HERALD
COLLEGE
KATHMANDU

6CS005– High Performance Computing

Student Name: Sabin Ghatani

Student Id: 1828467

B.Sc. (Hons) Computer Science

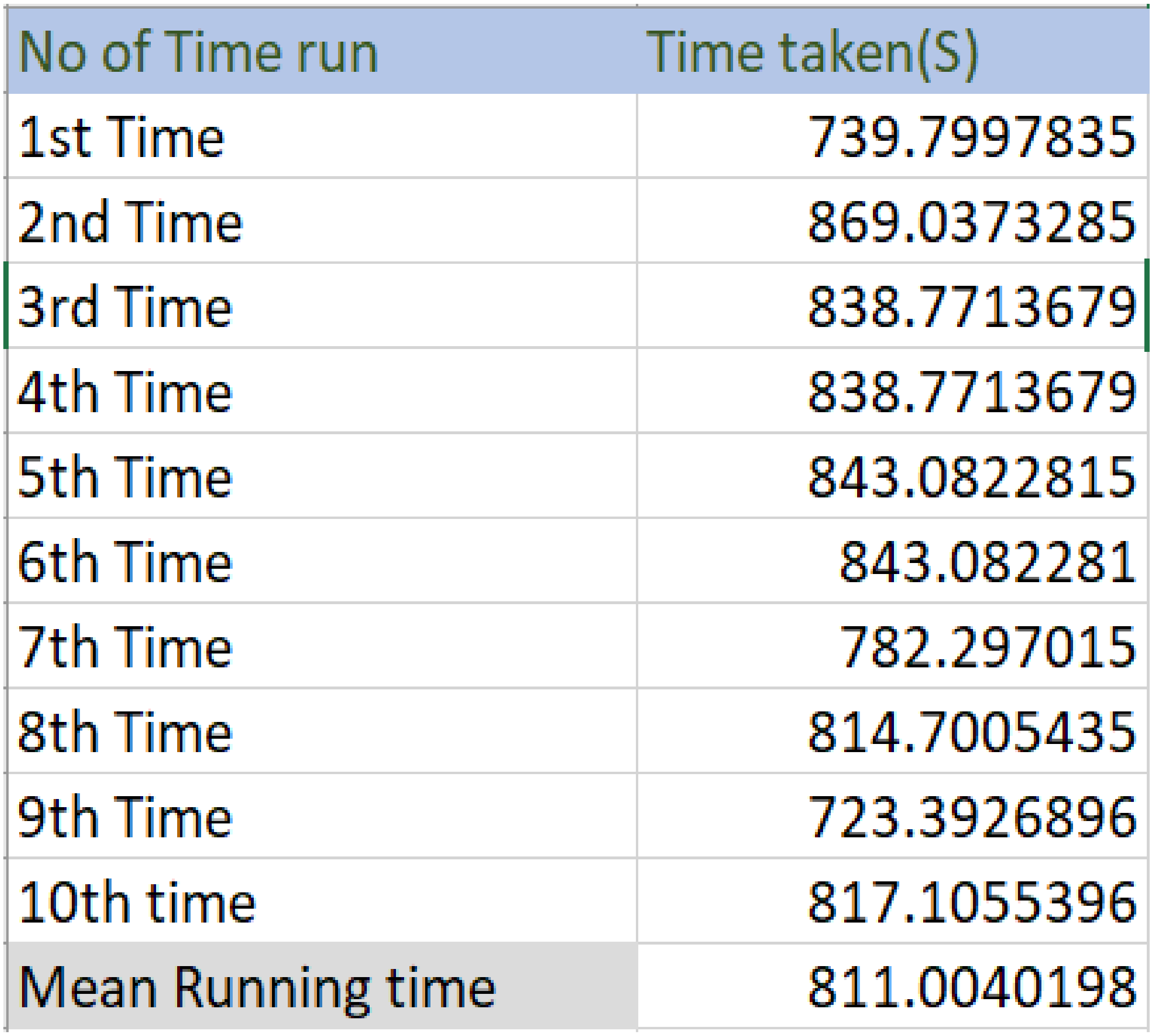
Submitted date: Jan 05, 2019

Contents

1. POSIX Threads

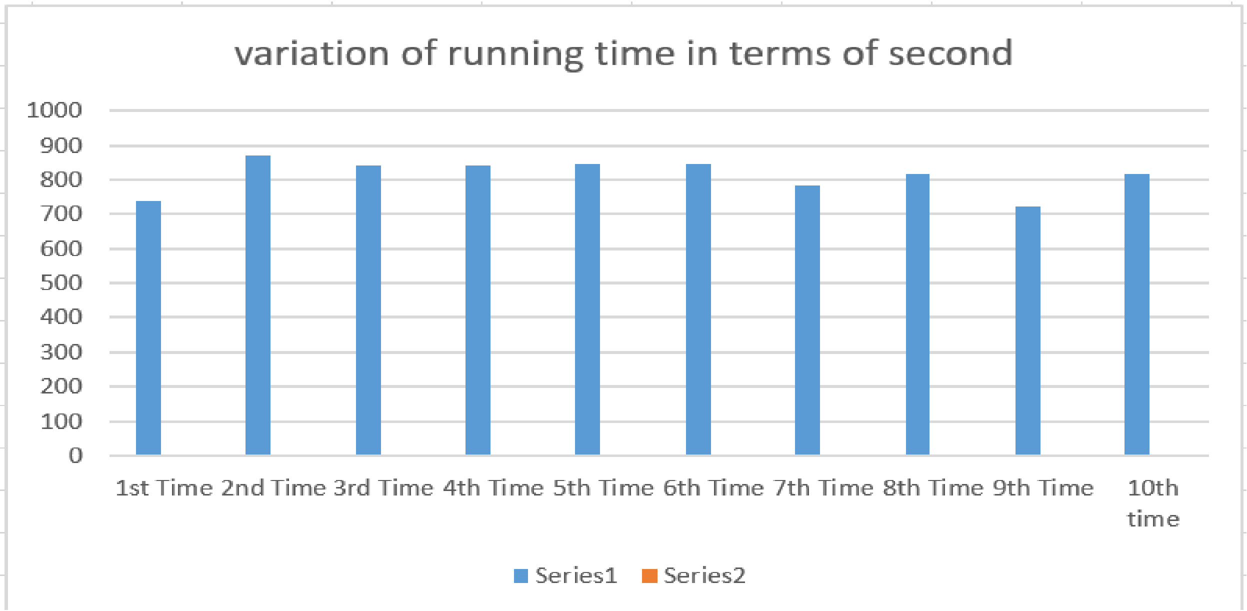
1.1 Password Cracking

The running time of program when it runs for 10 times



No of Time run	Time taken(S)
1st Time	739.7997835
2nd Time	869.0373285
3rd Time	838.7713679
4th Time	838.7713679
5th Time	843.0822815
6th Time	843.082281
7th Time	782.297015
8th Time	814.7005435
9th Time	723.3926896
10th time	817.1055396
Mean Running time	811.0040198

Chart in term of second to see how program time varies from each time when run



Hypothesis Time estimation

As we can see when the initials are two the mean running time for the process is 13.50 minute. Which means it takes more than 13 minute to go through two loops and through more than 67 thousand combinations. When an initial is increased by one it goes through alphabet A to Z in an iteration. Which means the program needs to be run 26 times more to find the combinations. As we have mean time of 13.50 for 1 running process when it is run for 26 times the estimated mean time will be

= 26*13.50

= 351 minutes

= 5.85 hour

Hence the projected estimated time for running a program when an initial is increased by one is 5 hour and 51 minutes.

Code of password cracking of three initial

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <time.h>
#include<crypt.h>

/**************************************************************
*********
*******
  Demonstrates how to crack an encrypted password using a
simple
  "brute force" algorithm. Works on passwords that consist only
of 2
uppercase
  letters and a 2 digit integer. Your personalised data set is
included
in the
  code.

  Compile with:
    cc -o CrackAZ99-With-Data CrackAZ99-With-Data.c -lcrypt

  If you want to analyse the results then use the redirection
operator
to send
  output to a file that you can view using an editor or the
less
utility:

    ./CrackAZ99-With-Data > results.txt

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

int n_passwords = 4; //assinng no of password
//passwords
char *encrypted_passwords[] = {

"$6$KB$3MiAO5oLs/.coZCPQ2QYOy8Ozo3v7QzGdwBEv3N7E0pJen3CJ63DmYXI
Zz6KEsykHmGsu3Dh1KCNe0niN0wvx/",
"$6$KB$J4IvaQyaBTaxqgcXSRy1ekd5MJe8ZEwlgiHVGz1lBlN7E3IoHmL6crz9
230xUd9ciGjXbTf60drGnuUg9mFm20",

"$6$KB$nTF3p1cNAysQAF/gFwaXB.7L2YEguvQi4qLJi/aSkN1MP1vXUGgt36FE
FbdD8tElusQda178XE.kGMBoZgLiB0",

"$6$KB$rmRaZq1xk.DPrBiN3K2XH5mnujGY51hILP8v4RYIllaVDgqr3in5hKnp
m52i9VS/sgJ0OBjD5u62k0sSleQwD/"
};

/**
 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 crack(char *salt_and_encrypted){
  int x, y, z,s;      // Loop counters
  char salt[7];    // String used in hashing the password. Need
space
  char plain[7];   // The combination of letters currently
being checked
  char *enc;       // Pointer to the encrypted password
  int count = 0;   // The number of combinations explored so
far
```

```
  substr(salt, salt_and_encrypted, 0, 6);

  for(x='A'; x<='Z'; x++){
    for(y='A'; y<='Z'; y++){
       for(s='A'; s<='Z'; s++){
           for(z=0; z<=99; z++){
          sprintf(plain, "%c%c%02d", x, y, z);
          enc = (char *) crypt(plain, salt);
          count++;
          if(strcmp(salt_and_encrypted, enc) == 0){
            printf("#%-8d%s %s\n", count, plain, enc);
          } else {
            printf(" %-8d%s %s\n", count, plain, enc);
          }
        }
      }
    }
  }
  printf("%d solutions explored\n", count);
}


