

1 document A5 autorisé, avec rappel de cours officiel en recto et notes personnelles manuscrites au verso.

Donnez vos codes avec la syntaxe réduite favorisant la déclarativité pour ProLog / Erlang comme celle utilisée en cours

(par ex., pour Prolog : avec contraintes, mais sans coupure, **is** ou négation ;

pour Erlang : avec des gardes, mais sans **if**, **case**, ni liste en compréhension)

I. Sur la base du projet – en ProLog (barème indicatif : 7 points)

Rappel du projet : le projet est basé sur le jeu classique "pierre/feuille/ciseaux" . Les stratégies du jeu sont des prédicats ayant le profil suivant :

%% nomStratégie(en entrée : une liste Historique, en sortie : un coup CoupProposé)

%% rem. l'historique est la liste des manches jouées

%% ex. d'historique : [[pierre,feuille],[pierre,pierre],[feuille,ciseaux]]

%% pour faciliter l'accès à l'historique, la dernière manche jouée est en premier

%% en premier dans une manche, le coup joué par l'humain, en second celui de l'IA

Pour rappel :

jouerUneManche(StrategieHumain, StrategieIA, Historique, [CoupHumain,CoupIA]):

call(StrategieHumain, Historique, CoupHumain),

call(StrategieIA, Historique, CoupIA).

Q1.1 Spécifiez et réalisez un programme qui inverse les rôles Joueur/IA dans l'historique. Pour l'exemple précédent, cela donne l'historique : [[feuille,pierre],[pierre,pierre],[ciseaux,feuille]]

Q1.2 En utilisant la question précédente, donner le code pour faire jouer une manche pour une stratégieIA jouant contre elle-même.

Q1.3 Spécifiez et réalisez une stratégieIA qui fait la réponse la plus courante d'une liste de stratégiesIA donnée.

II. Indexer² - en ProLog ou en Erlang (barème indicatif : 7 points)

Au choix, vous réaliserez cet exercice en ProLog ou en Erlang.

Avant de répondre à la première question, lire l'ensemble de l'énoncé, penser à une solution (ou deux) et choisir ProLog ou Erlang pour faire ces solutions. Indiquer votre choix, ProLog ou Erlang, clairement, au début de chaque solution.

Q II.1 Indexer². Spécifiez et réalisez un programme qui indexe deux fois une liste en fournissant l'ordre de chaque élément dans la liste en partant du début et en partant de la fin.

Par ex. sur la liste [a, b, c, d, e], votre algorithme peut fournir la liste suivante qui a associé à chaque élément un triplet commençant par l'index en partant du début, puis donne l'élément lui-même et enfin donne l'index en partant de la fin : [[0, a, 4], [1, b, 3], [2, c, 2], [3, d, 1], [4, e, 0]]

Q II.2 Indexer²-bis. Trouver une seconde méthode, significativement différente de la première (cf. question précédente), pour réaliser votre programme. Spécifiez et réalisez cette seconde solution en indiquant ce qui la distingue de la méthode précédente.

Proposition de variantes pour la seconde méthode :

- analyser le problème selon les données vs selon les résultats
- se baser sur la production d'un premier index pour produire l'autre index vs échanger les rôles des deux index, ou faire avec les 2 en même temps
- utiliser prolog, avec ou sans contraintes ; ou erlang, avec ou sans fonctions d'ordre supérieur
- choisir prolog pour une méthode, erlang pour la seconde méthode (i.e. : faire une solution dans les 2 langages)
- utiliser un accumulateur pour introduire une récursivité terminale
- etc.

Q II.3 Correct-Complet-Termine. Prouvez que votre premier programme est correct, complet et qu'il termine.

III. Triangle ou pyramide ? (barème indicatif : 6 points)

Les nombres triangulaires sont les entiers $N*(N+1)/2$, ils peuvent être obtenus par la somme $1+2+3+...+N$ (avec N termes). En partant du N^e nombre triangulaires, on peut passer au $(N+1)^e$ nombre triangulaires en ajoutant $N+1$, i.e. : $triangle(N+1)=triangle(N)+N+1$.

O	O
O O	+
O O O	O O
O O O O	O O

Les nombres pyramidaux sont les entiers $N*(N+1)*(2N+1)/6$, ils peuvent aussi être obtenus par la somme $1+2^2+3^2+...+N^2$ (avec N termes). En partant du N^e nombre pyramidal, on peut passer au $(N+1)^e$ nombre pyramidal en ajoutant $(N+1)^2$, i.e. : $pyramide(N+1)=pyramide(N)+(N+1)^2$.

Exemple de nombres triangulaire ou pyramidal, ici $triangle(4)=10$ et $pyramide(2)=5$

Les premiers nombres triangulaires ou pyramidaux sont donc : 0, 1, 3 (triangle), 5 (pyramidal), 6 (triangle), 10 (triangle), 14 (pyramidal), 15 (triangle), etc.

Problème : quel est le plus grand nombre, triangulaires ou pyramidal, que l'on peut écrire avec 9 chiffres (i.e. entre 100 000 000 et 999 999 999).

Q III.1 Résolution séquentielle : En utilisant la définition des nombre triangulaires ou pyramidaux que vous préférez (parmi les définitions données en introduction), spécifiez et programmez en Erlang une fonction réalisant la résolution du problème.

Q III.2 Parallélisation sur 2 cœurs : Paralléliser la résolution du problème en organisant les calculs précédents pour les exécuter sur une machine ayant au moins 2 cœurs. Par exemple, en cherchant sur un cœur le plus grand nombre triangulaire et en parallèle en cherchant sur l'autre cœur le plus grand nombre pyramidal.