6CS005 Learning Journal - Semester 1 2019/20

Bishrut Neupane 1928726

Table of Contents

1 MPI

1.1 Password Cracking

```
1.  #include <stdio.h>
2.  #include <string.h>
3.  #include <stdlib.h>
4.  #include <crypt.h>
5.  #include <time.h>
6.  #include <math.h>
7.  #include <mpi.h>
8.  /**************************************************************************
9.  *******
10.   Demonstrates how to crack an encrypted password using a simple
11.   "brute force" algorithm. Works on passwords that consist only of 2
12. uppercase
```

```
  letters and a 2 digit integer. Your personalised data set is included
in the
  code.


********************************************************************
******/
int n_passwords = 4;

char *encrypted_passwords[] = {

"$6$KB$6SsUGf4Cq7/Oooym9WWQN3VKeo2lynKV9gXVyEG4HvYy1UFRx.XAye89TLp/OTcW7cGpf9UlU0F.cK/S9CfZn1",

"$6$KB$1ocIiuN6StvEskjsYoYBid/gy8zXybieNCm9uM94nRw.ik9I04W3DJg0E52dswnozLmM0BIlzRZxgd.TleBwp1",

"$6$KB$L4mWcpv6rMAbZdxfSsuAL2UZhbJ4vSGAAxk.vEcRKvIuPpwcSRKHzi3BXzWQWaH1p1ubwaFl.06CRQv6bVo3M1",

"$6$KB$OpOIZac00sMBfYemANRTO3lKNZCFLegKAMakI3i2fk78/vZgo01X5mdG/1R1K0Ohs0V1AuxfOK7KY.th3dInb0"
};

/**
 Required by lack of standard function in C.
*/

void substr(char *dest, char *src, int start, int length){
  memcpy(dest, src + start, length);
  *(dest + length) = '\0';
}

/**
 This function can crack the kind of password explained above. All
combinations
 that are tried are displayed and when the password is found, #, is put
at the
 start of the line. Note that one of the most time consuming operations
that
 it performs is the output of intermediate results, so performance
experiments
 for this kind of program should not include this. i.e. comment out the
printfs.
*/

void crackAM(char *salt_and_encrypted){
  int x, y, z;     // Loop counters
  char salt[7];    // String used in hashing the password. Need space

  char plain[7];   // The combination of letters currently being checked
```

```
60.   char *enc;       // Pointer to the encrypted password
61.   int count = 0;   // The number of combinations explored so far
62.
63.   substr(salt, salt_and_encrypted, 0, 6);
64.
65.   for(x='A'; x<='M'; x++){
66.     for(y='A'; y<='Z'; y++){
67.       for(z=0; z<=99; z++){
68.         sprintf(plain, "%c%c%02d", x, y, z);
69.         enc = (char *) crypt(plain, salt);
70.         count++;
71.         if(strcmp(salt_and_encrypted, enc) == 0){
72.           printf("#%-8d%s %s\n", count, plain, enc);
73.         } /*else {
74.           printf(" %-8d%s %s\n", count, plain, enc);
75.         }*/
76.       }
77.     }
78.   }
79.   printf("%d solutions explored\n", count);
80. }
81.
82. void crackNZ(char *salt_and_encrypted){
83.   int x, y, z;     // Loop counters
84.   char salt[7];    // String used in hashing the password. Need space
85.
86.   char plain[7];   // The combination of letters currently being checked
87.   char *enc;       // Pointer to the encrypted password
88.   int count = 0;   // The number of combinations explored so far
89.
90.   substr(salt, salt_and_encrypted, 0, 6);
91.
92.   for(x='N'; x<='Z'; x++){
93.     for(y='A'; y<='Z'; y++){
94.       for(z=0; z<=99; z++){
95.         sprintf(plain, "%c%c%02d", x, y, z);
96.         enc = (char *) crypt(plain, salt);
97.         count++;
98.         if(strcmp(salt_and_encrypted, enc) == 0){
99.           printf("#%-8d%s %s\n", count, plain, enc);
100.               } /*else {
101.                 printf(" %-8d%s %s\n", count, plain, enc);
102.               }*/
103.            }
104.          }
105.        }
106.        printf("%d solutions explored\n", count);
```

```
107.      }
108.      int time_difference(struct timespec *start,
109.                          struct timespec *finish,
110.                          long long int *difference) {
111.        long long int ds =  finish->tv_sec - start->tv_sec;
112.        long long int dn =  finish->tv_nsec - start->tv_nsec;
113.
114.        if(dn < 0 ) {
115.          ds--;
116.          dn += 1000000000;
117.        }
118.        *difference = ds * 1000000000 + dn;
119.        return !(*difference > 0);
120.      }
121.      int main(int argc, char *argv[]){
122.
123.       struct timespec start, finish;
124.        long long int time_elapsed;
125.        clock_gettime(CLOCK_MONOTONIC, &start);
126.
127.
128.
129.      int size, rank;
130.
131.        MPI_Init(NULL, NULL);
132.        MPI_Comm_size(MPI_COMM_WORLD, &size);
133.        MPI_Comm_rank(MPI_COMM_WORLD, &rank);
134.        if(size != 3) {
135.          if(rank == 0) {
136.            printf("This program needs to run on exactly 3 processes\n");
137.          }
138.        } else {
139.          if(rank ==0){
140.            int x;
141.            MPI_Send(&x, 1, MPI_INT, 1, 0, MPI_COMM_WORLD);
142.            MPI_Send(&x, 1, MPI_INT, 2, 0, MPI_COMM_WORLD);
143.
144.
145.          }
146.      else if (rank == 1){
147.              int number, i;
148.              MPI_Recv(&number, 1, MPI_INT, 0, 0, MPI_COMM_WORLD,MPI_STATUS_IGNORE);
149.          for ( i = 0; i<n_passwords;i<i++){
150.              crackAM(encrypted_passwords[i]);
151.
152.              }
153.            }
```

```
154.           else {
155.               int number, i;
156.               MPI_Recv(&number, 1, MPI_INT, 0, 0, MPI_COMM_WORLD,MPI_STATUS_IGNORE);
157.           for (i = 0; i<n_passwords;i<i++){
158.               crackNZ(encrypted_passwords[i]);
159.           }
160.              }
161.         }
162.         MPI_Finalize();
163.        clock_gettime(CLOCK_MONOTONIC, &finish);
164.             time_difference(&start, &finish, &time_elapsed);
165.             printf("Time elapsed was %lldns or %0.9lfs\n", time_elapsed,
166.                                                 (time_elapsed/1.0e9));
167.         return 0;
168.       }
```

Insert a table that shows running times for the original and MPI versions.

SN	Original	MPI
1	508.144274294	256.459796178
2	504.983681529	256.450385224
3	508.551703587	256.450721487
4	506.553751119	257.213704845
5	507.681018088	257.213911363
6	505.525211300	257.214199365
7	505.730813811	258.198209344
8	506.607479040	258.198138104
9	505.055320878	258.198150646
10	507.994538268	255.766154208

Write a short analysis of the results

1.2 Image Processing

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <GL/glut.h>
#include <GL/gl.h>
#include <malloc.h>
#include <signal.h>
#include <mpi.h>
/*****************************************************************************
  Displays two grey scale images. On the left is an image that has come from an
  image processing pipeline, just after colour thresholding. On the right is
  the result of applying an edge detection convolution operator to the left
  image. This program performs that convolution.