//Calculating time for the program to run
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(int argc, char *argv[]){ //main method
  int i;
struct timespec start, finish;
  long long int time_elapsed;

  clock_gettime(CLOCK_MONOTONIC, &start);

  for(i=0;i<n_passwords;i<i++) {
    crack(encrypted_passwords[i]);
  }
  clock_gettime(CLOCK_MONOTONIC, &finish);
  time_difference(&start, &finish, &time_elapsed);
  printf("Time elapsed was %lldns or %0.9lfs\n", time_elapsed,
                                         (time_elapsed/1.0e9));

  return 0;
}
```

Comparison of 3 initial hypothesis and actual result

When the program is run with one extra initial the program execution time is 5.0835 hour. Which means it takes 5 hour and 5 minutes. Which is 47 minutes less than my initial hypothesis. C program execute faster than other languages as it does not need to change the codes into other form to execute. C compiler directly optimizes and compiles and generates machines codes. (Haufler, 2016) Similarly the background running function makes C programming faster. So it takes less time to execute than my hypothesis.

Code of password cracking using multithread

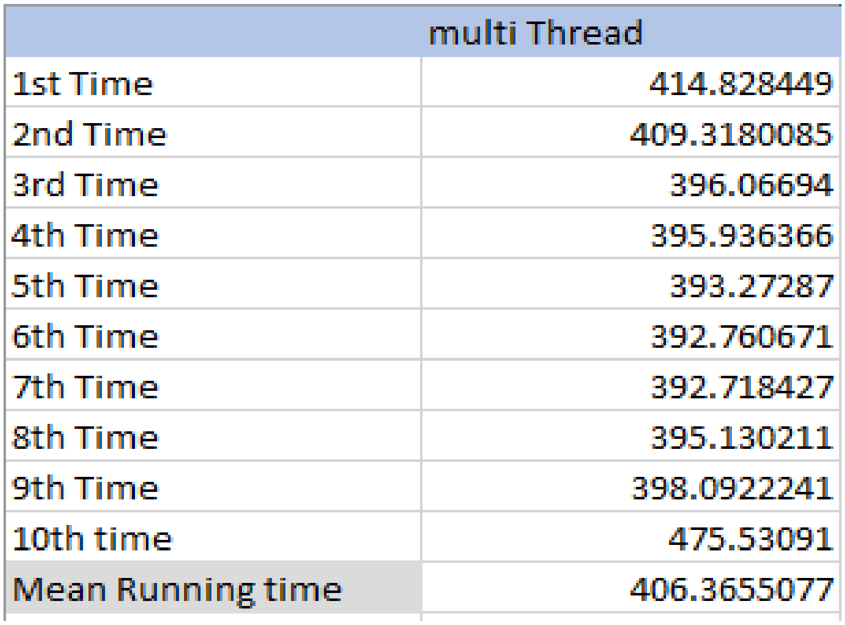
```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <crypt.h>
#include <time.h>
#include <pthread.h>

#include <ctype.h>
#include <errno.h>
#include <sys/stat.h>
#include <string.h>

#include <math.h>


/***************************************************************
*********
***
  Demonstrates how to crack an encrypted password using a
simple
  "brute force" algorithm. Works on passwords that consist only
of 2
uppercase
```

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

  Compile with:
    cc -o ab ab.c -lcrypt

  If you want to analyse the results then use the redirection
operator
to send
  output to a file that you can view using an editor or the
less
utility:

    ./CrackAZ99-With-Data > results.txt

  Dr Kevan Buckley, University of Wolverhampton, 2018
***************************************************************
*****
******/
int n_passwords = 4;

char *encrypted_passwords[] = {

"$6$KB$3MiAO5oLs/.coZCPQ2QYOy8Ozo3v7QzGdwBEv3N7E0pJen3CJ63DmYXI
Zz6KEsykHmGsu3Dh1KCNe0niN0wvx/",

"$6$KB$J4IvaQyaBTaxqgcXSRy1ekd5MJe8ZEwlgiHVGz1lBlN7E3IoHmL6crz9
230xUd9ciGjXbTf60drGnuUg9mFm20",

"$6$KB$nTF3p1cNAysQAF/gFwaXB.7L2YEguvQi4qLJi/aSkN1MP1vXUGgt36FE
FbdD8tElusQda178XE.kGMBoZgLiB0",

"$6$KB$rmRaZq1xk.DPrBiN3K2XH5mnujGY51hILP8v4RYIllaVDgqr3in5hKnp
m52i9VS/sgJ0OBjD5u62k0sSleQwD/"
};

/**
 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 *multi_block1(void *args){
  int x, y, z;     // Loop counters
  char *salt_and_encrypted = args;
  char salt[7];    // String used in hashing the password. Need
space for \0
  char plain[7];   // The combination of letters currently
being checked
  char *enc;       // Pointer to the encrypted password
  int count = 0;   // The number of combinations explored so
far

  substr(salt, salt_and_encrypted, 0, 6);

  for(x='A'; x<='M'; x++){
    for(y='A'; y<='Z'; y++){
      for(z=0; z<=99; z++){
        sprintf(plain, "%c%c%02d", x, y, z);
        enc = (char *) crypt(plain, salt);
        count++;
        if(strcmp(salt_and_encrypted, enc) == 0){
          printf("#%-8d%s %s\n", count, plain, enc);
        } else {
          printf(" %-8d%s %s\n", count, plain, enc);
        }
      }
    }
  }
  printf("%d solutions explored\n", count);
}
```

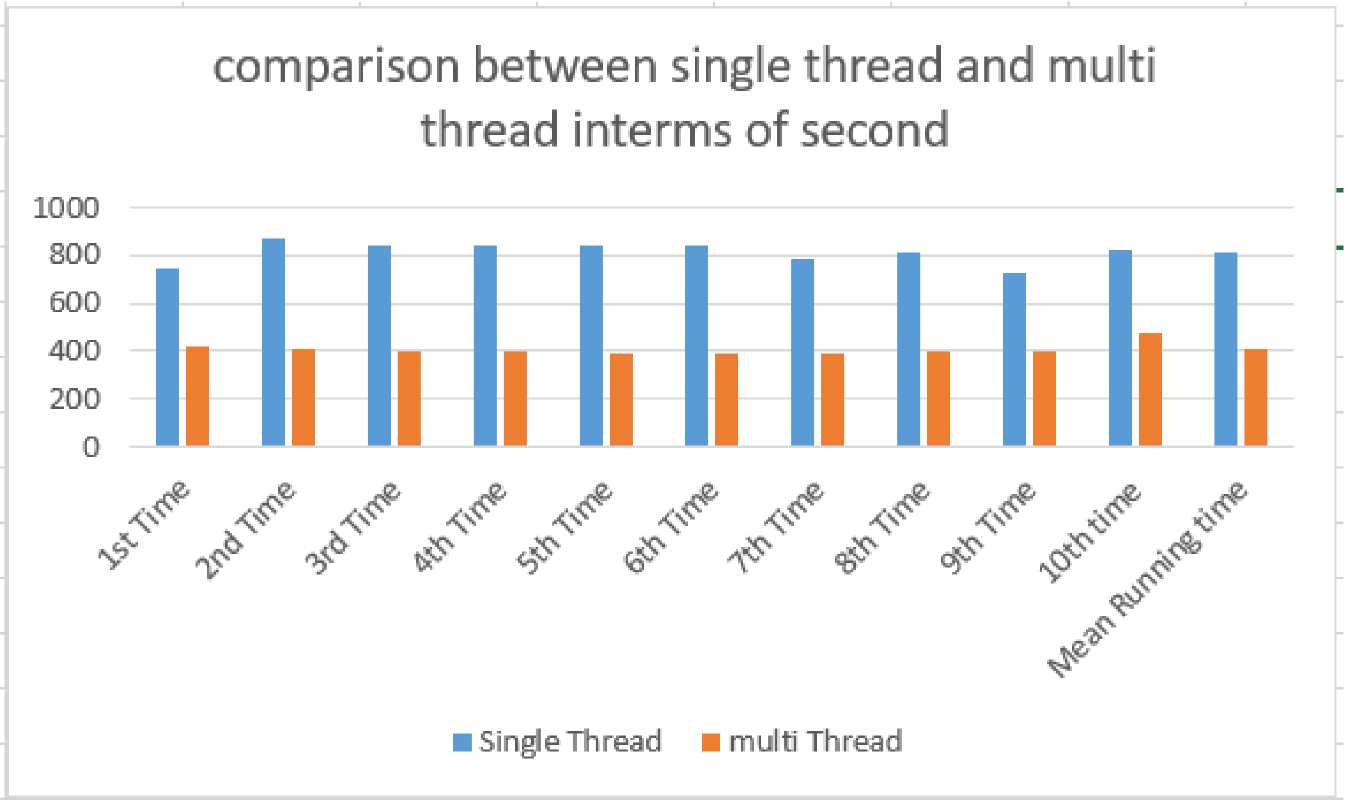
```
void *multi_block2(void *args){
  int x, y, z;      // Loop counters
  char *salt_and_encrypted = args;
  char salt[7];     // String used in hashing the password. Need
space for \0
  char plain[7];    // The combination of letters currently
being checked
  char *enc;        // Pointer to the encrypted password
  int count = 0;    // The number of combinations explored so
far

  substr(salt, salt_and_encrypted, 0, 6);

  for(x='N'; x<='Z'; x++){
    for(y='A'; y<='Z'; y++){
      for(z=0; z<=99; z++){
        sprintf(plain, "%c%c%02d", x, y, z);
        enc = (char *) crypt(plain, salt);
        count++;
        if(strcmp(salt_and_encrypted, enc) == 0){
          printf("#%-8d%s %s\n", count, plain, enc);
        } else {
          printf(" %-8d%s %s\n", count, plain, enc);
        }
      }
    }
  }
  printf("%d solutions explored\n", count);
}

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(int argc, char *argv[]){
  struct timespec start, finish;
  long long int time_elapsed;
   int i;
   pthread_t s1, s2;
  clock_gettime(CLOCK_MONOTONIC, &start);
  for(i=0;i<n_passwords;i<i++) {
    pthread_create(&s1, NULL, multi_block1,(void *)
encrypted_passwords[i]);
    pthread_create(&s2, NULL, multi_block2,(void *)
encrypted_passwords[i]);

    pthread_join(s1, NULL);
    pthread_join(s2, NULL);
 }

  clock_gettime(CLOCK_MONOTONIC, &finish);
  time_difference(&start, &finish, &time_elapsed);
  printf("Time elapsed was %lldns or %0.9lfs\n", time_elapsed,
                                         (time_elapsed/1.0e9));
return 0;
}
```

Running time of password cracking in multithread



	multi Thread
1st Time	414.828449
2nd Time	409.3180085
3rd Time	396.06694
4th Time	395.936366
5th Time	393.27287
6th Time	392.760671
7th Time	392.718427
8th Time	395.130211
9th Time	398.0922241
10th time	475.53091
Mean Running time	406.3655077

