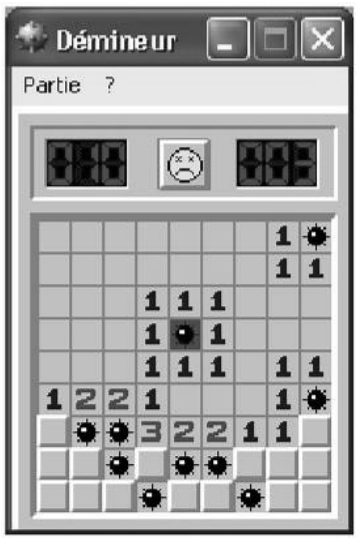
Le démineur est un jeu de réflexion dont le but est de trouver les mines qui sont cachées aléatoirement dans un champ.

Si la case choisie contient une mine, la partie est perdue.

Si la case choisie ne contient pas de mine alors apparaîtra un chiffre indiquant le nombre de mines qui se trouvent dans les cases directement voisines de celle jouée (nombre compris entre 0 et 8).

Par exemple si le numéro découvert est un 2, cela indique qu'il y a 2 mines cachées parmi les 8 cases qui touchent directement celle choisie.

La partie est gagnée lorsque toutes les cases sont découvertes, exceptées celles minées qui doivent avoir été marquées.



Pour réaliser un programme qui implémente ce jeu, On utilisera un tableau d'entiers à deux dimensions nommé ***CHAMP*** (taille N*N),

- ***CHAMP [i,j]=-***1 signifie que la case de coordonnées (i,j) contient une mine.
- ***CHAMP [i,j]*** ∈ [0..8] signifie que la case de coordonnées (i,j) est vide et la valeur correspond au nombre de cases voisines contenant une mine.

On utilisera un second tableau à deux dimensions d'entiers ***JEU*** (taille ***N*N***), pour mémoriser les cases choisies par l'utilisateur :

- ***JEU [i,j]*** =0 si la case aux coordonnées (i,j) n' a pas encore été choisi par le joueur.
- ***JEU [i,j]*** =1 si le joueur a déjà choisi précédemment la case de coordonnées (i,j).

Après chaque sélection d'une case par le joueur et si la case ne contient pas de mine, le programme affichera la grille de jeu dont l'apparence d'une case sera :

- Le caractère **?** pour une case encore inconnue
- Le caractère ***** pour une case qui contient une mine.
- Le nombre de cases voisines contenant une mine pour une case déjà choisi.

En début de partie le joueur doit entrer la taille de la grille ***N***, ainsi que le nombre de mines ***nbmines.***

La fin du jeu et du programme intervient de deux manières différentes :

- Victoire : le joueur a découvert les cases vides ***(CV = N*N - nbmines).***
- Défaite : le joueur est tombé sur une mine. on affichera « perdu » !

E

-1122112-1-12224-12-1-121122100

e de

??????????????? ?????????L'afficha ge suivant au début de partie

⇨

sé

??2??????????4??? ?????????Après sélection des cases (1,3) et (3,3)

⇨

??2?????*???4??????? ?????Après sélection des cases (1,3) (3,3) et (2,4) .. Perdu !

co

su

1)

Supposons qu'on a fait les déclarations suivantes :

CONSTANTE MAX=100

TYPE GRILLE=TABLEAU[1..MAX, 1..MAX] D'entier

1) Présenter l'affichage pour le joueur et le contenu du tableau ***JEU*** Après sélection des cases (1,3) (3,3) (2,4) (5,5) et (5,1)

2) Ecrire une procédure `INIT` qui initialise toutes les cases de la grille ***JEU*** à l'état « non encore sélectionné » et les cases du ***CHAMP*** à « case vide ».

3) Ecrire la fonction `caseJouable` retourne vrai (faux sinon) si la case de coordonnées (L, C) sur la grille ***JEU*** n'a pas été encore sélectionné et dont les coordonnés sont valides.

4) Ecrire la fonction `caseMinee` retourne vrai (faux sinon) si la case de coordonnées (L, C) sur la grille ***CHAMP*** est minée.

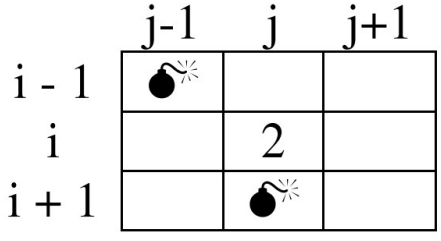
5) Ecrire une procédure `pose_Mines` permettant de positionner aléatoirement ***nbmines*** mines sur la grille de ***CHAMP***. Pour placer une mine aléatoirement, il faut :

- choisir un nombre aléatoire compris entre 1 et N pour la ligne et la colonne.
- Ensuite on regarde si la case choisie est une case comporte déjà une mine en utilisant la fonction `caseMinee`.
- On répète ce processus pour placer ***nbmines.***

On suppose qu'on dispose de la ***HASARD (Binf, Bsup)*** qui fournit un nombre aléatoire compris entre ***Binf et Bsup.***

6) Pour remplissage complet de la grille de ***CHAMP***, écrire une procédure `comp_Mines` permettant de calculer le nombre de cases minées directement voisines d'une case de coordonnées (L, C) et mettre à jour cette grille.

Une méthode pour parcourir le contour d'une case consiste à utilisé les indices ***i-1, i, i+1*** pour les lignes ***j-1, j, et j+1*** pour les colonnes.



7) En utilisant la fonction `caseJouable,` écrire une procédure `COUP` permettant de gérer le coup d'un joueur sur la grille ***JEU*** en lui demandant les coordonnées de la case choisie, puis d'appliquer ce choix.

8) La partie s'arrête quand toutes les cases ont été découvertes sauf celles contenant des mines (partie gagnée) ou bien lorsqu'une mine est touchée (partie perdue). Écrire une fonction `partie_gagne` retourne VRAI (faux sinon) si ***N*N - nbmines*** case est découverte et aucune mine n'est touchée.

9) Ecrire une fonction `partie_perdu` analogue à `partie_gagne` (décrite en question 8) qui retourne VRAI (faux sinon) si une mine est touchée.

10) Ecrire une procédure `jouer` qui permet de dérouler le jeu et d'afficher le résultat finale. Le schéma général d'une partie est le suivant :

- Affichage de la grille de ***JEU*** et la grille ***CHAMP*** selon le modèle donné par l'énoncé
- Saisir un coup du joueur en utilisant la procédure `COUP` .
- Refaire les deux étapes précédentes jusqu'à ce que le joueur perde, gagne ou abandonne la partie (il faut donner la possibilité d'arrêter le jeu).