  Things to note:
    - A single unsigned char stores a pixel intensity value. 0 is black, 256 is
    white.
    - The colour mode used is GL_LUMINANCE. This uses a single number to
    represent a pixel's intensity. In this case we want 256 shades of grey,
    which is best stored in eight bits, so GL_UNSIGNED_BYTE is specified as
    the pixel data type.


  To compile adapt the code below wo match your filenames:
    mpicc ip_coursework_069.c -o ip_coursework_069 -lglut -lGL -lm -lrt

    mpiexec -n 5 -quiet ./ip_coursework_069

  Dr Kevan Buckley, University of Wolverhampton, 2018
*****************************************************************************/
```

```
    #define width 100
    #define height 72

    unsigned char image[], results[width * height];
    //int startIndex, endIndex;

int time_difference(struct timespec *start, struct timespec *finish,
                              long long int *difference) {
  long long int ds =  finish->tv_sec - start->tv_sec;
  long long int dn =  finish->tv_nsec - start->tv_nsec;

  if(dn < 0 ) {
    ds--;
    dn += 1000000000;
  }
  *difference = ds * 1000000000 + dn;
  return !(*difference > 0);
}

 void detect_edges(unsigned char *in, unsigned char *out) {
 int i;
 int n_pixels = (width * height);

  for(i=0;i<n_pixels;i++) {
    int x, y; // the pixel of interest
    int b, d, f, h; // the pixels adjacent to x,y used for the calculation
    int r; // the result of calculate

    y = i / width;
    x = i - (width * y);

    if (x == 0 || y == 0 || x == width - 1 || y == height - 1) {
    results[i] = 0;
    } else {
    b = i + width;
    d = i - 1;
    f = i + 1;
    h = i - width;

    r = (in[i] * 4) + (in[b] * -1) + (in[d] * -1) + (in[f] * -1)
        + (in[h] * -1);

    if (r > 0) { // if the result is positive this is an edge pixel
        out[i] = 255;
    } else {
        out[i] = 0;
    }
```

```
78.     }
79.     }
80.     }
81.
82.     void tidy_and_exit() {
83.     exit(0);
84.     }
85.
86.     void sigint_callback(int signal_number){
87.     printf("\nInterrupt from keyboard\n");
88.     tidy_and_exit();
89.     }
90.
91.     static void display() {
92.     glClear(GL_COLOR_BUFFER_BIT);
93.   glRasterPos4i(-1, -1, 0, 1);
94.   glDrawPixels(width, height, GL_LUMINANCE, GL_UNSIGNED_BYTE, image);
95.   glRasterPos4i(0, -1, 0, 1);
96.   glDrawPixels(width, height, GL_LUMINANCE, GL_UNSIGNED_BYTE, results);
97.   glFlush();
98. }
99.
100.       static void key_pressed(unsigned char key, int x, int y) {
101.         switch(key){
102.           case 27: // escape
103.           tidy_and_exit();
104.           break;
105.           default:
106.           printf("\nPress escape to exit\n");
107.           break;
108.         }
109.       }
110.
111.       int main(int argc, char **argv) {
112.         signal(SIGINT, sigint_callback);
113.         // printf("image dimensions %dx%d\n", width, height);
114.         // struct timespec start, finish;
115.         // long long int difference;
116.         int account = 0;
117.
118.         int size, rank;
119.         // clock_gettime(CLOCK_MONOTONIC, &start);
120.         MPI_Init(NULL, NULL);
121.         MPI_Comm_size(MPI_COMM_WORLD, &size);
122.         MPI_Comm_rank(MPI_COMM_WORLD, &rank);
123.         if(size != 5) {
124.           if(rank == 0) {
```

```
125.           printf("This program needs 5 processes\n");
126.           }
127.         } else {
128.           if(rank ==0){
129.               struct timespec start, finish;
130.           long long int time_elapsed;
131.           clock_gettime(CLOCK_MONOTONIC, &start);
132.               MPI_Send(&results[0], 1800, MPI_UNSIGNED_CHAR, 1, 0, MPI_COMM_WORLD);
133.               MPI_Send(&results[1800], 1800, MPI_UNSIGNED_CHAR, 2, 0, MPI_COMM_WORLD);
134.               MPI_Send(&results[3600], 1800, MPI_UNSIGNED_CHAR, 3, 0, MPI_COMM_WORLD);
135.               MPI_Send(&results[5400], 1800, MPI_UNSIGNED_CHAR, 4, 0, MPI_COMM_WORLD);
136.
137.               MPI_Recv(&results[0], 1800, MPI_UNSIGNED_CHAR, 1, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
138.                   MPI_Recv(&results[1800], 1800,MPI_UNSIGNED_CHAR, 2, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
139.               MPI_Recv(&results[3600], 1800,MPI_UNSIGNED_CHAR, 3, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
140.               MPI_Recv(&results[5400], 1800,MPI_UNSIGNED_CHAR, 4, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
141.
142.                   clock_gettime(CLOCK_MONOTONIC, &finish);
143.         time_difference(&start, &finish, &time_elapsed);
144.         printf("Time elapsed was %lldns or %0.9lfs\n", time_elapsed,
145.                (time_elapsed/1.0e9));
146.                   glutInit(&argc, argv);
147.                   glutInitWindowSize(width * 2,height);
148.                   glutInitDisplayMode(GLUT_SINGLE | GLUT_LUMINANCE);
149.
150.                   glutCreateWindow("6CS005 Image Progessing Courework");
151.                   glutDisplayFunc(display);
152.                   glutKeyboardFunc(key_pressed);
153.                   glClearColor(0.0, 1.0, 0.0, 1.0);
154.
155.                   glutMainLoop();
156.
157.                   tidy_and_exit();
158.
159.
160.           } else {
161.           if(rank == 1){
162.
163.               //startIndex = 0;
164.               //endIndex = 1799;
165.               MPI_Recv(&results[0], 1800, MPI_UNSIGNED_CHAR, 0, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
166.               detect_edges(image, results);
167.                   MPI_Send(&results[0], 1800, MPI_UNSIGNED_CHAR, 0, 0, MPI_COMM_WORLD);
168.           }
169.           else if(rank == 2){
170.               //startIndex = 1800;
171.               //endIndex = 3599;
```

```
172.
173.                 MPI_Recv(&results[1800], 1800, MPI_UNSIGNED_CHAR, 0, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
174.             detect_edges(image, results);
175.                 MPI_Send(&results[1800], 1800, MPI_UNSIGNED_CHAR, 0, 0, MPI_COMM_WORLD);
176.           }
177.           else if(rank == 3){
178.           // startIndex = 3600;
179.            //endIndex = 5399;
180.
181.            MPI_Recv(&results[3600], 1800, MPI_UNSIGNED_CHAR, 0, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
182.            detect_edges(image, results);
183.                    MPI_Send(&results[3600], 1800, MPI_UNSIGNED_CHAR, 0, 0, MPI_COMM_WORLD);
184.
185.           }
186.           else if(rank == 4){
187.               // startIndex = 5400;
188.               // endIndex = 7199;
189.
190.                MPI_Recv(&results[5400], 1800, MPI_UNSIGNED_CHAR, 0, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
191.            detect_edges(image, results);
192.                    MPI_Send(&results[5400], 1800, MPI_UNSIGNED_CHAR, 0, 0, MPI_COMM_WORLD);
193.           }
194.           }
195.         }
196.         MPI_Finalize();
197.         // clock_gettime(CLOCK_MONOTONIC, &finish);
198.         // time_difference(&start, &finish, &difference);
199.         // printf("run lasted %9.5lfs\n", difference/1000000000.0);
200.         return 0;
201.       }
202.
203.       unsigned char image[] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
204.         0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
205.         0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
206.         0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
207.         0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
208.         0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
209.         0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
210.         0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
211.         0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
212.         0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
213.         0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
214.         0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
215.         0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
216.         0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
217.         0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
218.         0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
```

