

Année universitaire 2015-2016  
**Examen**  
**Matière : Structures de Données II**

Date : 18/05/2016  
Filière / Classe : 1<sup>ère</sup> ING-INFO

Durée : 1h30  
Documents : Non Aut.

*Les copies propres et bien soignées sont très appréciées. Il sera tenu compte de la lisibilité des réponses.*  
*Bonne chance !*

**Problème 1 (10 pts):**

Nous désirons simuler un jeu avec 52 cartes entre deux joueurs. Le jeu se déroule en plusieurs tours jusqu'à ce que l'un des deux joueurs gagne.

On distribue, initialement, les 52 cartes aux 2 joueurs. Chaque joueur rassemble ses cartes en un paquet de 26 face cachée, devant lui. On appellera ce paquet le jeu du joueur.

Les cartes sont munies d'une relation d'ordre : le roi est la plus forte carte, puis viennent la dame, le valet, le 10, 9, ..., 3, 2, As.

Un tour est organisé comme suit : chaque joueur tire la carte du dessus de son jeu et la pose sur la table, face découverte, à côté de son jeu, sur sa pile: cette pile sera la pile de bataille. Celui qui a la carte la plus forte ramasse les 2 cartes et les enfile dans son jeu.

Lorsque les deux joueurs posent deux cartes de même valeur, il y a "bataille". Chaque joueurs tire la carte suivante de son jeu et la pose, face cachée, sur la carte précédente de la pile de bataille. Puis il tire une deuxième carte de son jeu qu'il pose cette fois-ci face découverte sur cette pile de bataille. Il y a maintenant 3 cartes sur la pile de bataille de chaque joueur. C'est la dernière carte découverte qui départagera les joueurs. Si ces cartes sont aussi identiques, il y a à nouveau, bataille.

Lorsqu'un des deux joueurs gagne, il remporte toutes les cartes des deux piles de bataille (la sienne et celle de son adversaire) et il met toutes les cartes de ces 2 piles sous son jeu. Ainsi le tour se termine et on passe au tour suivant.

Les tours se succèdent jusqu'à ce que l'un des deux joueurs gagne.

Lorsqu'un des deux joueurs n'a pas de cartes pour commencer un nouveau tour ou pour continuer la bataille en cours, c'est la notion de famine. Il perd et c'est son adversaire qui gagne.

Pour simuler le jeu, les structures de données utilisées sont des files et des piles.

Le jeu de chaque joueur peut être représenté par une **file**. Au début du tour, le joueur prend la carte du dessus (défiler) et s'il gagne, il prend les cartes jouées dans les deux piles de bataille et les ajoute par-dessous (enfiler).

Les **piles de bataille** sont des **piles**, car on ajoute toujours sur le sommet de la pile de bataille et on regarde en comparant les cartes du sommet. Quand on gagne, on reprend les cartes une par une à partir du sommet des piles de bataille (dépiler) et on les ajoute au jeu du joueur gagnant en les enfilant.

Une carte est définie dans notre problème par une valeur (Roi, Dame, ..., As), une couleur (trèfle, carreau, cœur, pique) et sa visibilité (retournée et cachée ou visible)

1.
  - a. Définir la structure de données **CARTE** permettant de représenter et manipuler les cartes.
  - b. Définir la structure de données **PILEBATAILLE** permettant de représenter les piles de cartes en utilisant des structures chaînées.
  - c. Définir la structure de données **JEU** permettant de représenter les files de cartes.
  - d. Un joueur est caractérisé par son jeu et par sa pile de bataille. Définir la structure de données **JOUEUR** permettant de représenter un joueur.

2. Nous supposons disposer de la fonction **Compare\_carte(C1, C2 : CARTE) : entier** qui retourne :

1 : si la carte C1 est de valeur supérieure à C2;

0 : en cas d'égalité;

-1 : sinon.

Écrire la procédure **Tour(var J1, J2: JOUEUR)** qui simule le déroulement d'un tour de notre jeu. Au début du tour nous supposons que chaque joueur dispose d'au moins d'une carte dans son jeu.

Indication : vous pouvez développer d'autres fonctions pour simplifier le travail.

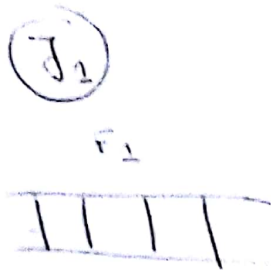
3. Nous supposons disposer de la **procédure Init\_joueurs(var J1, J2 : JOUEUR)** qui permet d'initialiser les deux joueurs (distribuer les cartes).

En supposant que les primitives de gestion des piles et des files sont prédéfinies, écrire une fonction **Jeu(...)** : **entier** qui simule le déroulement de notre jeu, elle retourne 1 si le premier joueur a gagné, et 2 sinon.