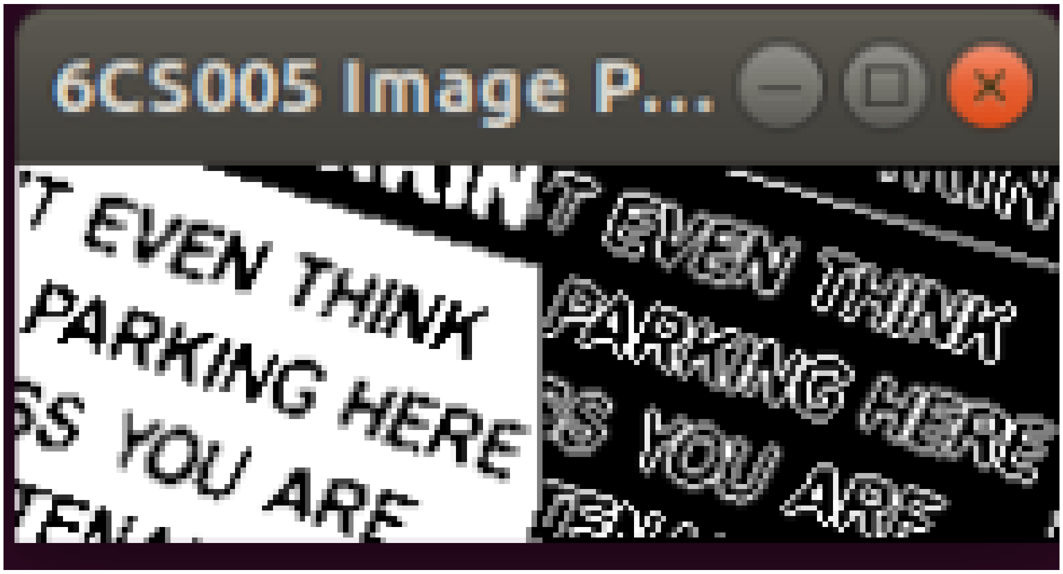
Comparison between original and multithread password cracking results



As we can see from the figure the running time of multithreads are so less than the single thread. In multithread there are two threads which are designed to do same function in a programs due to which it can perform multiple task in at once. Which allows programs to run faster by dividing the tasks and execute faster. So the execution time of multithread is less than the execution of program without thread.

1.2 Image Processing

The image displayed is



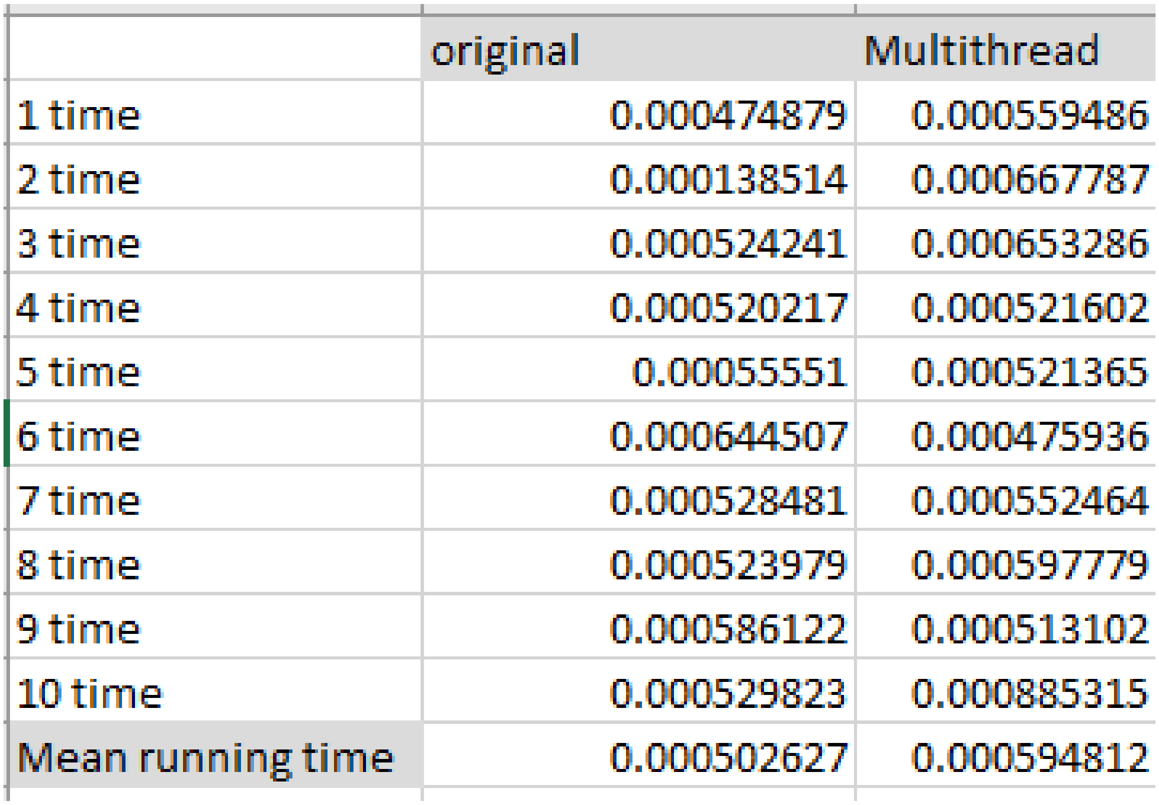
Code of multithread image processing

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <GL/glut.h>
#include <GL/gl.h>
#include <malloc.h>
#include <signal.h>

/**************************************************************
****************
  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:
    cc -o ip_coursework ip_coursework.c -lglut -lGL -lm

  Dr Kevan Buckley, University of Wolverhampton, 2018
****************************************************************
***************/
#define width 100
#define height 72

unsigned char image[], results[width * height];

typedef struct arguments {
unsigned char *input;
unsigned char *output;
  int start;
  int stride;
}
arguments_t;

void edges(unsigned char *image,unsigned char *results) {
  pthread_t t1, t2, t3, t4;

  arguments_t t1_arguments;
  t1_arguments.start = 0;
  t1_arguments.stride = 4;
  t1_arguments.input = image;
  t1_arguments.output = results;


  arguments_t t2_arguments;
  t2_arguments.start = 1;
  t2_arguments.stride = 4;
  t2_arguments.input = image;
  t2_arguments.output = results;

  arguments_t t3_arguments;
  t3_arguments.start = 2;
  t3_arguments.stride = 4;
  t3_arguments.input = image;
  t3_arguments.output = results;

  arguments_t t4_arguments;
  t4_arguments.start = 3;
  t4_arguments.stride = 4;
  t4_arguments.input = image;
  t4_arguments.output = results;

  void *detect_edges();

  pthread_create(&t1, NULL, detect_edges, &t1_arguments);
  pthread_create(&t2, NULL, detect_edges, &t2_arguments);
  pthread_create(&t3, NULL, detect_edges, &t3_arguments);
  pthread_create(&t4, NULL, detect_edges, &t4_arguments);

  pthread_join(t1, NULL);
  pthread_join(t2, NULL);
  pthread_join(t3, NULL);
  pthread_join(t4, NULL);
}



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;
      }
    }
  }
}

void tidy_and_exit() {
  exit(0);
}

void sigint_callback(int signal_number){
  printf("\nInterrupt from keyboard\n");
  tidy_and_exit();
}

static void display() {
  glClear(GL_COLOR_BUFFER_BIT);
  glRasterPos4i(-1, -1, 0, 1);
  glDrawPixels(width, height, GL_LUMINANCE, GL_UNSIGNED_BYTE,
image);
  glRasterPos4i(0, -1, 0, 1);
  glDrawPixels(width, height, GL_LUMINANCE, GL_UNSIGNED_BYTE,
results);
  glFlush();
}

static void key_pressed(unsigned char key, int x, int y) {
  switch(key){
    case 27: // escape
      tidy_and_exit();
      break;
    default:
      printf("\nPress escape to exit\n");
      break;
  }
}


//Calculating time
```

```
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(int argc, char **argv) {
  signal(SIGINT, sigint_callback);

  printf("image dimensions %dx%d\n", width, height);

struct timespec start, finish;
  long long int time_elapsed;

  clock_gettime(CLOCK_MONOTONIC, &start);

  edges(image, results);
  clock_gettime(CLOCK_MONOTONIC, &finish);
  time_difference(&start, &finish, &time_elapsed);
  printf("Time elapsed was %lldns or %0.9lfs\n", time_elapsed,
                                         (time_elapsed/1.0e9));

  glutInit(&argc, argv);
  glutInitWindowSize(width * 2,height);
  glutInitDisplayMode(GLUT_SINGLE | GLUT_LUMINANCE);

  glutCreateWindow("6CS005 Image Progessing Courework");
  glutDisplayFunc(display);
  glutKeyboardFunc(key_pressed);
  glClearColor(0.0, 1.0, 0.0, 1.0);

  glutMainLoop();

  tidy_and_exit();

