

TP 2 : arbres binaires et récursivité

Exercice 1

La course à 20 est un jeu à deux joueurs. Chaque joueur à son tour annonce un chiffre compris entre 1 et 3. Au fur et à mesure, on fait le total des chiffres annoncés : le joueur qui arrive à 20 exactement a gagné.

Exemple

A joue 3	total : 3	
B joue 1	total : 4	
...		
A joue 2	total : 18	
B joue 2	total : 20	B a gagné.

a) Écrire un programme qui permet à un joueur humain de jouer contre l'ordinateur. Pour déterminer le nombre à jouer par l'ordinateur, on utilisera la fonction prédéfinie **rand()** qui délivre un nombre tiré au hasard entre 0 et RAND_MAX valant au moins 32767 (disponible dans la bibliothèque `stdlib.h`), et l'opérateur modulo. Pensez à utiliser **srand()** en début de programme.

Attention : en fin de partie, le programme doit jouer de manière à ne pas dépasser 20 (et le joueur en fait autant).

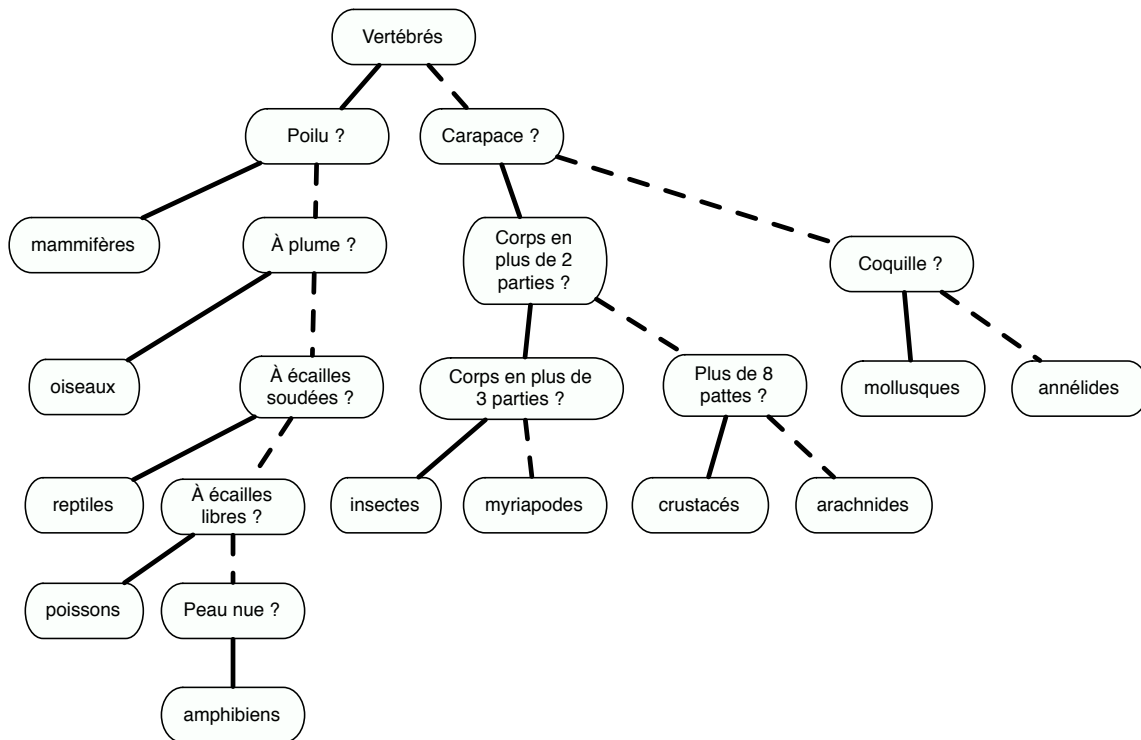
b) Modifier ce programme pour que l'ordinateur joue de manière optimale, en remplaçant le tirage aléatoire par une fonction qui détermine s'il existe une stratégie gagnante.

Exercice 2 – Mais quoi donc est-ce ?

Vous vous rendez chez un ami explorateur du monde qui vous présente des spécimens qu'il a ramené de ses multiples périples, mais comment savoir de quel animal il s'agit ?

On se propose de créer un programme permettant de répondre à cette question.

À l'aide d'une classification présentée sous forme arborescente comme suit, le programme devra déterminer l'espèce de l'animal en posant les questions adéquates.



Travail à faire :

a) Écrire un programme qui permet d'identifier l'animal que vous avez entre les mains. Vos choix porteront sur la structure de l'arbre : si la réponse est négative, on ira vers le fils droit, sinon vers le fils gauche.

b) Écrivez une fonction d'affichage en mode terminal donnant un résultat similaire à ce qui suit :

```
[ ] Est-il vertébré ?
  [oui] Est-il poilu ?
    [ ] [oui] C'est une vache !
    [ ] [non] A-t-il des plumes ?
      [ ] [oui] C'est un aigle !
      [ ] [non] A-t-il des écailles soudées ?
        [ ] [oui] C'est un crocodile !
        [ ] [non] A-t-il des écailles libres ?
          [ ] [oui] C'est un poisson-clown !
          [ ] [non] Est-il à peau nue ?
            [ ] [oui] C'est une grenouille !
            [ ] [non] A-t-il une carapace ?
              [ ] [oui] Son corps a-t-il plus de 2 parties ?
                [ ] [oui] Son corps a-t-il plus de 3 parties ?
                  [ ] [oui] C'est une abeille !
                  [ ] [non] C'est un scolopendre !
                  [ ] [non] A-t-il plus de 8 pattes ?
                    [ ] [oui] C'est un crabe !
                    [ ] [non] C'est une araignée !
                    [ ] [non] A-t-il une coquille ?
                      [ ] [oui] C'est une moule !
                      [ ] [non] C'est un lombric !
```

c) Complétez le programme précédent de manière à enrichir l'arbre des questions lorsque la solution n'a pas été trouvée. Le programme doit alors demander le nom de l'animal auquel pense le joueur, ainsi qu'une question permettant de différencier les deux animaux, puis modifier l'arbre pour y ajouter cette question et le nom de l'animal.

Exemple

Est-il vertébré ? **O**

Est-il poilu ? **O**

C'est une vache ! **N**

Je donne ma langue au chat.

Quelle est la réponse ? **C'est un lion !**

Donnez une question dont la réponse est oui pour **C'est un lion !** et non pour **C'est une vache !** : **Est-ce un carnivore ?**

Il suffit ensuite d'intégrer la question et la réponse au bon endroit dans l'arbre (qu'il faudra probablement réorganiser).

Fichiers fournis

- Les fichiers **arbre.c** et **arbre.h** contenant les primitives d'accès à un arbre binaire de chaînes de caractères.
- Le fichier **animaux.c** contient le squelette du programme et une fonction qui initialise l'arbre des questions en lisant le fichier **classif_animaux.txt**.