

Algorithmique et Programmation

Examen sur machine

G1: monasse(at)imagine.enpc.fr G2: courchay(at)imagine.enpc.fr
G3: eric.bughin(at)gmail.com G4: david.ok(at)imagine.enpc.fr
G5: francois.olivier.c.h(at)gmail.com G6: vialette(at)univ-mlv.fr

11/12/09

1 Enoncé

1.1 Spirale d'Ulam

Le mathématicien Stanislav Ulam, dans une période d'ennui lors d'une conférence, prit son stylo et machinalement écrivit les nombres entiers en spirale sur une feuille de papier, puis noircit les cases contenant un nombre premier. Des diagonales comportant un nombre remarquable de cases noires apparurent, que jusqu'à présent personne n'a pu expliquer. Le but est de reproduire la découverte d'Ulam sur machine.

On rappelle qu'un nombre premier n'a par définition que 1 et lui-même comme diviseurs, l'exception étant le nombre 1 qui n'est pas premier.

1.2 Travail demandé

Le travail se compose de deux parties. Dans la première, on va chercher les nombres premiers par un crible d'Erastosthène. Dans la seconde, on va dessiner la spirale par un robot qui se déplace et remplit sa case si celle-ci correspond à un nombre premier.

Consignes importantes : Il est plus important de livrer un code propre mais qui compile sans warning, même s'il ne répond pas à toutes les questions. Pour cela, vérifiez à chaque étape que votre programme compile et se lance correctement.

1. Créez un nouveau projet Imagine++ nommé `Ulam` et écrivez le `main`, qui demande à l'utilisateur une taille de fenêtre carrée et crée un tableau de variables booléennes `premier` contenant autant d'éléments que le nombre de cases. L'élément d'indice i du tableau sera `true` si le nombre i est premier.
2. Dans des fichiers *séparés*, créez des fonctions `crible` et `spirale`, qui pour l'instant ne font rien (nous les remplirons plus tard). Appelez ces fonctions depuis le `main`. A ce stade, vérifiez déjà que votre programme compile et se lance correctement, même s'il ne fait pas beaucoup...

1.2.1 Crible

3. Créez une fonction `init` qui initialise le tableau `premier` en mettant a priori tous les indices à `true` sauf 1.
4. Créez une fonction `barre_multiples` qui met à `false` les multiples (autres que lui-même) d'un nombre i dans `premier`.
5. Écrivez la fonction `crible`, qui appelle `barre_multiples` pour tous les nombres premiers.

1.2.2 Robot

6. Dans la fonction `spirale`, créez un tableau pour mémoriser le fait qu'une case a été visitée ou non. L'initialiser.

21	22	23	→	
20	7	8	9	10
19	6	1	2	11
18	5	4	3	12
17	16	15	14	13

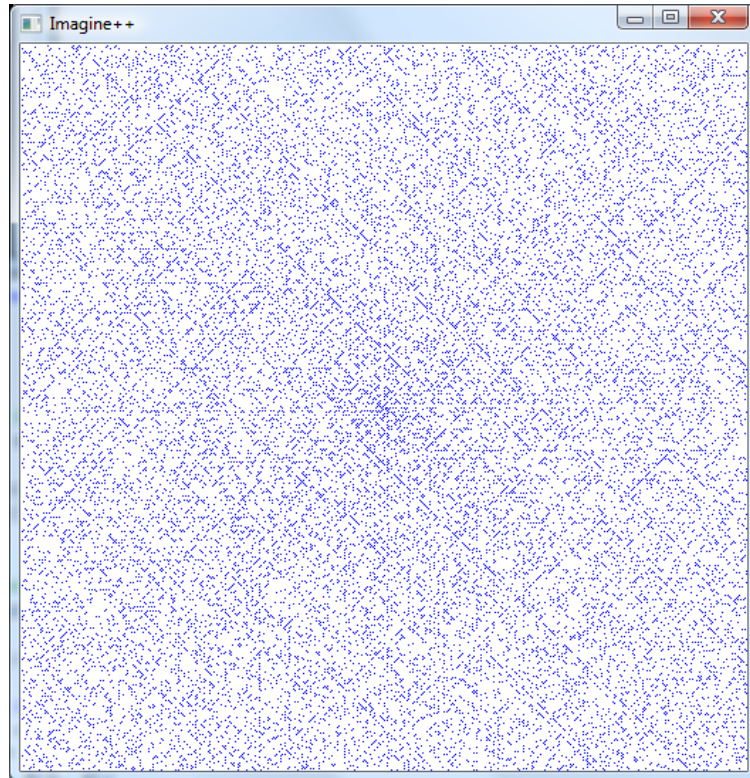


FIG. 1 – Ordre des cases parcourues par la spirale d’Ulam et résultat sur une fenêtre 512×512 .

7. Pour dessiner la spirale, on va écrire un robot qui part du centre de la fenêtre, a une direction initiale, et cherche à tourner à droite dès que la case n’a pas été visitée. Créez une structure **Robot** qui a une position et une direction codée sur un entier, qui prendra une valeur entre 0 et 3.
8. Créez la fonction **avance** qui fait bouger le robot d’une case dans sa direction.
9. Créez la fonction **tourne** qui fait tourner le robot sur sa droite si la case n’a pas été visitée. Cette fonction indique en retour si le déplacement a pu s’effectuer.
10. Dans la fonction **spirale**, faire partir le robot du centre de la fenêtre et se déplacer en appelant **tourne**, ou **avance** si le quart de tour s’est soldé par un échec. Le robot noircit sa position si elle correspond à un nombre premier.

Important : Quand vous avez terminé, créez une archive du projet à votre nom. Ne vous déconnectez pas avant que le surveillant ne soit passé vous voir pour copier cette archive sur clé USB.