Question 1.

Soit E={q|q(3) est défini}

Pourquoi E n'est pas calculable ? (réponse en deux lignes maximum) ?

Montrer que E est récursivement énumérable.

Soit F={q|q(3) n'est pas défini}

F est-il calculable, récursivement énumérable, non récursivement énumérable ?

Corrigé :

E n'est pas calculable car la fonction caractéristique a la même valeur que le prédicat P défini de la façon suivante P(q)=1 si et seulement si q(3) est défini or ce prédicat n'est pas trivial donc il est indécidable (théorème de Rice).

Pour montrer que E est récursivement énumérable on peut
- soit montrer que l'on peut afficher les éléments de E :
  Pour tout couple (q, t) faire si h(q,3,t) alors afficher q ;
- soit donner la fonction semi-caractéristique :
  int fsc (int q) { q(3) ; return 1 ; }

F est le complémentaire de E. Donc si F est r.é. comme E est aussi r.é on est déduirait que E est calculable (théorème du cours : si E et son complémentaire sont r.é alors E est calculable) ce qui est fautx. Donc F n'est pas récursivement énumérable.

Question 2.

Soient f et g, 2 fonctions calculables. E={x | f(x)*g(x) = 0}.

Donner un algorithme qui affiche les éléments de E.

Corrigé :

Pour tout couple (x,t) faire si (h(f,x,t) et h(g,x,t) et (f(x)=0 ou g(x)=0)) alors afficher x ;
Le point délicat est que f et g ne sont pas forcément totales donc
Pour tout x faire si (f(x)=0 ou g(x)=0) alors afficher x ne fonctionne pas.

Question 3.

Soient A et B deux ensembles A est décidable et B est récursivement énumérable mais non décidable.

Donner un exemple de A et B tels que :

AnB décidable : On prend A=∅

AuB décidable : On prend A=N

AnB récursivement énumérable mais non décidable : On prends A=N

AuB récursivement énumérable mais non décidable : On prends A= ∅

Question 4.

Le problème A se réduit polynomialement au problème B.

donc A est plus facile que B

Question 5.

Si on a un algorithme en $0(n^3)$ et que l'on multiplie par 1000 la puissance d'un ordinateur. La taille des problèmes que l'on peut traiter par unité de temps est multipliée par 10.

C'est vrai car $10^3 = 1000$. Donc si on multiplie la taille de la donnée par 10 le temps est multiplié par 1000 sur le même ordinateur et la même implémentation. Donc si la puissance de l'ordinateur est multipliée par 1000 on retrouve le même temps.

Question 6.

Soient G=(V,E), 2 sommets x et y de G et un entier k.

Question : Existe-t-il un chemin de longueur k dans G qui ne passe que par des sommets différents de G ?

Montrer que ce problème est NP-Complet.

Corrigé :

Chemin hamiltonien :
Données : G=(V,E), 2 sommets x et y de G.
Question : Existe-t-il un chemin de x à y passant par tous les sommets une et une seule fois ?

On a vu en cours et en TD que ce problème est NP-complet. Or ce problème est un cas particulier du problème car il suffit de prendre k=n-1. Le problème est donc plus compliqué qu'un problème NP-complet. Or il est dans NP car si on donne un chemin de x à y on peut vérifier s'il a les bonnes propriétés en temps polynomial (longueur k, 2 sommets consécutifs sont bien reliés dans G et chaque sommet apparaît au plus une fois). Il est donc NP-complet.