

Documentation des fonctions

Exercice 1 - ajouter_nœud

On invite l'utilisateur par un message d'entrez la liste des nœuds suivi d'un deuxième message l'invitant a entrez l'index du nœud.

Si un nœud existe dans la base des fait avec cette index un message d'erreur sera afficher et le processus recommence, sinon on demande au utilisateur de donner le nom de la personne, un message de confirmation de l'insertion suivi d'un message pour commencer le processus ou quitte.

Exercice 2 - supprimer_nœud

on invite l'utilisateur par un message de tape l'index du nœud à supprimer,
s'il existe on le supprime puis on affiche un message de confirmation en suite on supprime tout les lien associer avec ce nœud. Sinon on affiche un message que le Lien introuvable .

Exercice 3 - ajouter_lien

on invite l'utilisateur par un message a entrez la liste des liens suivi d'un deuxième message l'invitant a entrez l'index du premier nœud.

Si le nœud n'existe pas on afficher un message d'erreur, si existe un message l'invite a entre l'index du deuxième nœud .Si les deux nœud existant on cherche si il existe un lien entre ces nœud dans les deux sens, si il existe on afficher un message d'erreur et recommence le processus , sinon le lien sera ajouter un message de confirmation sera afficher suivi d'un message pour commencer le processus ou quitté.

Exercice 4 - supprimer_lien

on invite l'utilisateur par un message d'entrez l'index du premier nœud suivi d'un deuxième message l'invitant a entrez l'index du deuxième nœud .

si il existant un lien du premier nœud vers le deuxième ou le contraire , on le supprime suivi d'un message de confirmation, sinon on afficher un message que le lien n'existe pas .

Exercice 5 - verifier

le prédicat vérifier tout les liens dans la base des fait ,est pour chaque lien il valide l'existence des nœud qui le constitues , si il arrive avec un lien qui réfère a le même nœud ou nœud inexistant il affiche un message d'erreur sinon un message de validité est afficher.

Exercice 6 amitie(X,Y)

le prédicat amitié a une arité de 2 il retourne true ou false , le prédicat cherche pour chaque nom le nœud dans la base des faits

si true il vérifier avec leur index si un lien d'amitié existe dans les deux direction si n'existe pas il chercher si un intermédiaire existe entre les deux si il ne trouve rien le troisième prédicat inverse les nom pour recommence une nouvelle recherche .

Exercice 7 - commun(X,Y,Z)

prédicat « commun » a une arité 3 deux nom est une variable pour le résultat retourner pour chaque nœud il va chercher son index ,pour les la variable il va la matcher avec tous les nœud pour vérifier ensuite si il existe un lien entre le premier nœud et le la variable et le deuxième nœud et la variable , si existe on le retourne sinon retourne false.

Exercice 8 - intermediaires(X, Y,N)

le prédicat « intermédiaire » d'arité 3 , deux nom et une variable pour le retour on cherche les nœud des deux nom si ils existant on vérifier si il y a un lien direct entre eux si oui retourner 0.

si non le deuxième prédicat cherche le nom du premier nom et un nœud de la base des faits puis vérifier si il y a un lien si oui il rappelle le prédicat récursivement avec le nouveau nom et le même deuxième nom reçu en paramètre jusqu'à il tombe dans le premier prédicat notre point d'arrêt on suit il commence a compter le nombre des intermédiaires entre deux nom ce qui va être retourner dans la variable

Exercice 9 - suggestion(X,Y)

le prédicat « suggestion » a une arité 2 prend comme argument des nom chercher les nœud correctement puis fait appelle au prédicat intermédiaire qui lui retourner la liste des amis intermédiaire puis le prédicat sélectionner les distance supérieur a 1 et inferieur ou égale a 3 ce qui va être retourner sinon false.