```
219.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
220.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
221.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
222.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
223.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
224.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
225.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
226.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
227.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
228.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
229.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
230.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
231.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
232.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
233.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
234.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
235.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
236.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
237.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
238.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
239.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
240.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
241.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
242.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
243.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
244.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
245.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
246.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
247.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
248.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
249.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
250.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
251.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
252.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
253.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
254.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
255.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
256.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
257.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
258.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
259.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
260.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
261.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
262.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
263.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
264.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
265.        0,0,0,0,0,0,0,0,0,0,0,255,255,255,255,255,255,255,255,
```

```
255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,255,0,0,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,0,0,0,0,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,0,255,255,255,0,0,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,0,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,0,255,255,255,255,255,
255,255,255,0,255,255,0,255,255,255,255,0,255,255,0,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,0,255,0,255,255,0,255,0,
255,255,0,255,0,255,255,255,255,0,255,255,0,0,255,255,255,255,0,
0,0,255,0,255,255,255,255,255,255,255,255,255,255,0,255,255,0,0,
255,0,255,255,0,255,255,255,0,255,255,0,0,255,0,255,0,255,0,
0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,0,255,255,255,255,0,0,
255,255,0,255,0,0,255,0,255,0,0,0,0,0,0,255,255,0,0,
255,255,0,0,0,255,0,255,0,255,255,255,255,255,255,255,255,255,255,
255,255,255,0,0,255,0,255,255,0,255,255,0,255,255,255,0,255,255,
255,255,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,255,
255,255,0,255,255,0,255,0,0,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,0,255,255,255,0,255,255,255,255,255,
255,255,255,255,255,0,0,255,255,255,255,255,255,255,0,0,255,255,255,
```

```
313.        255,255,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
314.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
315.        255,255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,255,255,255,
316.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
317.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
318.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
319.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
320.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
321.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
322.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
323.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
324.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
325.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
326.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
327.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
328.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
329.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
330.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
331.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
332.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,255,0,0,255,
333.        0,255,255,0,255,0,0,255,255,0,0,255,255,255,0,255,0,0,255,
334.        0,0,0,255,0,255,255,255,0,255,255,255,255,255,255,255,0,255,255,
335.        255,0,255,255,255,255,255,0,255,255,255,255,0,255,0,255,0,0,255,
336.        255,0,0,0,0,0,0,0,255,0,0,0,255,0,255,0,255,255,255,
337.        255,255,255,255,255,0,255,0,255,255,0,0,255,255,255,255,255,0,255,
338.        0,0,0,255,0,0,0,255,0,0,0,255,255,0,255,255,255,255,0,
339.        0,255,0,0,0,255,0,0,255,0,0,255,0,0,255,255,255,255,0,
340.        0,255,0,0,0,255,0,255,0,0,0,255,0,0,255,255,0,0,0,
341.        0,255,0,255,255,255,255,0,0,255,255,0,0,255,0,0,255,0,0,
342.        255,0,255,255,255,255,255,255,255,255,255,0,0,255,255,0,255,255,255,
343.        255,255,255,0,0,255,255,255,255,255,255,0,255,255,0,255,255,0,255,
344.        255,255,255,255,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,
345.        255,255,255,255,255,0,255,0,0,255,0,0,0,255,255,255,0,255,0,
346.        255,255,255,0,255,0,255,255,255,255,0,255,255,255,255,255,255,255,255,
347.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,255,255,
348.        255,255,0,0,255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,
349.        0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
350.        255,255,255,255,255,255,255,255,255,255,255,0,255,255,255,255,255,255,255,255,
351.        0,255,255,255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,255,
352.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
353.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
354.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
355.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
356.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
357.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
358.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
359.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
```

```
360.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
361.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
362.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
363.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
364.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
365.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
366.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
367.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
368.        255,255,255,255,255,255,255,255,255,0,255,0,0,0,0,0,0,0,255,255,
369.        255,0,0,0,0,0,0,0,0,0,0,255,255,255,255,255,255,255,255,255,0,
370.        0,0,0,0,0,0,0,0,255,255,255,255,255,255,255,0,0,0,255,255,0,
371.        0,0,0,255,255,255,255,255,0,255,255,0,0,0,0,255,255,255,0,
372.        0,0,0,255,255,255,255,255,0,0,0,0,0,0,0,255,255,255,255,
373.        255,255,255,0,0,0,0,255,255,255,255,255,255,255,0,0,0,0,0,0,
374.        0,0,0,255,255,255,0,0,0,0,0,0,0,0,0,255,255,255,255,
375.        255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,255,255,255,255,0,
376.        0,0,255,255,255,0,0,0,0,255,255,255,0,0,255,255,0,0,0,
377.        0,255,255,255,0,0,0,0,255,255,255,255,0,0,0,0,0,0,0,
378.        0,0,255,255,255,255,255,255,0,0,0,0,255,255,255,255,255,255,0,
379.        0,0,0,0,0,0,0,0,255,255,255,0,0,0,0,0,0,0,0,
380.        0,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,
381.        255,255,255,255,0,0,0,255,255,255,0,0,0,0,0,0,0,0,0,
382.        255,255,0,0,0,0,255,255,255,0,0,0,0,255,255,255,0,0,0,
383.        0,0,0,0,0,0,0,0,255,255,255,255,255,0,0,0,0,255,255,
384.        255,255,255,255,0,0,0,0,0,0,0,0,0,255,255,255,0,0,0,
385.        0,0,0,0,0,0,255,255,255,255,255,255,255,255,255,0,0,0,0,0,
386.        255,0,0,0,0,0,255,255,255,0,0,0,255,255,255,0,0,0,0,
387.        0,0,0,0,0,255,255,0,0,0,0,255,255,255,0,0,0,0,255,
388.        255,255,0,0,0,0,255,255,255,0,0,0,0,255,255,255,255,255,0,
389.        0,0,0,255,255,255,255,255,255,0,0,0,0,255,255,255,255,255,255,
390.        255,255,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,
391.        0,0,0,0,255,255,255,0,0,0,0,255,255,255,0,0,0,255,255,
392.        255,255,0,0,0,0,0,0,0,0,255,255,0,0,0,0,255,255,255,
393.        0,0,0,0,255,255,255,0,0,0,0,255,255,255,0,0,0,0,255,
394.        255,255,255,255,0,0,0,0,255,255,255,255,255,255,0,0,0,0,255,
395.        255,255,255,255,255,255,255,0,0,0,0,255,255,255,255,255,255,255,255,
396.        255,255,255,255,255,0,0,0,0,255,255,255,0,0,0,0,255,255,255,
397.        0,0,0,255,255,255,255,0,0,0,0,255,0,0,0,255,255,0,0,
398.        0,0,0,0,0,0,0,0,0,255,255,255,0,0,0,255,255,255,255,
399.        255,0,0,0,0,255,255,255,255,0,0,0,0,255,255,255,255,255,255,
400.        0,0,0,0,0,0,0,0,0,255,255,255,0,0,0,0,255,255,255,
401.        255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,255,255,255,0,0,
402.        0,0,255,255,255,0,0,0,255,255,255,255,0,0,0,0,255,0,0,
403.        0,255,255,0,0,0,0,0,0,0,0,0,0,255,255,255,0,0,
404.        0,255,255,255,255,255,0,0,0,0,255,255,255,255,0,0,0,0,255,
405.        255,255,255,255,255,0,0,0,0,0,0,0,0,0,255,255,255,0,0,
406.        0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,
```