  return 0;
```

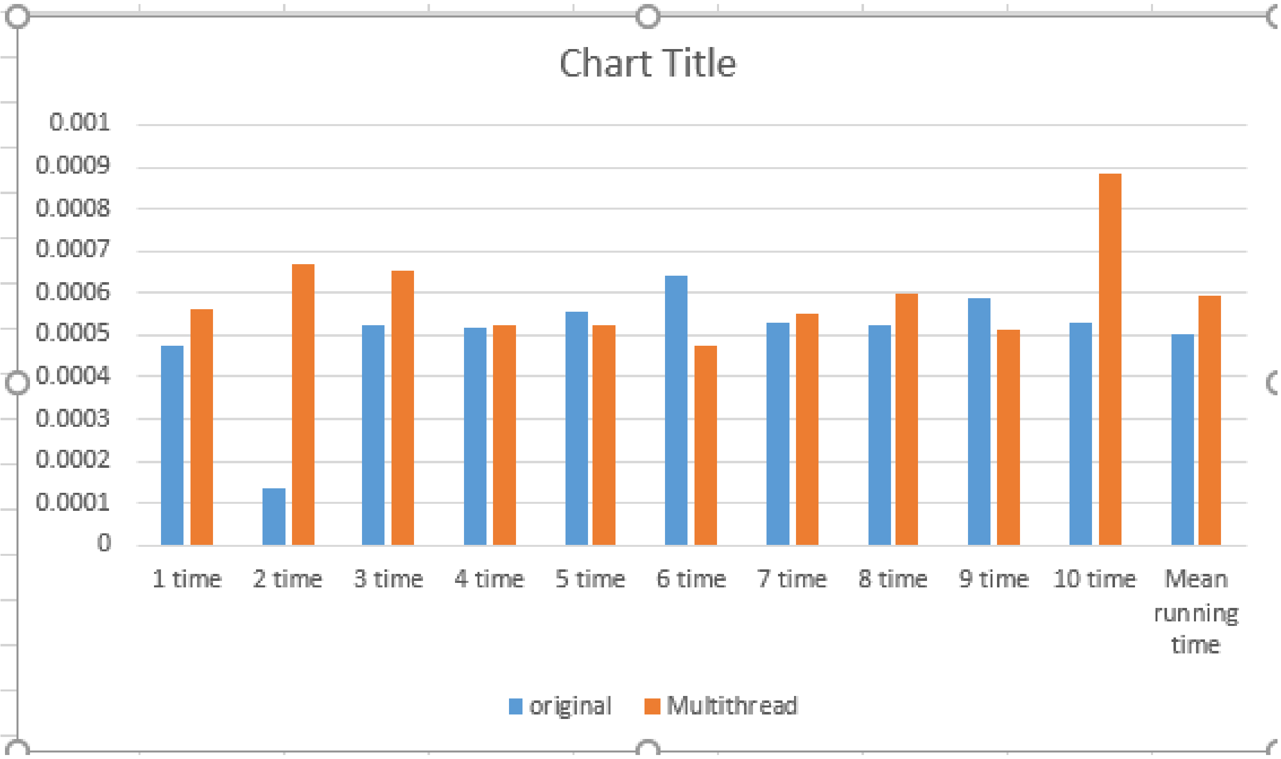
```
}

unsigned char image[] =
{255,255,255,255,0,0,255,255,255,255,255,255,255,0,255,255,0,0,

255,0,0,255,255,255,255,0,0,0,0,255,255,255,255,255,0,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,25
5,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,255,255,255,255,255,255,255,255,255,0,255,255,255,25
5,255,0,0,255,255,255,255,255,255,255,0,0,255,0,0,255,255,0,255
,255,255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,0,255,255,0,0,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,255,255,255,0,0,255,255,255,255,25
5,0,255,255,255,255,0,0,255,255,0,0,0,0,255,255,0,0,255,255,255
,255,255,0,0,255,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,255,255,0,0,255,255,255,0,0,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,255,255,255,0,0,0,255,255,255,255,0,0,0,0,0,0,
0,255,255,0,0,0,0,255,255,255,0,255,255,255,255,255,255,255,255
,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
,0,255,255,255,255,0,0,255,255,255,0,0,255,255,0,0,0,255,255,25
5,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,255,0,0,255,255,255,255,255,0,0,0,255,255,255,255,
255,255,0,0,
0,255,255,255,0,0,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,0,
255,255,0,0,255,255,255,0,0,255,255,255,255,0,0,0,0,0,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,0,0,0,255,255,255,255,0,0,255,255,255,2
55,
255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,0,0,255,0,0,255,255,255,0,0
,255,255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,0,255,0,255,255,255
,255,255,
0,255,255,255,255,0,0,255,255,0,255,255,255,255,255,255,255,255
,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,0,0,255,255,255,255,0,0,0,255,255,0,0,0,0,0,0,

0,0,255,255,255,0,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,0,
  0,255,255,255,255,0,0,0,0,0,0,0,255,255,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,255,255,0,
  0,0,0,255,255,255,0,0,255,255,0,0,255,255,0,0,0,0,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,0,0,255,255,255,255,0,0,0,0,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,0,255,255,255,255,255,255,255,255,255,255,0,0,0,255
,255,
  0,0,255,255,255,0,0,255,255,255,255,0,0,255,255,0,0,0,0,

0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,

255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,0,0,0,0,0,0,255,255,255,255,255,255,255,255
,
  255,0,0,255,255,0,0,255,255,255,0,0,255,255,255,255,0,0,255,
255,255,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,255
,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,0,0,255,255,255,0,0,255
,255,

255,255,255,255,255,255,255,0,0,255,0,0,255,255,255,255,0,0,255
,

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
,255,255,0,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,0,0,0,255,255,255,0,0,255,255,2
55,

255,255,0,255,255,255,255,255,255,255,255,255,255,0,0,0,0,255,2
55,

255,255,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,255,

255,0,0,255,255,255,255,0,0,255,255,255,255,255,255,255,255,255
,0,

0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,
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,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0
,0,255,

255,255,0,0,0,255,0,0,255,255,255,255,255,0,255,255,255,255,255
,

255,255,255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,2
55,255,255,

255,0,0,255,255,255,255,255,0,0,255,0,0,255,255,255,255,255,0,

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,0,0,0,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,0,255,255,255,255,0,0,255,255,255,255,255,255,0,0,255,0,0,

255,255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
255,255,255,255,255,255,255,255,0,0,255,255,0,0,255,255,255,255
,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,0,0,255,255,255,0,0,255,255,255,255,255
,255,

0,0,255,255,0,255,255,255,255,255,0,0,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,255,255
,255,0,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,0,255,255,255,2
55,0,0,

255,255,255,255,255,255,0,255,255,0,0,255,255,255,255,255,0,255
,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,0,255,0,0,255,255,255,2
55,0,

0,255,255,255,0,0,255,0,0,0,0,255,255,255,255,255,255,255,255,

255,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,0,0,

255,255,255,0,0,255,255,255,255,255,0,0,255,255,0,0,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,255,255,0
,
  0,255,255,255,0,0,255,255,255,0,0,0,0,0,255,255,255,255,255,
255,255,255,255,255,0,0,0,255,255,0,0,255,255,255,255,255,255,2
55,
  255,255,255,0,0,0,0,255,255,0,0,255,255,255,255,0,0,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0
,255,

255,255,255,255,0,0,255,255,0,0,0,255,255,255,255,0,0,255,255,

255,255,255,255,255,255,0,255,255,255,0,0,255,255,255,255,0,0,2
55,
  255,255,255,255,255,255,255,255,0,0,0,0,0,255,255,0,0,0,0,

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
,0,0,255,
  255,0,0,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,255,

255,0,0,255,255,255,255,255,255,255,255,0,0,0,255,0,0,255,255,

255,255,0,0,255,255,255,255,255,255,255,255,255,0,255,255,255,0
,0,

255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,0,0,255,255,0,0,255,255,255,255,255,255,255,0,0,0,255,
  255,255,255,0,0,255,255,0,0,255,0,0,0,0,255,255,255,0,0,

255,255,255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,2
55,0,

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,
  255,255,0,255,255,255,255,255,0,0,255,255,0,0,255,0,0,0,0,
  255,255,0,0,255,255,255,255,0,0,255,255,0,0,0,0,0,255,255,

255,255,255,255,0,0,255,255,255,0,0,0,255,255,255,255,255,255,2
55,

255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,0,0,255,255,255,255,0,0,255,255,0,0
,

0,0,0,255,255,255,255,0,0,255,255,255,255,0,0,255,255,255,255,

255,255,255,255,255,255,255,255,0,0,255,255,255,0,0,0,255,255,2
55,

255,255,255,255,255,255,255,255,255,0,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,0,0,0,255,255,255,255,255,255,255,0,0,255,255,255,255,0
,
  0,0,255,255,0,0,255,255,255,255,255,255,255,0,0,0,0,0,0,

255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,255,255,2
55,0,

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,

255,255,255,255,255,0,0,0,0,0,0,255,255,255,255,255,255,255,0,
  0,0,0,0,0,0,0,255,255,0,0,255,255,255,255,255,255,255,0,

0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0
,0,

255,255,255,255,255,0,0,255,255,0,0,255,255,255,255,255,255,255
,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,0,0,255,255,255,255,0,0,255
,255,
  255,255,255,255,0,0,0,0,0,255,255,0,0,255,255,0,0,255,0,

0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,0,255,255,255,0,0,0,0,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,0,0,255,255,255,0,0,255,255
,255,

255,0,0,255,255,255,255,255,255,255,0,0,255,255,255,255,0,0,255
,

255,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,2
55,

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,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,2
55,255,

0,0,255,255,255,0,0,0,255,255,255,255,255,255,255,0,0,255,255,

255,255,255,0,0,255,255,0,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,0,255,255,0,0,255,2
55,255,

0,0,0,255,255,0,0,255,255,255,255,255,255,255,255,255,255,255,2
55,

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
,255,255,0,

0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,255,255,0
,0,255,

255,0,255,255,255,0,0,0,0,255,0,0,255,255,255,255,255,255,255,

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
,255,0,0,
  255,255,0,0,255,255,0,0,255,255,0,0,255,0,255,255,0,0,255,

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
,255,0,255,

255,255,255,0,0,255,255,255,0,0,255,255,0,255,255,0,255,255,0,
0,255,0,0,255,255,255,255,255,0,0,255,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  0,255,255,0,0,255,255,255,0,0,255,255,255,0,0,255,255,0,0,
  0,0,255,255,255,0,255,255,0,0,0,0,0,0,0,255,255,255,255,

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,

255,255,255,255,0,0,255,255,0,0,255,255,255,0,0,255,255,255,255
,
  0,0,255,0,0,0,0,255,255,255,0,0,255,255,255,0,0,0,0,

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,

255,255,255,255,0,255,255,255,255,0,0,255,255,255,0,0,0,255,0,

0,255,255,255,255,0,0,255,255,0,0,0,255,255,255,255,0,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,255
,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,0,255,255,0,0,255,255,255,0,0,255,255,2
55,
  0,0,0,0,0,0,255,255,255,255,255,0,0,255,0,0,0,255,255,

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,

0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,

255,255,255,255,255,255,255,255,255,255,255,0,255,255,0,0,255,2
55,255,
  0,0,255,255,255,255,0,0,0,0,0,0,255,255,255,255,0,0,255,

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
,255,255,0,

0,255,255,255,255,0,0,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,0,0,255,255,255,255,0,0,0,2
55,
  255,0,0,0,0,0,0,0,0,255,255,0,0,255,255,0,0,0,0,

255,255,255,0,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,0,0,255,255,0,0,255,255,255,0,0,255,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0
,0,0,
  0,0,0,0,255,255,0,0,0,0,255,255,0,0,0,255,255,0,0,
255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,0,255,255,255,0,0,0,255,255,0,0,0,255,255,0,0,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,0,0,255,255,255
,255,255,
  255,0,0,0,0,0,255,0,0,255,255,255,0,0,255,255,255,255,0,

