

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<time.h>
4  #define NB_POINTS 100000
5  FILE * newFile(char * fileName){
6      FILE * f;
7      if( (f = fopen(fileName, "w")) == NULL) {
8          fprintf(stderr, "Impossible de creer le fichier %s\n", fileName);
9          exit(1);
10     }
11     return f;
12 }
13 int main(void){
14     int i, in;
15     float x, y;
16     FILE * f;
17     srand(time(NULL));
18     f = newFile("out.data");
19     for(i = 0, in = 0; i < NB_POINTS; i++) {
20         x = rand()/((float)RAND_MAX);
21         y = rand()/((float)RAND_MAX);
22         if(x * x + y * y <= 1.) {
23             fprintf(f, "%f\t%f\n", x, y);
24             in++;
25         }
26     }
27     printf("pi = %f\n", (4. * in) / NB_POINTS);
28     return 0;
29 }

```

```

1  set title "pi"
2  set data style dots
3  set term postscript landscape color
4  set output "pi.ps"
5  plot "out.data" using 1:2

```

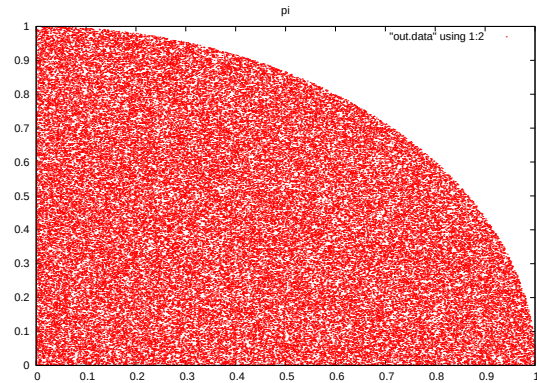
FIG. 1 – Calculer PI (le code C `pi.c`, le script pour GnuPlot `pi.dem`).

FIG. 2 – Le fichier postscript obtenu avec GnuPlot.

```

1  /* letemps.c --
2   * Fares Belhadj -
3   * amsi@ai.univ-paris8.fr
4   * 03/11/2004
5   */
6  #include <stdlib.h>
7  #include <stdio.h>
8  #include <sys/time.h>
9  #include "letemps.h"
10
11  static struct timeval ti;
12
13  extern void initTemps(void) {
14      gettimeofday(&ti, (struct timezone*) 0);
15  }
16
17  double getTemps(void) {
18      struct timeval t;
19      double diff;
20      gettimeofday(&t, (struct timezone*) 0);
21      diff = (t.tv_sec - ti.tv_sec) * 1000000
22            + (t.tv_usec - ti.tv_usec);
23      return diff/1000000.;
24  }

```

```

1  /* letemps.h --
2   * Fares Belhadj -
3   * amsi@ai.univ-paris8.fr
4   * 03/11/2004
5   */
6  #ifndef LETEMPS_H
7  #define LETEMPS_H
8  void initTemps(void);
9  extern double getTemps(void);
10 #endif /* LETEMPS_H */

```

FIG. 3 – Le temps (letemps.c et letemps.h).

Devoir 04 :

Modifier les fichiers `pi.c` et `pi.dem` pour obtenir un Postscript contenant tous les points aléatoires ; les points dans le cercle seront colorés autrement que ceux qui sont à l'extérieur du cercle. Envoyer, par email, le fichier Postscript obtenu.

Écrire un algorithme de tri. Tester-le sur des tableaux de *taille* = {10000, 20000, ..., 1000000}. À chaque fois, envoyer le temps de calcul (de tri) dans un fichier de données. Créer avec GnuPlot l'hitogramme des temps de calcul en prenant une échelle logarithmique pour les ordonnées (l'axe *y*). Envoyer le résultat par email.