```
255,255,255,0,0,0,0,255,255,255,0,0,0,255,255,255,255,255,0,
0,0,0,0,0,0,255,255,0,0,0,0,0,0,0,0,0,0,0,
255,255,255,0,0,0,0,255,255,255,255,0,0,0,0,255,255,255,255,
0,0,0,0,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,
255,255,255,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,
255,0,0,0,0,255,255,255,0,0,0,0,255,255,255,0,0,0,255,
255,255,255,255,0,0,0,0,0,0,0,255,255,0,0,0,0,0,0,
0,0,0,0,0,0,255,255,0,0,0,0,255,255,255,0,0,0,0,
255,255,255,255,255,0,0,0,0,255,255,255,255,255,255,0,0,0,0,
255,255,255,255,255,255,255,255,0,0,0,0,255,255,255,255,255,255,255,
255,255,255,255,255,255,0,0,0,0,255,255,0,0,0,0,0,255,255,
255,0,0,0,255,255,255,255,255,0,0,0,0,0,0,0,255,255,0,
0,0,0,255,255,255,255,0,0,0,0,255,255,0,0,0,0,0,255,
0,0,0,0,0,255,255,0,0,0,0,0,0,0,0,0,0,0,255,
255,0,0,0,0,0,0,0,0,0,255,255,255,0,0,0,0,255,255,
255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,
0,0,255,255,255,255,0,0,0,255,255,255,255,255,255,0,0,0,0,
0,255,255,255,0,0,0,0,255,255,255,255,0,0,0,0,255,255,255,
0,0,0,0,0,0,0,0,0,0,255,255,0,0,0,0,0,0,0,
0,0,0,0,255,255,0,0,0,0,0,0,0,0,0,255,255,255,0,
0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,
0,0,0,0,0,0,0,255,255,255,255,0,0,0,255,255,255,255,255,
255,0,0,0,0,0,255,255,255,0,0,0,0,255,255,255,255,0,0,
0,0,255,255,255,255,0,0,0,0,0,0,0,0,255,255,255,0,0,
0,0,0,0,0,0,0,0,0,255,255,0,0,0,0,0,0,0,0,
0,255,255,255,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,
255,255,0,0,0,0,0,0,0,0,255,255,255,255,255,255,0,0,0,
255,255,255,255,255,255,0,0,0,0,0,255,255,255,255,0,0,0,255,
255,255,255,0,0,0,255,255,255,255,255,255,0,0,0,0,0,255,255,
255,255,255,0,0,0,0,0,0,0,0,0,0,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
```

```
454.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
455.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
456.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
457.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
458.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
459.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
460.        255,255,255,255,255,255,255,255,255,255,255,0,0,255,255,0,0,255,255,0,
461.        255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,255,255,255,0,
462.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
463.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
464.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
465.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
466.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
467.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
468.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
469.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
470.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
471.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
472.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
473.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
474.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
475.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
476.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
477.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
478.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
479.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
480.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
481.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
482.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
483.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
484.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
485.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
486.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
487.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
488.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
489.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
490.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
491.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
492.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
493.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
494.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
495.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
496.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
497.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
498.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
499.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
500.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
```

```
501.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
502.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
503.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
504.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
505.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
506.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
507.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
508.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
509.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
510.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
511.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
512.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
513.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
514.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
515.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
516.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
517.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
518.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
519.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
520.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
521.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
522.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
523.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
524.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
525.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
526.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
527.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
528.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
529.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
530.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
531.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
532.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
533.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
534.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
535.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
536.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
537.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
538.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
539.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
540.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
541.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
542.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
543.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
544.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
545.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
546.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
547.        0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
```

```
548.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
549.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
550.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
551.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
552.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
553.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
554.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
555.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
556.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
557.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
558.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
559.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
560.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
561.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
562.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
563.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
564.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
565.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
566.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
567.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
568.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
569.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
570.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
571.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
572.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
573.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
574.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
575.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
576.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
577.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
578.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
579.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
580.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
581.          0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
582.       };
```

Insert a table that shows running times for the original and MPI versions.

SN	Original	MPI
1	0.000128093	0.000563566
2	0.000194618	0.000372350
3	0.000387604	0.000581927
4	0.000407624	0.000361783
5	0.00096145	0.000365554
6	0.000759508	0.000551932
7	0.000248477	0.000478734
8	0.000453902	0.000741825
9	0.000178767	0.000473569
10	0.000301390	0.000473569

Write a short analysis of the results

1.3 Linear Regression

```
1.  #include <stdio.h>
2.  #include <math.h>
3.  #include <mpi.h>
4.  #include <time.h>
5.
6.  /****************************************************************************
7.  * This program takes an initial estimate of m and c and finds the associated
8.  * rms error. It is then as a base to generate and evaluate 8 new estimates,
9.  * which are steps in different directions in m-c space. The best estimate is
10. * then used as the base for another iteration of "generate and evaluate". This
11. * continues until none of the new estimates are better than the base. This is
12. * a gradient search for a minimum in mc-space.
13. *
14. * To compile:
15. *   mpicc 69.c -o 69 -lm
16. *
17. * To run:
18. *   mpirun -n 9 ./69
19. *
20. * Dr Kevan Buckley, University of Wolverhampton, 2018
21. ****************************************************************************/
22.
```

```
typedef struct point_t
{
   double x;
   double y;
} point_t;

int n_data = 1000;
point_t data[];

double residual_error (double x, double y, double m, double c)
{
   double e = (m * x) + c - y;
   return e * e;
}

double rms_error (double m, double c)
{
   int i;
   double mean;
   double error_sum = 0;

   for (i = 0; i < n_data; i++)
   {
      error_sum += residual_error (data[i].x, data[i].y, m, c);
   }

   mean = error_sum / n_data;

   return sqrt (mean);
}
int time_difference(struct timespec *start, struct timespec *finish,
                    long long int *difference) {
                       long long int ds =  finish->tv_sec - start->tv_sec;
                       long long int dn =  finish->tv_nsec - start->tv_nsec;

                       if(dn < 0 ) {
                          ds--;
                          dn += 1000000000;
                       }
                       *difference = ds * 1000000000 + dn;
                       return !(*difference > 0);
}
int main () {


