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 ?

C compte le nombre de boucles faites pour chaque k et chaque n, donc 6x6 = 36 boucles car 6 possibilités de 0 à 5 pour chacun k puis 6 possibilités pour n. Donc la valeur de c affichée est c= 36.

On retrouve graphiquement ce résultat en considérant les points à coordonnées entières dans un carré de côté 5.

1. Comment modifier la ligne 3 pour ne pas faire apparaître la même liste deux fois ?

Modification ligne 3 : Pour n variant de k à 5, ( il est inutiles de garder les valeurs de n inférieure à k car les listes ont déjà été comptabilisées dans les boucles précédentes )

En effet , graphiquement il suffit de garder les points au dessus ou sur la diagonale.

1. Comment interpréter graphiquement le fonctionnement de cet algorithme dans un carré de côté 5 ?



Le test «  si k² + n² ≤ 25 » teste graphiquement si

un point de coordonnées ( k ; n ) appartient ou

non au cercle de centre 0 et de rayon 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.

M(x ;y) appartient au cercle de centre O et de rayon 5 ⇔ OM = 5

⇔ OM² = 25

⇔ x² + y² = 25

⇔ y² = 25 – x²

Puisque de plus x ≥ 0 et y ≥ 0 ,

M est dans le quart du cercle supérieur droit ⇔ y =

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.

|  |  |
| --- | --- |
| x | -∞ -5 5 +∞ |
| 25-x² | - 0 + 0 - |

25 – x² est un trinôme avec deux racines évidentes : -5 et 5 avec a négatif, donc 25 – x² est positif sur [ 0 , 5 ]

* 1. Construire le tableau de variations de *f* sur [0 ; 5].

|  |  |
| --- | --- |
| x | -∞ 0 +∞ |
| 25-x² |  |

25 – x² est un trinôme avec pour sommet de la parabole : le point d’abscisse 0 et a négatif ( parabole tournée vers le bas).

D’après le cours u et ont les mêmes variations donc :

|  |  |
| --- | --- |
| x | 0 5 |
| 25-x² | 25  0 |
| f(x)= | 5  0 |

1. 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 ?



25 listes affichées avec le premier algorithme : 25

15 listes différentes

{ 0 ; 0 } , { 0 ; 1 } ; …

**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.

n varie de k à 

1. 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 .

c prend la valeur 0

Pour k variant de 0 à 5

Pour n variant de k à 

afficher la liste {k,n}

Fin du Pour

Fin du pour

afficher c

remarque : dans la programmation avec les trois logiciels ci-dessous le  « partie entière » : E () n’a pas été nécessaire, donc je ne l’ai pas envisagé avec les élèves)

1. 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 ) .

|  |
| --- |
| Sur TI  PROGRAM : IREMSOB2  : 0→C  :For(K,0,5)  :→R  :For(N,K,R)  :C+1→C  :Disp “.”,K,N  :Pause  :End  :End  :Disp  « C= »,C |

Sur TI

PROGRAM : IREMSOB1

: 0→C

:For(K,0,5)

:For(N,0,5)

:C+1→C

:If K²+N²≤25

:Disp “.”,K,N

:Pause

:End

:End

:Disp  « C= »,C

Remarque : pour un début d’année, je n’ai pas utilisé les listes de la calculatrice.

1. 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 … les résultats sont vidéoprojetés ( fichier joint )

On constate expérimentalement que c’est le deuxième algorithme sans les tests avec moins de boucles qui est plus rapide. L’explication est à chercher auprès d’informaticiens ……… !!!!