

Programmation - Projet fonctions

1 Consignes

1. Une fois votre projet achevé, vous devez rendre votre fichier C sur “syntax error”.
2. La date de fin du projet est celle fixée sur “syntax error”. Aucune soumission après cette date ne sera acceptée.
3. Il est formellement interdit de recopier des lignes de code d’une tierce personne. Toute tricherie avérée entraînera la note de zéro (l’entraide reste bien sûr autorisée et encouragée).

Restrictions :

1. Les seules fonctions de la libC autorisées sont `printf` et `scanf`.
2. Il est interdit d’utiliser des boucles (Si vous ne savez pas ce que c’est ne vous inquiétez donc pas).

2 Le stud poker

Le stud poker se joue avec un jeu de 52 cartes. Chaque carte a une *valeur* et une *couleur* (pique, coeur, trèfle et carreau). Les valeurs sont ordonnées comme suit, de la moins forte à la plus forte : de 2 à 10, puis valet, dame, roi at as (qui est aussi le 1).

Au stud poker, chaque joueur a cinq cartes en main. Les cinq cartes de la main d’un joueur forment une combinaison. Voici les différentes combinaisons, classées par ordre croissant de valeur :

1. Carte haute : Aucune des combinaisons suivantes :
2. Paire : deux cartes de même valeur.
3. Double paire : deux fois deux cartes de même valeur.
4. Brelan : trois cartes de même valeur.
5. Suite : cinq cartes dont les valeurs se suivent (L’as peut aussi être vu comme le 1, mais il faut choisir : Les valeurs “as - 2 - 3 - 4 - 5” forment une suite, tout comme les valeurs “10 - valet - dame - roi - as”, mais, par exemple, les valeurs “roi - as - 2 - 3 - 4” ne forment pas une suite).
6. Couleur : cinq cartes de même couleur.
7. Full : un brelan et une paire.
8. Carré : quatre cartes de même valeur.
9. Quinte flush : suite dont les cartes ont la même couleur.

Si un joueur a une combinaison meilleure que celle de l’autre joueur, il gagne. Si les deux joueurs ont le même type de combinaison, on procède comme suit :

1. Si les joueurs ont tous les deux une Carte haute, c’est celle dont le rang de la plus haute carte est le plus élevé qui l’emporte. En cas d’égalité on procède de même avec le rang des cartes restantes, en commençant par la plus forte, et en allant jusqu’à la moins forte, tant qu’il y a encore égalité.

2. Si les joueurs ont tous les deux soit un Breton, soit une Paire, soit un Carré, c'est celle dont le rang est le plus élevé qui l'emporte. En cas d'égalité on procède de même avec le rang des cartes restantes, en commençant par la plus forte, et en allant jusqu'à la moins forte, tant qu'il y a encore égalité.
3. Si les joueurs ont tous les deux soit une Suite, soit une Couleur, soit une Quinte flush, c'est celle dont le rang de la plus haute carte est le plus élevé qui l'emporte (pour la suite 1 2 3 4 5, la carte la plus haute est le 5 et non l'as).
4. Si les joueurs ont tous les deux un Full, c'est celui dont le rang du regroupement de 3 cartes est le plus élevé qui l'emporte. En cas d'égalité, on procède de même avec le rang du regroupement de 2 cartes.
5. Si les joueurs ont tous les deux une double paire, c'est celle dont le rang de la paire la plus forte est le plus élevé qui l'emporte. En cas d'égalité, on procède de même avec le rang de la paire la moins forte. En cas d'égalité on procède de même avec le rang de la carte restante.

3 Le travail

Une carte d'un jeu de 52 cartes sera représenté par :

- un `int` compris entre 1 et 13 pour la valeur de la carte (1=as, 11 = valet, 12 = dame et 13 = roi) et
- un `char` pour la couleur de la carte 'P' pour pique, 'C' pour coeur, 'T' pour trèfle et 'K' pour carreau.

Dans un fichier `poker.c`, Écrivez un programme qui lit sur l'entrée standard un entier suivi d'un caractère, le tout dix fois. Chaque entier/caractère est séparé de celui d'après par des espaces. Le premier entier est la valeur de la première carte du joueur 1. Le premier caractère est la couleur de la première carte du joueur 1. Les dix premiers entier/caractères sont les cartes du joueurs 1. Les dix suivants les cartes du joueur 2. On pourra supposer que ce qui est lu sur l'entrée standard est toujours valide. Inutile de vérifier si la même carte apparaît deux fois (par exemple deux 3 de coeur).

Le programme doit ensuite afficher sur une ligne "Player1 wins\n" si le joueur 1 gagne, "Player2 wins\n" si le joueur 2 gagne, et "Draw\n" si il y a égalité.

Voici un exemple d'exécution du programme :

```
$> gcc -o poker poker.c
$> echo "1 P 3 P 10 P 4 P 12 P 2 C 3 K 2 P 3 T 3 C" | ./poker | cat -e
Player2 wins$
$>
```

Dans l'exemple ci-dessus, le premier joueur a une couleur avec toutes ses cartes Pique. Le deuxième a un full avec 2Coeur, 3Carreau, 2Pique, 3Trèfle et 3Coeur. Le full l'emporte sur la couleur et donc le joueur 2 gagne.

Conseils :

1. Ne paniquez pas.
2. Décomposez le problème en sous-problèmes. **Utilisez les fonctions.** Par exemple, une fonction qui retourne la carte de valeur la plus haute, parmi cinq valeurs, puis une autre qui retourne celle qui est la deuxième plus haute, puis une autre pour la troisième plus haute, etc... Cela pourrait vous aider à ordonner les valeurs des cartes d'un jeu.

3. Une possibilité (mais ce n'est pas la seule) est de faire une fonction f qui à une combinaison de cinq cartes, associe une valeur, de telle sorte que pour deux combinaisons c_1, c_2 , on ait $f(c_1) > f(c_2)$ si et seulement si la combinaison c_1 est meilleure que la combinaison c_2 .