

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
2 // ===== EXAMEN MACHINE 2009/2010 =====
3
4 // -- Fichier crible.h --
5 #pragma once
6
7 void crible(bool premier[], int N);
8
9 // -- Fichier crible.cpp --
10 #include "crible.h"
11
12 void init(bool premier[], int N)
13 {
14     premier[1] = false;
15     for(int i=2; i <= N; i++)
16         premier[i] = true;
17 }
18
19 void barre_multiples(bool premier[], int N, int i)
20 {
21     for(int j=2*i; j <= N; j+=i)
22         premier[j] = false;
23 }
24
25 void crible(bool premier[], int N)
26 {
27     init(premier, N);
28     for(int i=1; i <= N; i++)
29         if(premier[i])
30             barre_multiples(premier, N, i);
31 }
32
33 // -- Fichier spirale.h --
34 #pragma once
35
36 void spirale(bool premier[], int N);
37
38 // -- Fichier spirale.cpp --
39 #include "spirale.h"
```

```

40 #include "Imagine/Graphics.h"
41 using namespace Imagine;
42
43 int dx[4] = {+1, 0, -1, 0};
44 int dy[4] = {0, +1, 0, -1};
45
46 struct Robot {
47     int x, y;
48     int dir;
49 };
50
51 // Essaie de tourner a droite et d'avancer le robot si la case est libre
52 bool tourne(Robot& r, bool occupe[], int N)
53 {
54     Robot s;
55     s.dir = (r.dir+1)%4;
56     s.x = r.x + dx[s.dir];
57     s.y = r.y + dy[s.dir];
58     if(occupe[s.y*N+s.x])
59         return false;
60     r = s;
61     return true;
62 }
63
64 // Avance le robot dans sa direction
65 void avance(Robot& r)
66 {
67     r.x += dx[r.dir];
68     r.y += dy[r.dir];
69 }
70
71 // Dessine la spirale d'Ulam.
72 void spirale(bool premier[], int N)
73 {
74     OpenWindow(N,N);
75
76     bool* occupe = new bool[N*N]; // Liste des cases deja remplies
77     for(int i=0; i < N*N; i++)
78         occupe[i] = false;
79     Robot r = {N/2, N/2, 3};
80
81     for(int i=1; true; i++) {
82         if(r.x < 0 || N <= r.x || r.y < 0 || N <= r.y)
83             break;
84         occupe[r.y*N + r.x] = true;
85         if(premier[i])
86             FillRect(r.x, r.y, 1, 1, Blue);
87         if(! tourne(r, occupe, N))
88             avance(r);
89     }
90
91     delete[] occupe;
92     Terminate();
93 }
94
95 // -- Fichier Ulam.cpp --
96 // Imagine++ project
97 // Project: Ulam
98 // Author: Pascal

```

```
99 // Date:      2009/12/3
100
101 #include "crible.h"
102 #include "spirale.h"
103 #include <iostream>
104 using namespace std;
105
106 int main()
107 {
108     int N;
109     cout << "Entrez la taille de la fenetre :" << endl;
110     cin >> N;
111     bool* premier = new bool[N*N+1]; // Indice 0 non utilise
112     crible(premier, N*N);
113     spirale(premier, N);
114     delete[] premier;
115     return 0;
116 }
117
```