   struct timespec start, finish;
   long long int time_elapsed;
```

```
70.    clock_gettime(CLOCK_MONOTONIC, &start);
71.
72.
73.    int rank, size;
74.    int i;
75.    double bm = 1.3;
76.    double bc = 10;
77.    double be;
78.    double dm[8];
79.    double dc[8];
80.    double e[8];
81.    double step = 0.01;
82.    double best_error = 999999999;
83.    int best_error_i;
84.    int minimum_found = 0;
85.    double pError = 0;
86.    double baseMC[2];
87.
88.    double om[] = { 0, 1, 1, 1, 0, -1, -1, -1 };
89.    double oc[] = { 1, 1, 0, -1, -1, -1, 0, 1 };
90.
91.
92.    MPI_Init (NULL, NULL);
93.    MPI_Comm_size (MPI_COMM_WORLD, &size);
94.    MPI_Comm_rank (MPI_COMM_WORLD, &rank);
95.
96.    be = rms_error (bm, bc);
97.
98.    if (size != 9)
99.    {
100.            if (rank == 0)
101.            {
102.               printf
103.                  ("This program is made for run with only 9 processes.\n");
104.               return 0;
105.            }
106.         }
107.
108.         while (!minimum_found)
109.         {
110.
111.            if (rank != 0)
112.            {
113.               i = rank - 1;
114.               dm[i] = bm + (om[i] * step);
115.               dc[i] = bc + (oc[i] * step);
116.               pError = rms_error (dm[i], dc[i]);
```

```
117.
118.                MPI_Send (&pError, 1, MPI_DOUBLE, 0, 0, MPI_COMM_WORLD);
119.                MPI_Send (&dm[i], 1, MPI_DOUBLE, 0, 0, MPI_COMM_WORLD);
120.                MPI_Send (&dc[i], 1, MPI_DOUBLE, 0, 0, MPI_COMM_WORLD);
121.
122.
123.                MPI_Recv (&bm, 1, MPI_DOUBLE, 0, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
124.                MPI_Recv (&bc, 1, MPI_DOUBLE, 0, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
125.                MPI_Recv (&minimum_found, 1, MPI_INT, 0, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
126.             }
127.             else
128.             {
129.                for (i = 1; i < size; i++)
130.                {
131.                   MPI_Recv (&pError, 1, MPI_DOUBLE, i, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
132.                   MPI_Recv (&dm[i-1], 1, MPI_DOUBLE, i, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
133.                   MPI_Recv (&dc[i-1], 1, MPI_DOUBLE, i, 0, MPI_COMM_WORLD, MPI_STATUS_IGNORE);
134.                   if (pError < best_error)
135.                   {
136.                      best_error = pError;
137.                      best_error_i = i - 1;
138.
139.                   }
140.                }
141.
142.                if (best_error < be)
143.                {
144.                   be = best_error;
145.                   bm = dm[best_error_i];
146.                   bc = dc[best_error_i];
147.                }
148.                else
149.                {
150.                   minimum_found = 1;
151.                }
152.
153.                for (i = 1; i < size; i++)
154.                {
155.                   MPI_Send (&bm, 1, MPI_DOUBLE, i, 0, MPI_COMM_WORLD);
156.                   MPI_Send (&bc, 1, MPI_DOUBLE, i, 0, MPI_COMM_WORLD);
157.                   MPI_Send (&minimum_found, 1, MPI_INT, i, 0, MPI_COMM_WORLD);
158.                }
159.             }
160.          }
161.
162.
163.          if(rank==0) {
```

```
164.             printf ("minimum m,c is %lf,%lf with error %lf\n", bm, bc, be);
165.             clock_gettime(CLOCK_MONOTONIC, &finish);
166.             time_difference(&start, &finish, &time_elapsed);
167.             printf("Time elapsed : %lldns or %0.9lfs\n", time_elapsed,
168.                (time_elapsed/1.0e9));
169.          }
170.
171.          MPI_Finalize();
172.          return 0;
173.       }
174.
175.       point_t data[] = {
176.         {82.73,128.67},{79.53,133.54},{66.86,124.65},{69.21,135.74},
177.         {82.20,122.07},{84.32,120.46},{71.12,93.14},{85.64,121.42},
178.         {69.22,116.28},{83.12,137.30},{84.31,113.18},{75.60,121.42},
179.         {69.04,91.83},{85.41,131.06},{17.44,58.69},{68.92,119.86},
180.         {69.95,110.05},{ 0.15, 5.39},{73.96,118.70},{27.70,64.64},
181.         {97.97,158.15},{56.21,100.99},{30.27,48.32},{37.47,89.65},
182.         {98.98,144.03},{92.61,133.89},{ 4.72,32.88},{19.51,57.43},
183.         {94.74,145.50},{31.66,71.27},{94.76,134.53},{32.73,59.95},
184.         {32.64,54.53},{38.78,69.06},{91.47,150.49},{77.99,119.35},
185.         {33.38,65.87},{79.28,123.62},{39.69,72.53},{95.47,140.97},
186.         {82.64,137.69},{25.53,51.33},{68.58,85.98},{92.25,132.34},
187.         {74.79,101.30},{ 1.32,18.87},{53.85,95.13},{78.75,128.26},
188.         { 2.91,21.77},{90.68,128.55},{11.44,35.27},{30.72,56.54},
189.         {49.06,74.08},{49.09,83.45},{62.54,104.58},{38.83,72.26},
190.         {78.43,130.83},{69.49,122.49},{27.27,56.35},{80.06,131.95},
191.         { 5.73,39.00},{80.21,140.42},{ 8.47,36.12},{86.98,152.43},
192.         {64.26,108.56},{95.74,133.36},{15.06,48.67},{31.96,72.31},
193.         {95.27,141.34},{61.10,89.26},{27.51,68.47},{26.48,60.30},
194.         {92.61,128.38},{ 8.25,47.51},{90.69,118.91},{45.40,79.96},
195.         {23.59,53.12},{46.71,68.27},{21.15,50.29},{27.99,76.29},
196.         { 7.75,43.57},{13.70,43.56},{74.85,97.83},{50.93,103.11},
197.         {33.80,64.85},{80.99,125.37},{92.41,126.27},{92.61,134.36},
198.         {34.70,55.32},{35.07,55.04},{86.87,157.26},{41.99,90.46},
199.         {16.27,44.43},{36.31,83.84},{22.35,73.11},{89.11,127.19},
200.         {56.11,77.28},{51.90,75.07},{35.74,94.18},{10.66,29.60},
201.         {61.27,114.15},{77.55,117.04},{61.17,99.68},{15.54,55.33},
202.         {91.99,143.18},{12.91,21.82},{48.52,89.94},{54.88,90.86},
203.         {73.59,131.33},{ 5.49,13.95},{92.31,147.29},{48.50,89.49},
204.         {40.02,58.26},{48.22,81.96},{17.08,52.59},{34.27,66.17},
205.         {59.06,94.26},{92.71,134.53},{37.70,65.30},{77.11,111.38},
206.         {43.27,74.12},{79.71,123.45},{ 0.86,38.69},{ 3.00,17.76},
207.         {56.03,80.33},{17.66,43.27},{18.39,47.08},{31.08,83.84},
208.         {32.64,77.85},{51.68,84.57},{78.46,134.18},{ 9.57,40.28},
209.         {68.38,98.26},{30.29,67.59},{86.15,131.86},{16.82,64.91},
210.         { 3.35,20.88},{65.78,98.73},{50.70,90.92},{38.26,71.11},
```

