TD 1S3 Etude et comparaison d’algorithmes année 2012-2013

**Partie I :**

Dans cet exercice, k et n sont deux entiers.

On appelle **liste** un couple de deux entiers n et k sans importance d’ordre (la liste {n,k} est identique à la liste {k,n})

1. On donne l'algorithme suivant :

c prend la valeur 0

Pour k variant de 0 à 5

Pour n variant de 0 à 5

c prend la valeur c + 1

Si k² + n² ≤ 25

Alors

afficher la liste {k,n}

Fin du Si

Fin du Pour

Fin du pour

afficher c

1. Parmi les listes suivantes, dire lesquelles sont affichées par cet algorithme ( barrer ceux qui ne conviennent pas ): {1 ; 4} {3 ; 4} {4 ; 4} {4 ; 0}
2. Quelle est la valeur de c affichée à la fin de l'exécution de l'algorithme ?
3. Comment modifier la ligne 3 pour ne pas faire apparaître la même liste deux fois ?
4. Comment interpréter graphiquement le fonctionnement de cet algorithme dans un carré de côté 5 ?

**Partie 2**

1. On se place désormais dans un repère orthormé d’origine O , avec x ≥ 0 et y ≥ 0.

A l’aide de l’équation du cercle de centre O et rayon 5, exprimer l’ordonnée d’un point du quart de cercle en fonction de son abscisse.

1. Soit la fonction *f* définie sur [0 ; 5] par *f*(*x*) =
   1. Justifier le fait que 25 – *x*² est positif sur cet intervalle.
   2. Construire le tableau de variations de *f* sur [0 ; 5].
2. On donne ci-dessous, la courbe d'équation y = . Ecrire sur le graphique toutes les listes affichées par l’algorithme donné au début du TD. Combien y en a-t-il? Combien y en t-il de différents ?



**Partie 3**

1. Pour k entier variant de 0 à 5 , déduire de l’étude précédente dans quel intervalle doit varier l’entier n vérifiant à la fois la condition : k² + n² ≤ 25 et la condition de ne pas compter deux fois les listes identiques.
2. Construire un algorithme permettant d’afficher les listes de nombres entiers { k ; n } vérifiant la propriété suivante: k² + n² ≤ 25 sans utiliser la condition SI .
3. Programmer les deux algorithmes précédents( partie 1 avec modification ligne 3 et celui ci-dessus), sur calculatrice, puis sur le logiciel « ALGOBOX » , puis sur le logiciel Xcas ( s’aider du manuel ) .
4. Comparaison des deux algorithmes ( ouverture ):

« *le deuxième algorithme permet une économie du nombre d’étapes ( boucles) mais demande des calculs plus compliqués ( racine carrée)* »:

On teste à l’aide du logiciel Python la rapidité des deux algorithmes ( l’un par balayage , l’autre par description) … les calculs sont si rapides que l’on fait une étude plus générale avec un quart de cercle de grand rayon …

Que constate-t-on expérimentalement ?