0,255,255,0,0,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,0,0,255,0,0,255,255,255,0,0,0,0,255,255,0,0,0,

0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,0,
  0,255,255,255,255,255,255,0,0,0,255,255,0,0,255,255,255,0,0,

255,255,255,255,0,0,255,255,255,0,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,0,255,255,0,0,255,255,0,0,255,255,0,0,0,0,

255,255,0,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255
,

255,255,255,255,0,0,255,255,255,255,255,255,255,0,0,255,255,0,0
,

255,255,255,255,0,0,255,0,0,0,0,255,255,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,0,0,255,255,0,0,255,0,0,255
,
  255,0,0,255,0,0,255,255,0,0,255,0,0,0,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,255,255,
  0,0,255,0,0,255,255,255,255,0,0,0,0,0,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,0,0,255,255,255,255,255,0,255,2
55,0,
  0,255,255,0,0,255,0,0,255,255,0,255,255,0,0,255,255,255,0,

0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,2
55,

255,255,0,0,255,255,0,0,0,0,255,255,255,255,255,0,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,255,2
55,255,255,
  255,0,0,255,255,0,0,255,255,0,0,0,0,255,255,0,0,255,255,

0,0,255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,0,0,255,255,255,255,0,255,255,255,0,0,0,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,255,255
,255,255,
  0,0,255,255,255,255,0,0,255,255,0,0,255,255,0,0,0,0,255,

255,255,0,255,255,0,0,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,0,0,255,255,255,0,0,0,255,255,255,0
,

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
,255,255,0,
  0,255,255,255,255,255,0,0,255,0,0,0,0,0,255,255,0,0,255,
0,0,0,0,255,255,255,0,0,255,255,255,255,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,255,0,0
,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,
  255,255,255,255,255,0,0,255,255,255,255,0,0,0,0,0,255,0,0,

255,255,0,0,255,255,0,0,0,255,255,255,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,0,

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
,255,0,0,

0,255,255,255,255,255,255,255,255,255,0,0,255,255,255,255,255,0
,0,

255,255,255,255,0,0,255,255,0,0,255,0,0,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,0,0,2
55,

255,255,255,0,0,255,255,255,255,0,0,255,255,0,0,255,255,255,255
,
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
,255,255,0,
  0,0,255,0,0,255,255,255,0,0,0,0,255,255,255,255,255,255,255,

255,255,0,0,255,255,255,255,0,0,255,255,255,255,255,0,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,0,0,0,0,0,255,255,255,0,0,255,255,0,0,0,0,255,255,

255,255,255,255,255,255,255,255,0,0,255,255,255,255,0,0,255,255
,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,

255,255,255,255,255,0,0,0,255,255,255,255,255,255,0,0,255,255,0
,
  0,255,0,0,255,255,255,255,255,255,255,255,255,0,0,0,0,0,255,

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,

255,255,255,255,255,0,255,255,255,255,255,0,0,255,255,255,255,2
55,255,

255,0,0,255,0,0,255,255,0,255,255,255,255,255,255,255,255,0,0,

0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,0,0,0,255,255,255,255,0,0,2
55,

255,255,255,255,255,255,0,0,0,0,0,255,255,0,0,255,255,255,255,

255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,0,0,0,0,0,0,0,255,255,255,255,255,255,255,255,255,

255,255,255,255,255,255,255,255,0,0,255,255,255,255,0,0,0,0,255
,
  255,255,255,0,0,0,0,0,0,0,255,255,0,0,0,0,255,255,0,

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,255,255,255,2
55,
  0,0,0,0,0,255,255,255,0,0,0,0,0,255,255,255,255,0,0,
0,255,255,255,255,0,0,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,0,

0,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,255
,255,

255,255,255,255,255,0,0,255,255,0,0,255,255,255,0,0,255,255,255
,

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,
  255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255
,

0,0,255,255,255,255,255,255,255,255,255,0,255,255,255,0,0,255,2
55,

0,0,255,255,255,255,0,0,255,0,0,255,255,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,255,255,255,255,255,255,255,255,

255,255,255,255,255,255,0,255,255,255,255,255,255,255,255,255,0
,0,255,
  255,255,0,0,255,255,0,0,0,0,0,0,0,255,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,255,255,255,

0,255,255,255,255,255,255,255,255,255,255,0,0,255,0,0,0,0,255,
  255,255,0,0,255,255,255,0,0,0,255,0,0,0,0,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0
,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,255,255,255,0,

0,0,255,255,255,0,0,255,255,255,255,255,255,255,255,255,0,0,0,

0,0,255,255,255,255,255,0,0,255,255,255,255,0,0,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0
,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

255,255,255,255,0,0,0,255,255,255,255,0,255,255,255,255,255,255
,255,

255,255,255,0,0,255,255,255,255,255,255,255,0,0,255,255,255,255
,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

255,255,0,0,0,255,255,255,255,255,0,255,255,255,255,255,0,0,255
,

255,255,255,255,255,255,255,255,0,0,255,255,255,255,255,0,255,0
,0,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,255,255,255,0,0,0,255,255,255,255,255,0,255,255,255,
255,255,255,0,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0
,

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,
  255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,255,255,255,0,0,255,255,255,0,0,0,255,255,255,255,

255,0,255,255,255,255,255,255,0,0,255,255,255,255,255,255,255,2
55,255,

0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,255,255,255,0,255,255,255,255,0,255,255,255,255,0,

0,255,255,255,255,255,255,0,255,255,255,255,255,255,0,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,255,255,255,255,0,255,255,255,255,0,

255,255,255,255,0,255,255,255,255,255,255,255,0,255,255,255,255
,255,255,

0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,255,255,0,0,0,255,255,255,255,0,0,

255,255,255,255,0,0,255,255,255,0,255,255,255,255,255,255,255,0
,255,
0,255,255,255,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,255,255,255,0,0,0,255,

255,255,255,0,0,0,255,255,255,0,0,255,255,255,255,255,255,255,2
55,

255,255,255,0,255,0,0,255,0,0,0,255,255,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,255,255,255,0,0,255,255,

255,255,0,0,255,255,255,255,0,0,0,255,255,255,255,0,255,255,255
,

255,255,255,255,0,255,255,255,255,255,0,0,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,255,255,

255,0,0,255,255,255,255,0,0,255,255,255,255,0,0,0,255,255,255,

255,0,0,255,255,255,255,255,255,0,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,255,255,255,255,

0,0,255,255,255,255,0,0,255,255,255,255,255,255,255,255,255,255
,0,

0,0,255,255,255,255,0,0,255,255,255,255,255,255,0,0,255,255,255
};
```

Time of original program and multithread program of image processing



	original	Multithread
1 time	0.000474879	0.000559486
2 time	0.000138514	0.000667787
3 time	0.000524241	0.000653286
4 time	0.000520217	0.000521602
5 time	0.00055551	0.000521365
6 time	0.000644507	0.000475936
7 time	0.000528481	0.000552464
8 time	0.000523979	0.000597779
9 time	0.000586122	0.000513102
10 time	0.000529823	0.000885315
Mean running time	0.000502627	0.000594812

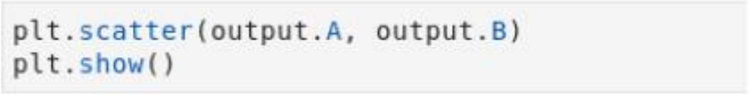
Graph showing comparison between original and multithread of image processing



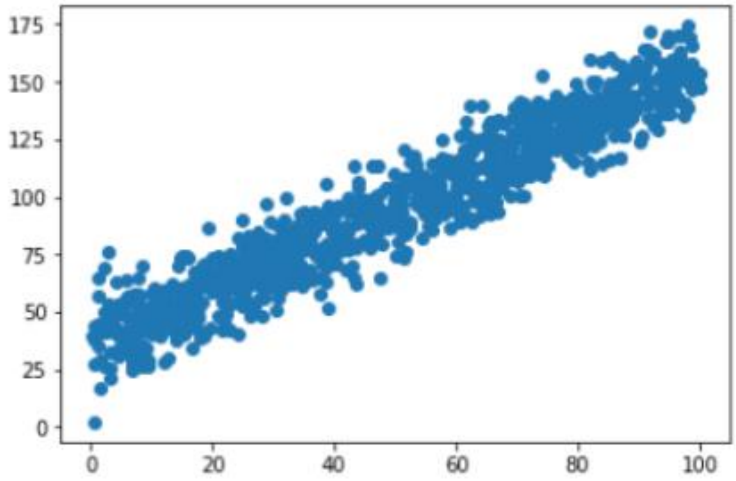
By the above test data and graph we can see the time taken by the multithread to process image is comparatively more than without thread. In general multithread takes shorter time than without thread. But in this case threads are being used and the small image pixel is being segmented and given to two different thread. Which gets processed in each thread. After the work is done it again merges both output and display which causes this program time consuming. Hence, in this program multithread takes more time than without thread.

1.3 Linear Regression

The figure of scatter plot of my data



