

# Algorithmique et Programmation

## Examen sur machine - Solution

G1: keriven(at)cermics.enpc.fr    G2: Jean-Philippe.Pons(at)sophia.inria.fr  
G3: charpiat(at)clipper.ens.fr    G4: thomas.deneux(at)ens.fr  
G5: pierre(at)senellart.com        G6: gmellier(at)melix.org

09/01/04

### 1 Crible d'Ératosthène

Fichier main.cpp

```
1 // question 1
2 #include <iostream>
3 using namespace std;
4
5 // question 3
6 void multiples(bool p[], int n, int a)
7 {
8     for (int i=2*a; i<n; i+=a)
9         p[i] = false;
10 }
11
12 int main()
13 {
14     // question 2
15     const int N = 1000;
16     const int M = 200;
17     const int L = 40;
18     bool prem[N];
19     for (int i=0; i<N; i++)
20         prem[i]=true;
21
22     // question 4
23     for (int i=2; i<L; i++){
24         if (!prem[i])
25             continue;
26         multiples(prem,N,i);
27     }
28
29     // question 5
30     int k=0;
31     for (int i=1; i<M; i++){
32         if (!prem[i])
33             continue;
34         ++k;
35         cout << i << ", ";
36     }
37     cout << endl << "Il y a " << k << " nombres premiers entre 0 et " << M << endl;
```

```
38     return 0;
39 }
```

## 2 Calcul de $\pi$ par la méthode de Monte Carlo

Fichier point.h

```
1  #pragma once
2
3  struct PointPlan {
4      double x;
5      double y;
6  };
7
8  PointPlan GenerePoint();
9  void AffichePoint(PointPlan p,int taille_fenetre,Color c);
10 double Norme2(PointPlan p);
```

Fichier point.cpp

```
1  #include <win>
2  #include <cstdlib>
3  using namespace Win;
4  using namespace std;
5
6  #include "Point.h"
7  PointPlan GenerePoint() {
8      PointPlan p;
9
10     p.x=rand()/double(RAND_MAX)*2-1;
11     p.y=rand()/double(RAND_MAX)*2-1;
12
13     return p;
14 }
15
16 void AffichePoint(PointPlan p,int taille_fenetre,Color c) {
17     DrawPoint(int(p.x*taille_fenetre/2+taille_fenetre/2),
18             int(p.y*taille_fenetre/2+taille_fenetre/2),
19             c);
20 }
21
22 double Norme2(PointPlan p) {
23     return p.x*p.x+p.y*p.y;
24 }
```

Fichier main.cpp

```
1  #include <win>
2  #include <cstdlib>
3  #include <ctime>
4  using namespace Win;
5  using namespace std;
6
7  #include "Point.h"
8
9  const int taille_fenetre=512;
10 const int nb_iterations=50000;
11
12 int main()
13 {
14     srand(unsigned int(time(NULL)));
15     OpenWindow(taille_fenetre,taille_fenetre);
```

```

16     DrawCircle(taille_fenetre/2,taille_fenetre/2,taille_fenetre/2,Black);
17
18     int compteur=0;
19
20     for(int i=0;i<nb_iterations;++i) {
21         PointPlan p=GenerePoint();
22         if(Norme2(p)<=1) {
23             AffichePoint(p,taille_fenetre,Blue);
24             ++compteur;
25         } else
26             AffichePoint(p,taille_fenetre,Red);
27
28     }
29
30     cout << "Pi vaut a peu pres (" << nb_iterations << " itérations) : "
31         << 4.*compteur/nb_iterations << endl;
32     Terminate();
33     return 0;
34 }

