

Travail pratique POKER MACHINE

15% production d'algorithmes (France): remise du programme général fonctionnel le :

Vendredi 17 novembre 2017

15% programmation 1(Lyne) : remise du travail final et fonctionnel

Vendredi 7 Décembre 2017

À faire seul ou en équipe de deux maximum.



- LE JEU DE POKER MACHINE

En général :

Vous devez concevoir un programme simulant un jeu de "Poker Machine".

Vous aurez besoin d'un jeu de 54 cartes.

Les CARTES sont représentées par : (struct)

string nom : chaîne en majuscules contenant ("AS" pour AS, "2" pour 2, ..., "10" pour dix, "J" pour valet, "Q" pour dame, "K" pour roi et "JK" pour joker)

int valeur : un entier entre 2 et 15 ("J" vaut 11, "Q" vaut 12, "K" vaut 13, "AS" vaut 14 et "JK" vaut 15 les autres sont équivalents à leur nom)

string sorte : chaîne en minuscules pour "trèfle", "carreau", "cœur", "pique" et "jk1" et "jk2" pour les jokers.

bool donne: booléen pour dire si la carte a été donnée ou non.

Un JEU DE CARTES est représenté par un tableau de 54 cartes.

Le champ 'donne' est mis à false pour toutes les cartes lors de la construction du jeu

Nous allons simuler un jeu de poker à l'aide du **fichier testspoker.txt** (ce sera plus facile pour vous de tester votre programme avec des cartes prédéterminées au lieu d'attendre le hasard... Et pour nous plus facile à corriger.... Alors il faut utiliser la même structure de fichier que nous. Modifier les cartes au besoin, mais attention les majuscules et minuscules sont importantes. Il ne faut pas donner une carte plus d'une fois !)

Exemple de fichier

50.00	le montant de la mise pour ce jeu
Q 12 trèfle	les 5 cartes avec leur nom, leur valeur et leur sorte
3 3 carreau	les unes en dessous des autres
3 3 pique	
AS 1 cœur	
2 2 pique	
102.25	le montant de la mise pour le prochain jeu
3 3 trèfle	
3 3 carreau	

3 3 pique
3 3 cœur
JK 15 jkl

Le prochain montant si ce n'est pas la fin de fichier. Donc soit un montant, soit la fin du fichier.

Simulation (faite avec les données ci-haut en exemple)

Nous devrions commencer par demander le montant de la mise à l'utilisateur, mais nous la prendrons dans le fichier testspoker.txt et on l'affiche à l'écran

Votre mise : 50.00\$

On devrait donner une première main de 5 cartes au hasard, mais nous allons la prendre dans un fichier.

Q trèfle	3 carreau...	3 pique...	AS cœur...	2 pique ...
---------------	--------------	------------	------------	----------------

Vous devez trier et vérifier les gains de cette main et afficher la main et les gains à l'écran

2 pique	3 carreau...	3 pique...	Q trèfle...	AS cœur...
-----------------	--------------	------------	-------------	------------

Vos gains : 0.00\$

Puis, on demande à l'utilisateur combien de cartes il veut changer dans cette main (on ne demande qu'une seule fois)

Combien de cartes voulez-vous changer ?

2

(Ce nombre doit être entre 0 et 5. Si ce nombre est 5, on change toutes les cartes sinon on demande quelle(S) carte(S) il veut changer)

Entrez les numéros des 2 cartes à changer

1 5

On prend alors deux cartes au hasard dans le paquet de 54 cartes (2 cartes non encore données) et on change les cartes mentionnées (par exemple le 3 de cœur pour la 1ère et l'AS de pique pour la 4ième)

On trie la main de nouveau, on revérifie les gains et on affiche le tout

Cette fois, on ne demande plus s'il veut changer de cartes. Il n'avait droit qu'à une seule fois

3 cœur...	3 carreau...	3 pique...	AS pique...	AS cœur...
-----------	--------------	------------	-------------	------------

Vos gains : 500.00\$

FULL HOUSE

**Puis on lit de nouveau une mise dans le fichier
Si ce n'est pas la fin du fichier, on recommence le jeu !**

Votre mise : 102.25\$

Etc...

Les tâches générales sont les suivantes :

- 1- Construire le jeu de cartes (tableau de 54 cartes à initialiser avec les 54 cartes possibles. Ne pas oublier d'initialiser le donne à false en début de chaque jeu)**
- 2- Lire dans le fichier la mise du joueur (float représentant les dollars misés)**
- 3- Donner une première main de 5 cartes (et l'afficher à l'écran)
(Tableau de 5 cartes différentes prise dans le fichier testspoker.txt).**
- 4- Trier la main selon les valeurs de carte (plus facile pour trouver les gains)**
- 5- Évaluer cette première main (*voir NOTE au bas du document)**
 - i. (Commencer par ne pas gérer les JK, donc ne pas les donner en random lors du changement de carte) et lorsque votre programme fonctionnera sans les jokers, traiter les jokers comme une frime. À vous de trouver où s'en servir pour trouver le plus gros gain.)**
 - ii. S'il y a gain, afficher le gain possible ainsi que les cartes placées selon ce gain (exemple : si 3 pareilles et 2 pareilles, présenter les cartes pareilles une à côté de l'autre. En fait, selon le tri !)**
- 6- Demander à l'utilisateur combien de cartes il veut changer entre 0 et 5**

Si 0, on ne change aucune carte
Si 5, on change toutes les cartes.
Sinon, il faut demander quelles cartes changer :
L'utilisateur doit alors saisir le numéro de la carte à changer (entre 1 et 5)
Il ne peut changer une carte plus d'une fois
- 7- Redonner de nouvelles cartes pour remplacer celles qui doivent être modifiées.
(peut se faire dans la tâche 6)**
- 8- S'il y a eu changement de carte(s) évaluer de nouveau la main et présenter le gain en affichant également les cartes en ordre du gain.**
- 9- Vérifier si ce n'est pas la fin du fichier en lisant une nouvelle mise. Si ce n'est pas la fin du fichier, on réinitialise le jeu et on repart à la tâche 3**

Les gains possibles selon la main et la mise du joueur :

- | | |
|--|----------------|
| 1. Straight Flush Royale "AS" "K" "Q" "J" "10" de même sorte | = 40 x la mise |
| 2. Straight Flush suite de 5 cartes dans une même sorte | = 30 x la mise |
| 3. 4 cartes pareilles (sortes différentes, bien entendu) | = 25 x la mise |
| 4. Straight suite de 5 cartes sans être dans la même sorte | = 20 x la mise |
| 5. Flush 5 cartes de la même sorte (pas une suite) | = 15 x la mise |
| 6. Full House 2 cartes pareilles + 3 cartes pareilles | = 10 x la mise |
| 7. 3 cartes pareilles | = 5 x la mise |
| 8. 2 paires | = 3 x la mise |

***** À noter que l'évaluation du gain sera plus facile si le tableau contenant la main est trié en ordre croissant sur la valeur de la carte.**

**La commande RANDOM retourne un nombre aléatoire
Voir l'exemple de code donné sur Léa.**