```
plt.scatter(output.A, output.B)
plt.show()
```



For the linear regression problem I have guessed three different values of m and c which are

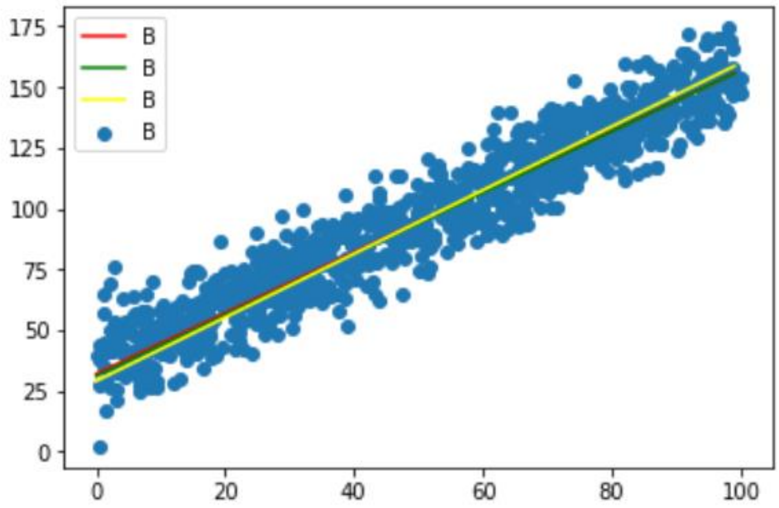
For the 1st guess: m=1.25 c=32

For the 2nd guess: m=1.26 c=31

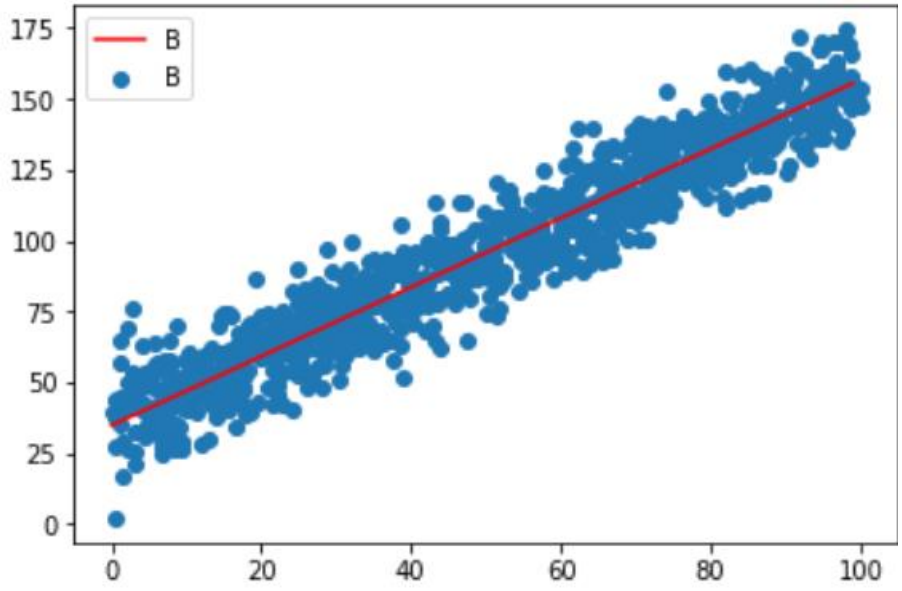
For the 3rd guess: m=1.31 c=29

M and c are the slope and the intercept of a line passing from a b scatter. By the look of scatter it seems like it is inclined to be 45 degree or 40 degree. I have guessed three random values which I think will intersect the scatter from the closest of all.

When the three data is plotted in a scatter diagram it gives the result which is shown below:



The actual result of regression when plotted to m=1.22 and c = 34.78 is shown below



Comparison with my guess

The guesses that I have done is closer to what is expected. The third guess seems to be little closer to the expected line besides other two. So my guess of regression is quite near of actual result. If I have guessed a upper value then it might touch the actual line.

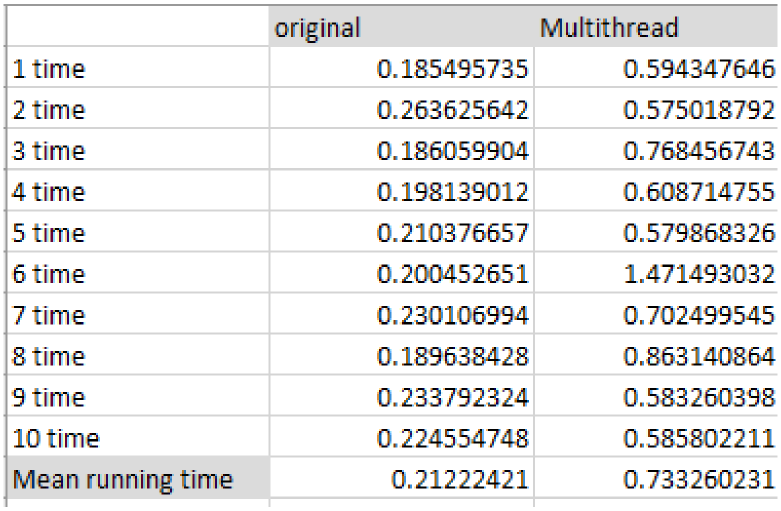
Linear Regression using Multithread using POSIX.

```
#include <stdio.h>
#include <math.h>
#include <pthread.h>
#include <time.h>
/**************************************************************
****************
 * This program takes an initial estimate of m and c and finds
the associated
 * rms error. It is then as a base to generate and evaluate 8
new estimates,
 * which are steps in different directions in m-c space. The
best estimate is
 * then used as the base for another iteration of "generate and
evaluate". This
 * continues until none of the new estimates are better than
the base. This is
 * a gradient search for a minimum in mc-space.
 *
 * To compile:
 *   cc -o lr_coursework lr_coursework.c -lm
 *
 * To run:
 *   ./lr_coursework
 *
 * Dr Kevan Buckley, University of Wolverhampton, 2018

***************************************************************
**************/





double bm = 1.3;
  double bc = 10;
  double be;
  double dm[8];
  double dc[8];
  double e[8];
  double step = 0.01;
  double best_error = 999999999;
  int best_error_i;
  int minimum_found = 0;
double om[] = {0,1,1, 1, 0,-1,-1,-1};
```

```
  double oc[] = {1,1,0,-1,-1,-1, 0, 1};


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);
}

void *thread_function(void *args){
int *a = args;
int i = *a;

 dm[i] = bm + (om[i] * step);
      dc[i] = bc + (oc[i] * step);

e[i] = rms_error(dm[i], dc[i]);
      if(e[i] < best_error) {
        best_error = e[i];
        best_error_i = i;
pthread_exit(NULL);
}
}
```

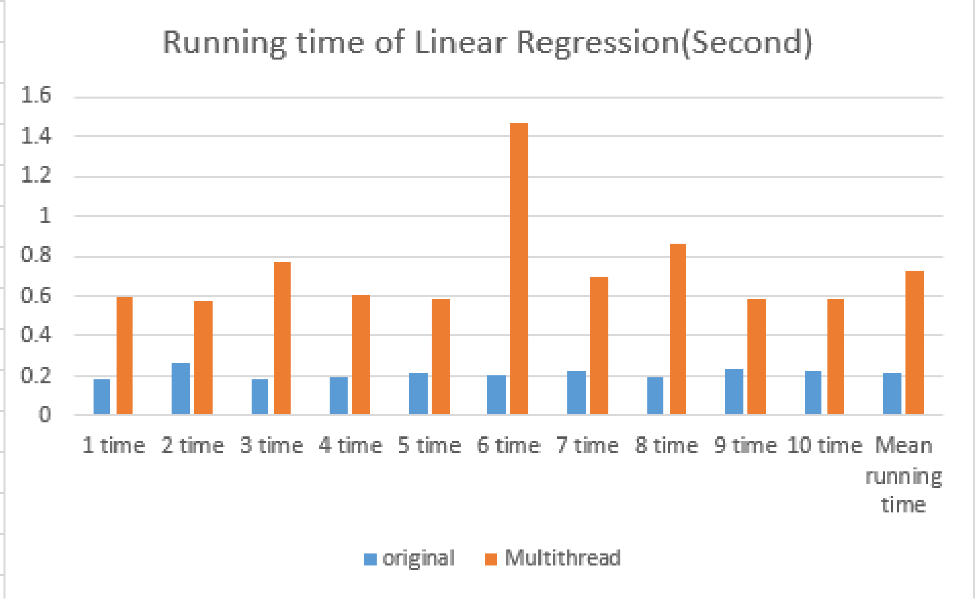
```
int time_difference(struct timespec *start,
                    struct timespec *finish,
                    long long int *difference) {
  long long int ds =  finish->tv_sec - start->tv_sec;
  long long int dn =  finish->tv_nsec - start->tv_nsec;

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


int main() {
struct timespec start, finish;
  long long int time_elapsed;
   clock_gettime(CLOCK_MONOTONIC, &start);
int i;

pthread_t threads[8];

  be = rms_error(bm, bc);

//calculate times




  while(!minimum_found) {

      for(i=0;i<8;i++) {
     pthread_create(&threads[i], NULL,(void*)thread_function,
&i);
     pthread_join(threads[i], NULL);
}




    printf("best m,c is %lf,%lf with error %lf in direction
%d\n",
      dm[best_error_i], dc[best_error_i], best_error,
best_error_i);
    if(best_error < be) {
      be = best_error;
      bm = dm[best_error_i];
      bc = dc[best_error_i];
    } else {
      minimum_found = 1;
    }

  }

clock_gettime(CLOCK_MONOTONIC, &finish);
  time_difference(&start, &finish, &time_elapsed);
  printf("Time elapsed was %lldns or %0.9lfs\n", time_elapsed,
                                          (time_elapsed/1.0e9));
  printf("minimum m,c is %lf,%lf with error %lf\n", bm, bc,
be);


pthread_exit(NULL);