Jeu = { J1, J2 }

J1 gagne  $\Leftrightarrow F_2$  de J2 est vide

J2 gagne  $\Leftrightarrow F_1$  de J1 est vide



↑  
Silex



↑  
Pile Balaille



a - Type CARTE = enregistrement

valeur : {Roi, Dame, ..., As}

couleur : {Trèfle, Carreau, Cœur, Pique}

visibilité : Boolean

fin enregistrement

(c.75)

b -

PileLst = enregistrement

carte : CARTE

suiv : ^PileLst

Fin enregist

Pile Balaille = ^PileLst

(c.75)



c. Filet = enregistrement  
carte : CARTE  
suiv : ^Filet  
Finenregistrement

(0,75)

Jeu, z / ^Filet

Jeu = enregistrement  
Tête, Queue : ^Filet

d. joueur = enregistrement  
pile-bat : PILEBATAILLE  
jeu : JEU

(0,75)

Finenregistrement

2. Procédure Tom (sur  $J_1, J_2$  : joueurs)  
sur  $C_1, C_2$  : CARTE

Début

$C_1 \leftarrow \text{Tête} (J_1 \cdot \text{jeu})$

$C_1 \cdot \text{visibilité} \leftarrow \text{vrai}$

Défiler ( $J_1 \cdot \text{jeu}$ )

Empiler ( $J_1 \cdot \text{pile-bot}$ ,  $C_1$ )

$J_1$

$C_2 \leftarrow \text{Tête} (J_2 \cdot \text{jeu})$

$C_2 \cdot \text{visibilité} \leftarrow \text{vrai}$

Défiler ( $J_2 \cdot \text{jeu}$ )

Empiler ( $J_2 \cdot \text{pile-bot}$ ,  $C_2$ )

$J_2$

Selon (Compare-Conte ( $C_1, C_2$ )) faire

cas "1" : Ramasse ( $J_1, J_2$ )

cas -1 : Ramasse ( $J_2, J_1$ )

cas 0 :  $S_1$  (Bataille ( $J_1, J_2$ ) = 1) alors  
Ramasse ( $J_1, J_2$ )

sinon  
Ramasse ( $J_2, J_1$ )

FinSelon

Fin

Fin

Procédure Ramasse (var  $X_1, X_2$  : JOUEUR)  
var C : CARTE

Début

Tantque Non Vide ( $X_1$ . pile - bat) faire  
 $C \leftarrow$  Sommet ( $X_1$ . pile - bat)  
C.v.is.bilité  $\leftarrow$  Faux  
Dépiler ( $X_1$ . pile - bat)  
Enfiler ( $X_1$ . jeu, C)

Fin

Tantque Non Vide ( $X_2$ . pile - bat) faire  
 $C \leftarrow$  Sommet ( $X_2$ . pile - bat)  
C.v.is.bilité  $\leftarrow$  Faux  
Dépiler ( $X_2$ . pile - bat)  
Enfiler ( $X_1$ . jeu, C)  
fin

Fin

Fonction Bataille (var  $P_1, P_2$  : JOUEUR) : entier  
var C, C2 : CARTE

Début

Si Vide ( $P_1$ . jeu) alors Retourner (2)

Si non Empiler ( $P_1$ . pile - bat, Tête ( $P_1$ . jeu))  
Défiler ( $P_1$ . jeu)

Si Vide ( $P_2$ . jeu) alors Retourner (1)

Si non Empiler ( $P_2$ . pile - bat, Tête ( $P_2$ . jeu))  
Défiler ( $P_2$ . jeu)

Si vide ( $P_1$ .jeu) alors Retourner (1)

Si non

$C_1 \leftarrow \text{Tête}(P_1.\text{jeu})$

$C_1.\text{visibilité} \leftarrow \text{max}$

Empiler ( $P_1$ .pilebat,  $C_1$ )

Défiler ( $P_1$ .jeu)

Si vide ( $P_2$ .jeu) alors Retourner (1)

Si non

$C_2 \leftarrow \text{Tête}(P_2.\text{jeu})$

$C_2.\text{visibilité} \leftarrow \text{max}$

Empiler ( $P_2$ .pilebat,  $C_2$ )

Défiler ( $P_2$ .jeu)

Selon (Compare-Conte( $C_1, C_2$ )) faire

Cas 1 : ~~Renvoyer~~ ( $P_1, P_2$ )

Cas -1 : ~~Renvoyer~~ (2)

Cas 0 :

Renvoyer (Bataille ( $P_1, P_2$ ))

fin Selon

fin

fin

Function jeu (var  $J_1, J_2$  : Joueurs)

Début

Tout-joueurs ( $J_1, J_2$ )

Tantque Non Vide ( $J_1$  : jeu) et Non Vide ( $J_2$  : jeu) fin

Tour ( $J_1, J_2$ )

Finique

Si Vide ( $J_1$  : jeu) alors Retourner (2)

Sinon Retourner (1)

fs

Fin