```

### 3 Serpent

Fichier main.cpp

```

1  #include <win>
2  using namespace Win;
3  #include <iostream>
4  using namespace std;
5  #include <cstdlib>
6  #include <ctime>
7  // question 1
8  struct point
9  {
10     int x;
11     int y;
12 };
13 // question 2a
14 void dessine(point p, int zoom, Color c)
15 {
16     FillRect(p.x*zoom,p.y*zoom,zoom-1,zoom-1,c);
17 }
18 // question 2b
19 void dessine(point s[],int n, int zoom)
20 {
21     dessine(s[0],zoom,Blue);
22     for (int i=1; i<n; i++)
23         dessine(s[i],zoom,Red);
24 }
25 // question 3a
26 bool operator==(point a, point b)
27 {
28     return (a.x==b.x && a.y==b.y);
29 }
30 // question 3b
31 bool cherche(point s[], int n, point p)
32 {
33     for (int i=0; i<n; i++)
34         if (s[i]==p)
35             return true;
36     return false;
37 }

```

```

38 // question 3c
39 void decale(point s[], int n, point p)
40 {
41     for (int i=n-1; i>0; i--)
42         s[i]=s[i-1];
43     s[0]=p;
44 }
45 // question 3d
46 point operator+(point a, point b)
47 {
48     point p = {a.x+b.x,a.y+b.y};
49     return p;
50 }
51 // question 4a
52 const point dir[4] = {{1,0},{0,1},{-1,0},{0,-1}};
53 // question 4b
54 void avance(point s[], int n, int d, int zoom)
55 {
56     dessine(s[n-1],zoom,White);
57     point p = s[0] + dir[d];
58     decale(s,n,p);
59     dessine(s[0],zoom,Blue);
60     dessine(s[1],zoom,Red);
61 }
62 // question 5
63 bool sort(point a, int w, int h)
64 {
65     return (a.x<0 || a.x>=w || a.y<0 || a.y>=h);
66 }
67 void init_rand()
68 {
69     srand((unsigned int)time(0));
70 }
71 void change_dir(int& d)
72 {
73     int h=rand()%20;
74     if (h==0)
75         d=(d+1)%4;
76     else if (h==1)
77         d=(d+3)%4;
78 }
79 bool ok_dir(point s[], int n, int d, int w, int h)
80 {
81     point p = s[0]+dir[d];
82     // return (!sort(p,w,h));
83     // question 6
84     return (!sort(p,w,h) && !cherche(s,n,p));
85 }
86 // question 7
87 bool coince(point s[], int n, int w, int h)
88 {
89     for (int d=0;d<4;d++)
90         if (ok_dir(s,n,d,w,h))
91             return false;
92     return true;
93 }
94 // question 8
95 void allonge(point*& s, int& n, int d, int zoom)
96 {
97     point* ns=new point[n+1];
98     for (int i=0;i<n;i++)

```

```

99     ns[i]=s[i];
100    delete[] s;
101    s=ns;
102    n++;
103    point p=s[0]+dir[d];
104    dessine(s[0],zoom,Red);
105    dessine(p,zoom,Blue);
106    decale(s,n,p);
107 }
108 int main()
109 {
110     const int w=40;
111     const int h=30;
112     const int z=10;
113     OpenWindow(w*z,h*z);
114     // question 2c
115     //const int n=10;
116     //point ex[n]={1,2},{1,3},{1,4},{2,4},{3,4},{4,4},{4,5},{4,6},{4,7},{4,8}};
117     //dessine(ex,n,z);
118     // question 8
119     int n=10;
120     point* ex = new point[n];
121     for (int i=0; i<n; i++){
122         ex[i].x = 5;
123         ex[i].y = 5+n-i;
124     }
125     // question 5
126     const int nbp=50000;
127     init_rand();
128     int d=0;
129     //for (int i=1; i<=nbp; i++){
130     for (int i=1; i<=nbp && !coince(ex,n,w,h); i++){ // question 7
131         do {
132             change_dir(d);
133         } while(!ok_dir(ex,n,d,w,h));
134         // question 8
135         if (!(i%20))
136             allonge(ex,n,d,z);
137         else
138             // fin question 8
139             avance(ex,n,d,z);
140         MilliSleep(1);
141     }
142     delete[] ex;
143     cout << "Longueur finale: " << n << endl;
144     Terminate();
145     return 0;
146 }

```