```
        {85.52,132.23},{44.06,83.02},{44.09,86.42},{81.86,114.30},
        {33.98,69.09},{93.80,147.73},{59.58,103.07},{98.75,154.73},
        {88.98,120.59},{78.08,109.00},{82.77,133.94},{76.49,106.31},
        {55.38,85.71},{46.56,79.57},{83.92,141.58},{81.38,133.52},
        { 4.88,35.01},{ 4.57,17.99},{57.96,90.07},{33.42,63.80},
        { 9.95,34.53},{47.14,92.75},{63.17,105.19},{95.01,163.93},
        {30.36,57.81},{ 2.46,23.97},{69.75,115.88},{64.85,111.01},
        {25.18,56.58},{69.84,104.78},{40.43,51.98},{75.61,107.05},
        {36.75,69.37},{50.08,100.02},{64.97,103.68},{41.72,86.64},
        { 1.70,47.26},{99.93,141.75},{24.57,64.51},{75.23,116.35},
        { 1.95,18.53},{78.84,102.70},{67.38,97.71},{55.35,82.37},
        {58.10,100.09},{53.10,96.07},{41.24,83.81},{68.86,111.98},
        {87.36,86.88},{54.06,98.42},{64.12,90.56},{11.77,49.66},
        {99.43,134.33},{55.24,99.18},{56.44,74.73},{39.47,62.99},
        { 8.94,48.15},{92.91,130.45},{87.68,138.76},{80.37,116.69},
        {56.72,108.65},{ 0.76,24.26},{26.98,75.13},{ 0.39,42.16},
        {81.99,138.50},{88.32,117.16},{51.01,87.42},{21.38,55.45},
        {72.66,122.82},{18.04,53.56},{11.22,49.73},{36.75,60.26},
        {64.81,90.19},{72.72,121.14},{24.03,74.08},{41.38,81.38},
        {62.79,98.75},{63.66,109.17},{91.12,143.91},{ 7.41,34.06},
        {94.05,131.99},{53.12,90.28},{68.31,114.79},{25.33,67.23},
        {42.34,86.91},{94.61,131.38},{43.78,73.28},{50.18,78.10},
        {81.64,135.88},{11.27,44.45},{41.03,76.34},{21.25,57.54},
        {29.23,57.27},{35.74,75.16},{ 0.91,14.33},{30.08,59.05},
        {23.99,56.25},{90.79,120.98},{99.22,152.22},{94.21,143.09},
        {19.35,30.03},{82.04,113.25},{79.22,113.69},{83.40,144.06},
        {55.82,80.85},{42.49,48.94},{17.60,55.62},{35.65,81.91},
        {82.50,135.41},{81.15,114.46},{53.47,78.67},{44.30,73.73},
        {32.88,80.28},{99.26,147.55},{76.32,110.24},{78.97,110.27},
        {18.08,47.48},{87.01,140.40},{56.25,83.61},{42.62,55.40},
        {15.95,16.25},{47.85,106.69},{ 6.61,35.83},{66.38,116.30},
        {94.97,122.56},{42.29,73.37},{31.48,67.15},{69.67,105.40},
        {30.41,65.31},{ 2.98,19.40},{ 8.12,48.34},{80.41,127.03},
        {63.68,112.61},{24.60,78.23},{77.61,123.49},{39.87,38.20},
        {77.80,109.59},{58.53,107.63},{23.97,62.36},{ 7.77,27.38},
        { 0.80,41.55},{ 6.45,32.91},{45.32,82.24},{35.56,59.56},
        {65.05,97.68},{62.21,96.14},{86.61,121.99},{87.91,125.40},
        {48.08,88.87},{ 2.41,40.02},{69.55,119.31},{22.07,61.86},
        {61.87,121.40},{82.50,119.46},{26.97,38.40},{31.53,86.30},
        { 1.81,38.57},{72.57,108.34},{88.88,139.23},{63.90,95.79},
        {93.29,135.35},{86.26,143.55},{63.62,94.76},{20.24,38.84},
        {16.23,48.64},{72.87,108.22},{16.26,51.25},{37.86,66.06},
        {57.53,81.37},{61.66,97.20},{49.48,84.98},{95.20,142.45},
        {12.10,45.25},{47.79,84.80},{17.29,48.98},{47.11,87.23},
        {85.74,119.95},{89.94,142.94},{97.68,155.27},{78.73,123.81},
        {51.65,85.91},{52.82,96.05},{50.95,93.50},{16.14,37.21},
        {16.73,41.57},{57.25,95.50},{78.47,136.77},{42.35,75.64},
```

```
        {93.24,135.04},{12.56,38.20},{21.40,62.92},{70.60,136.98},
        {44.04,83.57},{ 6.43,36.61},{12.01,50.32},{79.61,119.78},
        {43.05,69.07},{14.42,53.01},{51.68,83.82},{25.59,55.77},
        { 9.14,31.58},{37.24,80.94},{15.73,69.21},{71.54,123.11},
        { 1.26,25.72},{ 4.25,38.46},{21.42,39.99},{44.12,79.01},
        {31.12,64.63},{85.27,143.62},{43.25,79.30},{77.27,104.30},
        {47.34,83.76},{90.57,125.82},{17.35,36.40},{82.01,130.41},
        {81.58,124.10},{68.62,117.62},{47.48,79.29},{ 4.30,26.77},
        { 6.94,32.22},{11.71,55.76},{22.62,54.74},{58.43,89.61},
        {69.10,111.51},{56.77,101.10},{67.10,102.75},{93.20,144.51},
        {83.61,128.56},{71.97,116.09},{75.19,122.16},{48.03,79.67},
        {97.95,143.80},{92.27,123.08},{23.88,63.39},{79.15,115.57},
        {24.42,51.27},{12.58,34.65},{46.58,78.16},{ 1.29,37.96},
        {17.09,45.61},{12.45,40.77},{82.75,107.46},{52.15,75.34},
        {39.51,68.51},{31.71,64.23},{39.36,72.00},{12.16,37.99},
        {83.13,127.76},{42.25,73.17},{45.32,77.14},{20.52,36.60},
        { 7.99,11.50},{23.34,55.47},{25.87,54.36},{78.73,112.49},
        {55.60,94.90},{31.98,73.40},{85.93,137.12},{58.56,97.64},
        {88.16,120.43},{78.65,136.60},{25.93,43.32},{84.83,136.32},
        {68.09,102.12},{68.36,111.80},{39.80,69.69},{ 0.38,27.89},
        { 4.49,27.85},{32.53,66.32},{54.23,97.63},{19.98,67.32},
        {90.62,143.43},{18.31,67.91},{95.66,146.41},{95.41,149.68},
        {71.64,111.15},{23.02,44.96},{97.06,154.54},{41.58,75.95},
        {79.80,130.01},{74.55,119.44},{72.19,113.27},{70.01,106.48},
        {75.24,94.18},{19.82,60.09},{96.31,137.91},{ 2.21,27.44},
        {40.52,70.36},{ 2.40,29.12},{35.24,57.25},{26.38,71.34},
        {26.02,59.48},{34.73,66.07},{45.15,78.23},{ 9.35,32.58},
        {19.37,57.18},{ 9.51,31.70},{15.03,49.81},{85.08,140.35},
        { 3.23,13.46},{58.26,108.47},{ 4.84,31.78},{49.49,83.50},
        {35.55,70.67},{26.51,55.44},{20.12,53.39},{72.73,119.37},
        {31.04,72.96},{30.66,58.35},{ 2.96,33.18},{18.68,31.50},
        {91.41,138.24},{44.67,81.81},{81.57,135.26},{ 0.17,26.66},
        {49.03,100.11},{54.47,102.27},{61.78,113.45},{22.67,59.51},
        {89.80,143.05},{33.05,78.20},{67.76,108.19},{ 7.64,41.18},
        {36.91,87.28},{95.44,147.27},{52.76,94.34},{ 3.52,29.51},
        {87.39,118.48},{41.48,64.71},{ 1.44,14.21},{95.04,136.99},
        {71.77,115.75},{23.39,47.58},{62.66,115.03},{15.98,34.38},
        {29.06,62.77},{ 2.94,28.25},{71.50,119.18},{65.24,119.14},
        {30.65,82.39},{16.36,38.82},{ 0.98,48.82},{33.19,56.41},
        {27.49,64.34},{53.69,102.47},{28.15,52.58},{40.21,66.07},
        {50.56,86.39},{74.71,97.44},{24.72,46.29},{48.05,80.63},
        {34.99,52.13},{66.75,115.96},{17.62,49.17},{98.99,157.80},
        {37.96,72.18},{56.88,105.06},{48.27,97.04},{71.18,138.90},
        {46.35,82.02},{10.43,44.65},{24.14,42.85},{82.21,144.13},
        {96.85,148.15},{93.68,126.32},{33.02,61.55},{66.73,108.51},
        {83.89,136.35},{80.85,91.16},{79.21,128.88},{84.37,119.84},
        {38.41,71.48},{47.49,85.53},{ 1.54,24.44},{68.32,106.44},
```