  return 0;
}

point_t data[] = {
  {77.96,128.03},{65.68,95.52},{82.85,133.97},{87.44,126.26},
  {65.25,115.66},{65.81,109.29},{78.07,137.65},{73.81,119.15},
  {83.67,130.20},{83.29,130.15},{65.45,92.82},{14.28,69.92},
  {59.72,113.57},{51.21,105.39},{18.33,70.24},{17.57,42.36},
  {19.28,86.46},{61.11,110.70},{ 3.06,32.92},{66.61,132.11},
  {50.28,86.85},{16.14,56.57},{71.80,131.85},{76.53,143.87},
  {22.16,49.15},{30.64,87.42},{58.97,98.66},{10.46,46.54},
  {65.53,106.21},{91.93,143.24},{73.47,128.02},{20.95,64.18},
  {62.50,110.38},{18.18,54.18},{21.48,57.86},{28.50,60.11},
  { 7.54,48.39},{27.41,78.05},{49.86,91.89},{27.28,85.08},
  {53.79,104.68},{21.85,57.80},{35.55,66.67},{95.93,155.19},
  {57.39,104.01},{42.54,96.87},{67.66,103.79},{82.65,134.83},
  {56.97,100.17},{64.07,112.48},{87.60,146.67},{36.95,90.02},
  {24.78,62.26},{65.78,106.87},{72.22,123.40},{85.91,138.57},
  {22.21,58.65},{23.45,65.71},{34.59,66.36},{40.13,82.01},
  {19.99,73.66},{47.56,101.54},{ 8.38,26.03},{61.23,96.47},
  {52.33,115.23},{61.95,116.68},{84.06,132.97},{47.14,96.40},
  {10.24,56.26},{42.03,87.61},{12.97,52.18},{82.86,150.30},
  {30.40,76.91},{66.49,114.01},{42.05,78.38},{68.17,120.08},
  {91.94,142.49},{66.60,97.85},{38.59,106.14},{67.52,114.10},
  { 0.68,39.30},{ 5.86,47.99},{87.24,138.51},{40.64,79.63},
  {85.67,145.03},{ 1.15,34.90},{57.79,100.48},{28.04,48.02},
  {79.74,149.06},{39.70,84.37},{47.29,113.66},{70.42,141.72},
  {17.33,66.95},{79.96,142.75},{38.66,63.02},{34.16,64.26},
  {82.46,122.55},{39.01,78.63},{36.46,78.62},{28.80,66.12},
  {39.23,75.54},{63.28,97.79},{20.99,74.67},{94.94,136.48},
  {65.50,132.78},{60.50,108.85},{36.39,90.05},{15.92,58.69},
  {56.87,106.49},{79.49,133.03},{65.81,119.26},{61.67,111.30},
  {13.16,62.03},{28.13,76.33},{33.71,71.92},{16.93,67.53},
  { 8.77,48.89},{ 3.03,25.86},{24.15,64.71},{78.16,138.89},
  {66.89,93.13},{70.14,133.04},{69.61,115.60},{60.69,92.18},
  {24.13,40.17},{ 0.62,44.00},{21.36,73.46},{79.55,141.25},
  {37.98,93.03},{ 9.27,51.73},{51.18,88.24},{24.94,78.03},
  {40.98,76.92},{55.01,103.17},{66.86,133.25},{ 6.21,39.08},
  {95.42,145.04},{91.66,159.55},{74.85,129.96},{33.35,82.30},
  {99.98,153.62},{ 7.78,57.71},{43.91,106.75},{85.56,141.85},
  {99.30,154.24},{76.92,135.95},{23.31,71.94},{83.06,124.53},
  {20.73,48.34},{ 7.61,64.85},{ 4.14,62.97},{48.41,93.44},
  {11.08,54.65},{23.22,66.51},{86.51,146.66},{63.93,96.90},
  {62.18,140.01},{54.58,92.40},{14.68,55.65},{74.08,152.67},
  {29.16,78.88},{34.39,70.74},{53.47,111.77},{79.47,115.22},
  {60.26,115.73},{ 4.35,53.66},{44.74,98.91},{36.52,77.68},
  {12.57,46.94},{ 9.35,29.42},{67.14,132.38},{94.48,155.46},
  {60.56,126.66},{82.58,148.39},{40.20,81.60},{97.03,152.57},
  {37.79,88.69},{92.35,131.07},{73.56,141.49},{60.68,89.01},
  {50.87,102.46},{80.10,134.53},{20.10,63.39},{56.11,85.56},
  {17.12,57.21},{43.41,79.14},{66.00,99.99},{55.44,104.05},
  {65.87,112.98},{87.30,140.25},{40.73,84.93},{35.28,76.61},
  {93.55,139.77},{51.48,73.31},{10.10,57.74},{65.11,111.70},
  {16.16,44.28},{62.35,90.61},{ 3.11,21.44},{97.40,152.47},
  {31.24,68.96},{20.00,58.73},{83.50,123.60},{72.10,109.50},
  {18.70,39.72},{80.72,123.39},{71.65,122.57},{47.08,90.29},
  {15.62,61.31},{39.27,88.61},{73.84,133.29},{48.34,93.10},
  {71.71,131.19},{ 4.44,49.00},{88.42,150.90},{34.22,92.45},
  {36.37,63.52},{22.69,60.34},{81.39,132.60},{61.47,104.34},
  {24.19,69.92},{28.64,65.00},{83.10,134.44},{55.87,97.39},
  {91.48,164.21},{86.12,158.13},{18.45,61.30},{70.00,115.55},
  {29.97,70.25},{ 7.80,50.03},{93.35,152.67},{98.77,166.10},
  {76.52,123.74},{76.27,140.22},{79.94,142.14},{ 9.42,26.12},
  {25.14,66.01},{42.29,84.73},{ 7.16,34.35},{34.35,79.77},
  {30.24,60.18},{97.12,158.57},{ 3.77,32.31},{18.59,71.26},
  {23.68,65.65},{77.69,134.56},{74.78,132.51},{38.54,79.86},
  { 1.41,29.22},{34.20,81.76},{46.56,83.08},{21.55,54.54},
  {36.12,80.97},{89.12,143.03},{74.69,139.63},{72.38,134.98},
  {90.52,126.38},{33.45,77.65},{14.45,48.11},{ 2.19,69.22},
  {34.24,68.57},{38.58,69.10},{31.53,76.53},{95.11,149.75},
  {82.77,134.49},{17.54,48.12},{27.36,80.42},{74.70,120.86},
  {18.22,61.57},{23.51,62.04},{43.08,64.64},{37.54,75.77},
  {40.53,96.29},{63.39,122.09},{57.60,116.76},{70.50,133.52},
  {51.09,93.63},{19.81,58.37},{16.62,33.92},{40.21,71.13},
  {83.14,140.67},{34.51,67.66},{51.98,88.01},{57.57,125.27},
  { 7.08,39.25},{41.83,82.22},{95.35,152.79},{27.18,68.47},
  {84.15,140.31},{28.04,72.94},{97.86,152.05},{76.82,124.28},
  {95.52,159.59},{14.04,60.42},{ 6.68,25.03},{46.02,85.90},
  {25.07,67.74},{12.15,59.67},{79.23,130.51},{67.33,124.69},
  {19.51,60.84},{65.77,111.26},{51.20,100.38},{79.65,128.45},
  {41.88,68.63},{97.73,169.94},{96.29,170.47},{69.36,138.89},
  {76.03,140.16},{71.63,129.66},{14.73,73.72},{30.55,69.36},
  {90.98,153.28},{15.44,40.55},{94.05,155.37},{95.58,146.36},
  {30.50,50.53},{40.93,82.23},{53.19,108.18},{49.46,103.72},
  {84.84,146.27},{56.25,92.68},{18.60,69.02},{12.41,57.91},
  { 2.98,46.08},{25.75,63.79},{97.67,138.01},{82.61,133.12},
  {15.31,50.96},{68.86,122.15},{96.44,143.45},{ 5.69,64.11},
  {78.34,119.62},{53.71,99.43},{13.96,37.77},{27.91,61.52},
  {56.45,105.71},{87.10,147.15},{64.17,102.53},{50.38,90.39},
  {58.45,94.56},{97.31,134.92},{93.78,145.98},{79.53,128.34},
  {55.63,107.26},{56.69,90.85},{17.24,58.16},{94.33,138.14},
  {84.76,142.18},{46.67,100.15},{10.99,44.86},{27.50,55.39},
  { 0.45,37.66},{10.59,44.13},{16.45,52.52},{33.86,76.86},
  {94.86,138.84},{91.84,172.00},{42.33,93.16},{74.66,119.60},
  {33.78,87.28},{ 7.40,28.44},{24.19,82.22},{96.11,160.10},
  {74.05,125.18},{71.22,109.69},{97.56,140.56},{ 8.84,47.04},
  {69.56,117.00},{ 8.41,42.56},{34.89,64.19},{85.11,115.74},
  {37.56,58.13},{12.59,48.39},{ 2.68,46.73},{62.78,99.69},
  {67.09,109.26},{87.49,126.32},{79.12,136.70},{ 7.17,57.74},
  {26.72,50.76},{35.61,88.60},{27.05,73.51},{71.28,140.53},
  {32.32,78.74},{ 0.41, 2.12},{79.39,135.08},{57.97,100.72},
  {60.76,116.50},{93.08,128.87},{78.11,120.73},{85.78,136.17},
  {53.83,96.46},{26.24,48.05},{78.45,133.71},{97.53,140.60},
  {70.22,129.62},{68.04,119.83},{57.92,97.63},{59.04,86.68},
  {91.06,140.68},{32.87,67.78},{96.49,155.45},{23.99,63.38},
  {83.65,136.74},{73.11,125.63},{12.25,55.87},{79.42,121.84},
  {57.58,111.96},{20.97,68.56},{70.31,136.14},{28.32,78.99},
  {32.13,84.83},{43.86,94.95},{58.82,95.92},{ 2.14,49.54},
  {32.16,76.00},{76.61,126.87},{86.98,117.12},{66.97,108.44},
  {69.58,124.78},{ 7.97,40.98},{86.41,129.47},{17.79,38.59},
  {25.39,62.60},{50.76,90.81},{59.80,104.48},{38.07,70.55},
  {35.73,85.25},{17.46,51.93},{71.16,124.51},{71.80,115.02},
  { 1.62,16.54},{91.26,155.89},{ 3.06,50.64},{29.66,64.84},
  { 7.09,53.98},{49.85,109.93},{14.36,45.56},{24.68,53.75},
  {51.44,105.05},{25.25,65.94},{40.55,78.79},{26.87,72.33},
  {27.84,74.47},{40.26,85.74},{62.00,110.62},{10.52,40.68},
  {40.51,76.33},{ 4.88,52.94},{28.29,81.65},{27.36,83.38},
  {79.84,125.13},{26.59,50.55},{51.70,108.24},{49.35,90.31},
  {32.62,69.09},{15.94,45.41},{69.23,121.30},{ 7.46,45.89},
  {84.07,129.79},{90.25,124.15},{47.88,79.40},{96.61,163.10},
  {15.78,73.16},{80.96,131.56},{31.45,62.49},{ 9.29,50.43},
  {35.18,84.87},{30.50,76.96},{89.17,143.70},{90.85,146.92},
  {59.81,96.39},{25.35,70.99},{ 5.98,55.09},{37.00,74.37},