```
305.         {22.82,54.16},{ 2.65,16.35},{19.91,53.53},{12.99,34.98},
306.         {30.87,57.17},{44.10,83.88},{15.84,31.99},{36.46,59.74},
307.         {26.25,79.73},{79.12,132.06},{86.26,132.45},{ 0.61,23.61},
308.         {33.94,59.37},{99.92,145.88},{26.20,53.99},{69.77,115.40},
309.         {69.07,107.00},{ 1.89,17.20},{38.25,81.40},{27.08,62.96},
310.         {23.09,53.98},{55.56,86.93},{ 6.68,50.41},{22.86,49.26},
311.         {17.25,50.25},{19.01,50.16},{35.07,85.09},{59.08,89.15},
312.         {87.02,128.83},{ 1.57,27.68},{97.76,148.25},{70.78,108.00},
313.         {38.01,65.83},{96.41,139.67},{ 2.86,22.44},{27.05,53.00},
314.         {90.99,134.97},{86.60,145.27},{54.66,99.42},{67.61,107.07},
315.         {85.16,137.50},{87.64,144.60},{14.69,36.65},{16.08,49.31},
316.         {14.45,44.07},{65.91,98.39},{50.74,90.72},{ 6.98,31.11},
317.         {52.76,83.96},{ 8.03,43.93},{17.58,52.58},{33.63,59.04},
318.         {87.65,137.34},{77.97,142.54},{30.56,69.47},{59.61,114.61},
319.         {14.05,53.07},{87.65,116.66},{33.19,75.96},{31.87,66.95},
320.         {25.89,57.59},{48.60,75.67},{80.25,109.89},{ 6.61,24.27},
321.         { 4.56,44.00},{40.17,62.33},{92.32,117.73},{75.07,112.71},
322.         {17.10,35.12},{39.06,66.60},{ 4.26,34.01},{52.95,102.49},
323.         {45.73,76.57},{ 4.72,29.94},{ 2.01,17.54},{39.08,88.44},
324.         {82.94,141.75},{44.51,90.97},{27.27,63.14},{60.16,95.38},
325.         {41.26,72.59},{66.50,104.49},{58.37,110.13},{62.11,96.01},
326.         {70.30,90.15},{18.47,47.61},{24.80,51.82},{79.02,133.40},
327.         {96.61,147.92},{18.14,33.27},{ 0.83,51.20},{99.67,143.65},
328.         {34.07,67.38},{57.28,110.02},{35.92,59.90},{66.15,124.45},
329.         {81.82,135.08},{ 2.97,28.49},{95.97,135.79},{51.17,80.95},
330.         {91.47,142.00},{94.09,121.08},{57.70,82.98},{67.96,100.92},
331.         {81.91,132.34},{11.55,39.74},{86.59,126.05},{ 5.36,41.72},
332.         {90.86,144.15},{81.02,137.56},{35.87,81.76},{63.73,105.92},
333.         {78.29,129.54},{96.72,150.04},{14.97,61.93},{45.76,77.17},
334.         {82.69,123.95},{85.82,132.89},{85.95,127.24},{15.04,36.92},
335.         {89.91,112.87},{30.86,58.13},{ 5.77,42.22},{75.24,108.41},
336.         { 8.43,32.09},{90.70,147.99},{80.16,112.57},{42.81,73.54},
337.         {82.47,123.41},{48.23,98.48},{77.48,143.96},{ 0.48,14.50},
338.         {29.75,63.12},{88.76,137.72},{33.59,70.61},{22.74,43.51},
339.         {82.15,116.11},{89.10,120.65},{26.56,68.17},{40.72,74.98},
340.         {68.46,99.23},{34.82,66.71},{36.56,67.33},{72.32,114.23},
341.         {29.65,65.99},{44.39,64.83},{82.08,116.35},{99.73,139.12},
342.         {79.04,118.48},{20.78,42.05},{72.39,96.47},{90.62,147.11},
343.         {35.99,59.11},{50.65,83.23},{59.04,100.47},{87.01,145.78},
344.         {43.71,76.56},{95.61,151.81},{50.25,88.96},{69.64,122.07},
345.         {40.07,79.38},{82.61,133.63},{20.84,39.75},{10.28,42.50},
346.         {47.43,70.82},{30.47,67.19},{69.16,100.10},{46.06,74.00},
347.         {93.78,152.76},{19.93,67.46},{79.61,130.88},{81.11,120.11},
348.         {76.16,123.94},{75.84,111.70},{50.97,85.30},{47.35,90.59},
349.         {93.21,115.44},{19.22,39.30},{11.67,29.58},{52.48,95.64},
350.         {38.76,59.62},{ 2.74,-2.03},{18.99,63.67},{82.38,128.08},
351.         {15.68,32.34},{39.19,83.38},{31.06,65.92},{28.91,73.05},
```

```
352.         {19.01,59.69},{76.62,117.74},{36.82,91.33},{86.28,121.19},
353.         {39.26,50.72},{41.45,70.26},{65.81,111.41},{77.09,117.88},
354.         {78.96,128.48},{16.41,56.61},{39.54,64.11},{72.45,110.54},
355.         {48.83,77.35},{27.61,51.82},{26.53,47.44},{83.06,111.09},
356.         {97.06,127.57},{89.01,146.82},{89.44,141.17},{69.18,100.25},
357.         { 1.11,11.60},{71.63,123.66},{92.93,151.73},{99.46,165.34},
358.         {36.49,71.56},{95.48,153.13},{65.33,102.37},{15.28,35.93},
359.         { 5.52,36.67},{ 0.78,42.47},{10.09,36.68},{ 5.75,37.39},
360.         {52.34,89.11},{14.55,47.37},{67.92,113.35},{36.66,77.34},
361.         {99.76,143.75},{26.67,58.72},{ 3.21,39.37},{87.70,124.12},
362.         {90.03,131.24},{51.54,91.39},{62.86,98.04},{52.75,90.87},
363.         {34.17,84.31},{62.00,89.08},{82.47,111.89},{61.38,123.48},
364.         {47.17,84.64},{20.91,53.51},{96.96,131.54},{46.06,85.14},
365.         {26.85,71.44},{91.67,138.51},{54.07,85.26},{51.63,89.63},
366.         {94.04,140.80},{67.75,107.07},{29.24,76.71},{38.29,75.78},
367.         {28.49,72.87},{60.51,102.28},{77.22,107.79},{99.25,145.86},
368.         {33.11,52.32},{72.47,125.80},{21.97,59.23},{14.25,61.11},
369.         {23.79,63.11},{77.78,109.13},{23.51,81.38},{66.92,110.89},
370.         {79.81,109.80},{56.72,94.63},{59.60,110.57},{57.68,104.54},
371.         {27.83,42.43},{47.80,87.68},{58.79,76.51},{78.33,126.71},
372.         {85.14,128.99},{71.61,116.42},{58.09,96.85},{44.89,71.34},
373.         {33.12,80.19},{98.79,130.09},{44.57,82.03},{88.63,142.61},
374.         {61.96,98.55},{58.54,106.80},{19.17,61.00},{13.51,26.68},
375.         {76.68,124.52},{82.62,138.53},{78.13,122.09},{37.10,60.33},
376.         { 8.82,48.63},{71.64,105.27},{68.44,115.07},{ 7.66,61.91},
377.         {64.37,96.58},{54.90,88.28},{78.35,133.29},{79.84,129.58},
378.         { 3.09,28.37},{48.62,76.00},{38.26,63.99},{42.05,102.17},
379.         {48.89,73.66},{54.38,100.05},{11.16,55.43},{63.24,110.69},
380.         {68.17,114.15},{68.68,109.15},{53.43,90.23},{67.45,106.67},
381.         {10.60,34.41},{56.81,90.86},{11.42,27.08},{36.93,93.13},
382.         {41.64,89.77},{69.74,103.98},{23.07,55.12},{44.98,83.65},
383.         {35.75,72.65},{14.80,56.15},{72.19,115.53},{51.10,80.69},
384.         {96.54,140.10},{15.04,62.30},{21.17,56.15},{46.42,79.63},
385.         {22.35,52.01},{35.47,54.95},{ 4.27,21.33},{84.37,139.55},
386.         {43.95,93.24},{86.56,132.82},{44.35,83.36},{76.81,114.79},
387.         { 1.05,31.66},{32.76,76.15},{83.66,120.90},{12.14,42.52},
388.         {25.85,55.83},{82.12,140.05},{75.33,126.93},{32.92,75.90},
389.         { 7.52,24.51},{25.42,41.55},{42.57,67.15},{87.36,150.38},
390.         { 0.51,17.68},{45.70,84.75},{58.74,88.68},{28.62,74.38},
391.         {73.22,113.45},{78.64,114.25},{42.40,92.03},{84.22,132.25},
392.         {54.24,73.34},{ 2.71,30.27},{54.11,84.97},{66.74,112.66},
393.         {28.80,57.88},{87.02,146.20},{32.02,63.03},{59.57,94.41},
394.         {40.46,79.73},{23.74,49.78},{87.58,140.94},{84.15,113.32},
395.         {32.48,63.48},{ 4.59,25.85},{98.00,128.35},{12.23,37.43},
396.         {66.17,102.97},{50.73,93.82},{74.68,137.79},{43.72,92.85},
397.         {53.95,91.99},{54.47,105.25},{56.70,104.89},{16.59,46.52},
398.         {71.56,115.18},{80.62,99.79},{71.29,101.42},{16.81,56.15},
```

```
        {48.88,84.93},{ 8.41,40.02},{93.98,147.39},{39.20,86.04},
        {61.75,90.80},{ 1.06,32.69},{21.40,33.33},{ 8.60,28.69},
        {38.80,61.88},{14.41,38.37},{40.14,70.01},{69.45,105.44},
        {14.41,43.93},{51.20,93.11},{39.10,57.10},{21.04,39.51},
        {10.12,30.43},{70.13,93.88},{ 1.74,20.56},{12.23,34.33},
        {98.81,151.87},{50.48,92.07},{ 6.98, 9.52},{24.08,69.94},
        {15.72,40.89},{83.99,127.44},{47.21,90.46},{88.31,138.70},
        {91.05,132.13},{45.22,62.24},{87.76,128.67},{99.37,168.24},
        {94.38,140.24},{31.30,67.65},{40.85,84.03},{40.91,79.56},
        {77.14,135.74},{50.92,80.52},{17.81,49.14},{90.30,135.15},
        {28.44,64.60},{49.23,85.12},{81.63,141.58},{83.04,111.19},
        {28.39,63.30},{ 8.61,44.11},{25.36,50.79},{51.35,93.32},
        {64.49,80.42},{96.17,134.31},{96.10,144.32},{47.58,83.36},
        {94.38,131.03},{41.97,69.05},{37.86,62.21},{26.97,65.30},
        {37.57,88.95},{65.08,108.58},{17.68,39.80},{63.75,103.14},
        {91.86,132.07},{76.35,121.19},{22.98,34.87},{96.46,140.54},
        { 9.38,31.40},{42.97,82.09},{20.56,49.02},{13.73,41.31},
        {37.35,63.18},{69.54,105.57},{38.17,83.30},{47.04,80.34},
        {48.79,98.00},{39.34,61.59},{82.57,125.55},{40.77,82.18},
        {13.62,53.38},{35.33,95.17},{95.36,148.79},{20.25,62.00},
        {47.48,86.54},{30.22,61.07},{83.90,120.30},{85.81,123.25},
        {84.29,130.44},{52.84,95.43},{96.72,140.32},{ 3.29,45.68},
        {71.77,98.66},{ 8.52,42.40},{22.55,54.27},{15.08,47.10},
        {91.29,130.23},{16.48,40.04},{44.84,72.14},{34.44,73.42},
        {36.26,78.30},{58.45,115.51},{96.59,150.22},{63.80,98.30},
        {85.92,120.14},{93.68,129.88},{74.09,119.30},{99.44,136.93},
        {88.39,131.55},{64.40,117.89},{13.87,47.30},{81.17,106.77}
    };
```

Insert a table that shows running times for the original and MPI versions.

SN	Original	MPI
1	0.059453180	0.311163362
2	0.06016975	0.324237720
3	0.059001101	0.270844888
4	0.059596028	0.319727684
5	0.059596028	0.267657759
6	0.059529706	0.330296460
7	0.059663874	0.279251090
8	0.059612447	0.304662446

9	0.059592362	0.276354851
	0.059743382	0.285958179

Write a short analysis of the results

2 Verbose Repository Log

```
Paste your verbose format repository log here. With subversion this can be achieved by the following:

        svn update

        svn -v log > log.txt

        gedit log.txt

Then select, copy and paste the text here
```