  {80.19,144.98},{94.28,135.99},{86.46,150.28},{34.03,75.79},
  { 4.75,34.83},{70.46,110.49},{94.34,167.80},{70.80,115.88},
  {16.07,57.87},{68.62,112.90},{95.65,146.35},{ 1.02,57.34},
  { 1.78,41.23},{21.44,60.15},{73.24,114.19},{44.80,89.07},
  {84.42,139.57},{98.01,138.74},{ 9.17,33.87},{87.40,133.66},
  {52.38,86.57},{61.45,112.93},{65.82,127.01},{83.91,158.51},
  {93.63,158.06},{61.78,120.56},{24.30,59.86},{28.54,80.27},
  {95.60,143.52},{61.49,117.24},{10.29,60.58},{11.44,50.16},
  {51.35,108.25},{ 7.62,50.81},{69.37,114.61},{50.44,97.38},
  {11.35,39.19},{17.17,43.10},{61.58,132.61},{28.84,69.42},
  {30.47,70.96},{98.22,151.09},{86.43,136.81},{53.19,118.09},
  {95.29,155.91},{98.84,148.83},{ 2.22,26.13},{47.59,85.31},
  {73.84,109.83},{17.56,47.04},{26.58,73.75},{56.82,105.65},
  {79.98,134.87},{68.84,112.36},{88.87,142.96},{62.43,119.22},
  {76.15,135.84},{43.70,103.87},{16.29,58.86},{53.38,96.60},
  {15.83,57.27},{51.27,120.72},{47.59,64.87},{71.91,129.93},
  {98.51,152.48},{71.10,109.44},{77.99,127.90},{23.87,62.56},
  { 8.34,51.65},{79.64,132.34},{31.04,72.40},{44.82,81.69},
  {59.45,109.71},{84.15,130.99},{82.20,127.12},{31.22,83.20},
  {86.18,141.01},{75.15,134.96},{44.27,95.34},{95.97,144.24},
  {14.43,49.55},{34.57,82.98},{63.67,104.58},{66.22,95.67},
  {88.43,134.55},{63.52,123.11},{77.81,136.16},{ 8.71,34.79},
  {29.07,60.32},{19.49,62.23},{ 9.98,58.80},{78.34,117.98},
  {38.96,76.73},{55.21,87.60},{ 5.69,48.63},{74.40,128.08},
  {61.88,107.77},{65.91,111.44},{78.58,122.76},{39.78,91.82},
  {74.49,116.43},{45.47,99.10},{45.39,95.83},{76.30,120.24},
  {10.79,42.01},{52.12,85.62},{62.14,114.29},{77.15,134.46},
  {78.85,143.47},{22.68,53.16},{83.14,126.70},{ 7.41,29.99},
  {82.26,132.70},{72.14,126.79},{99.55,147.42},{85.72,134.53},
  {64.30,139.77},{38.81,51.80},{40.39,87.73},{25.10,72.42},
  {25.08,66.21},{ 5.81,56.34},{14.18,41.56},{94.84,167.41},
  { 8.54,69.86},{63.85,112.79},{57.40,109.06},{21.58,42.54},
  {43.64,78.24},{77.65,141.23},{81.50,142.13},{67.89,111.36},
  {72.87,120.96},{27.00,62.02},{61.38,127.54},{83.46,136.99},
  {54.76,108.99},{15.45,47.88},{60.05,100.26},{22.10,72.44},
  {28.35,65.66},{96.71,151.35},{86.94,116.64},{65.73,130.87},
  {85.94,126.58},{38.88,96.56},{69.21,127.86},{43.69,61.97},
  {69.87,113.70},{82.25,150.00},{94.99,151.04},{25.88,83.24},
  {81.60,115.26},{73.98,117.58},{43.63,88.14},{70.31,100.08},
  {55.26,100.57},{25.59,71.33},{34.63,76.64},{ 2.50,41.37},
  {54.69,95.27},{78.95,129.89},{67.28,112.36},{89.06,136.95},
  {64.28,113.88},{85.81,145.44},{25.24,57.88},{80.48,135.92},
  {92.34,143.08},{46.56,90.96},{88.74,134.15},{53.17,104.03},
  {31.02,59.35},{ 5.52,52.83},{80.57,120.09},{98.37,169.48},
  {12.81,29.56},{21.78,49.52},{42.46,97.45},{75.97,122.99},
  {32.01,99.32},{15.05,60.83},{75.69,123.02},{36.09,68.64},
  {57.21,106.14},{91.60,134.65},{73.77,118.23},{ 0.03,39.93},
  {55.27,107.03},{52.33,89.48},{53.54,107.47},{49.96,74.23},
  {37.17,74.84},{84.24,130.66},{14.91,60.54},{65.04,130.72},
  { 9.44,41.64},{81.34,129.14},{71.28,100.66},{43.43,93.58},
  {75.01,113.17},{88.82,151.61},{29.93,71.33},{65.46,116.24},
  {22.97,61.19},{51.00,96.99},{60.59,109.94},{ 7.00,41.67},
  {25.77,71.27},{75.31,124.33},{ 4.36,40.75},{87.14,156.88},
  {17.37,61.50},{46.75,82.19},{13.05,42.65},{78.95,128.33},
  {73.10,134.15},{74.36,135.48},{98.65,146.97},{88.12,132.71},
  {28.60,82.30},{15.10,49.96},{49.63,97.09},{87.28,135.25},
  {61.89,111.58},{34.90,93.67},{49.99,100.83},{94.71,170.51},
  {25.88,73.00},{61.84,110.97},{55.36,102.57},{67.98,132.12},
  {21.37,64.84},{13.94,48.72},{43.71,90.80},{91.48,146.34},
  {31.14,59.33},{10.31,48.50},{94.61,149.76},{71.41,120.34},
  {21.14,64.98},{32.06,73.94},{66.20,105.95},{33.15,61.90},
  {86.19,131.68},{83.78,114.83},{54.33,96.58},{33.55,80.32},
  {96.60,170.21},{11.97,28.29},{63.59,124.56},{55.26,100.67},
  {63.14,123.76},{32.40,83.07},{18.12,63.35},{87.59,126.65},
  {50.10,97.98},{65.93,124.39},{62.49,103.95},{ 6.82,27.69},
  {46.80,85.68},{62.95,104.80},{45.74,78.09},{61.82,126.51},
  {92.80,134.13},{57.48,114.43},{74.40,109.61},{60.92,108.78},
  {30.89,79.66},{94.62,139.00},{49.83,98.73},{ 6.11,31.51},
  {45.54,88.00},{19.50,42.96},{25.52,66.51},{73.81,138.08},
  {29.34,88.98},{ 8.18,27.09},{30.62,65.44},{46.31,113.87},
  {41.02,86.84},{90.96,164.15},{22.01,45.41},{ 4.89,39.78},
  {95.35,142.65},{71.94,110.75},{63.35,105.06},{ 4.09,40.40},
  {88.54,139.61},{62.07,112.54},{27.70,56.09},{76.66,137.20},
  {83.12,143.20},{67.15,120.22},{60.00,99.28},{87.13,145.36},
  {94.71,148.17},{25.37,60.43},{64.45,125.47},{41.69,82.97},
  {39.60,86.26},{23.70,59.91},{95.30,155.36},{56.25,96.91},
  {10.09,46.74},{ 0.76,40.83},{59.30,94.67},{94.01,140.36},
  {88.44,141.46},{50.91,92.29},{42.26,99.49},{31.75,63.82},
  {48.52,104.01},{ 5.91,42.20},{80.60,128.78},{25.14,63.22},
  {74.28,124.28},{15.63,74.24},{97.86,149.97},{79.77,140.75},
  {65.69,118.02},{73.56,134.34},{17.85,53.07},{63.86,91.88},
  {13.97,53.39},{32.44,72.72},{ 8.17,56.60},{58.57,116.54},
  {37.35,65.08},{98.78,158.02},{70.98,130.54},{ 6.81,35.93},
  {85.39,160.17},{34.97,80.61},{ 2.64,43.13},{56.77,104.59},
  {81.91,112.09},{31.48,60.20},{34.81,61.30},{11.12,49.56},
  {71.51,128.10},{73.49,124.35},{72.99,112.97},{31.50,83.11},
  {99.98,147.74},{ 2.81,53.17},{42.82,98.70},{59.16,100.75},
  {23.89,72.61},{81.97,159.88},{46.85,94.22},{36.55,93.76},
  {64.96,95.23},{15.11,53.48},{65.91,113.75},{69.19,107.31},
  {28.06,63.30},{58.54,114.26},{89.15,153.48},{42.06,77.31},
  {40.50,76.81},{86.00,146.02},{ 3.20,48.79},{58.69,97.33},
  {35.28,78.08},{ 9.10,46.61},{25.91,66.15},{57.01,103.41},
  {14.91,73.97},{96.76,161.54},{99.67,149.40},{72.54,137.18},
  {30.50,74.61},{42.00,93.95},{59.93,109.89},{66.51,120.16},
  {80.06,138.60},{10.36,51.63},{ 1.60,27.94},{30.78,62.07},
  {66.55,102.64},{61.32,120.77},{91.03,150.60},{53.45,99.29},
  {42.37,78.63},{62.46,111.62},{66.04,112.54},{93.06,151.16},
  {51.95,88.51},{43.30,113.42},{64.13,99.00},{45.53,94.25},
  {39.47,87.77},{29.37,80.52},{92.61,162.12},{21.69,47.65},
  { 1.05,64.71},{40.01,92.37},{97.62,155.20},{70.10,106.73},
  {50.52,80.33},{11.96,51.24},{26.52,76.52},{77.52,132.84},
  {30.52,78.72},{30.52,78.19},{38.11,88.26},{76.86,128.87},
  {28.28,56.37},{30.49,65.71},{59.67,108.95},{89.64,157.43},
  {30.25,81.36},{22.29,69.28},{35.55,84.75},{68.06,113.95},
  {51.60,84.29},{10.09,43.39},{76.55,134.77},{71.33,115.11},
  {51.60,75.80},{54.79,109.48},{96.69,155.10},{14.77,49.52},
  {95.02,148.05},{96.72,141.18},{63.72,98.83},{91.93,140.47},
  {64.34,123.35},{88.86,137.02},{64.18,98.25},{70.04,110.77},
  {94.17,140.07},{75.24,124.76},{44.64,98.74},{87.41,153.95},
  {92.46,144.38},{19.06,66.13},{57.71,112.93},{ 7.45,39.86},
  {27.55,73.56},{56.19,108.21},{ 6.01,40.37},{62.74,89.47},
  {16.17,59.02},{ 8.79,30.74},{83.08,130.52},{24.23,54.52},
  {85.16,149.10},{24.81,90.03},{25.92,54.58},{54.33,82.03},
  {63.90,102.12},{68.66,100.77},{ 5.95,56.49},{42.26,93.76},
  {31.60,90.12},{44.62,89.67},{89.70,139.94},{24.77,52.46},
  {74.16,113.75},{85.36,142.27},{ 2.84,76.16},{91.59,150.81},
  {70.65,125.99},{26.70,76.22},{95.56,152.00},{ 0.44,27.70},
  {20.27,67.29},{ 5.23,37.86},{10.10,40.67},{74.38,130.22},
  {89.36,158.51},{ 4.21,42.73},{42.73,69.67},{72.56,111.50},
  {53.35,114.22},{28.76,96.68},{61.84,110.08},{16.56,62.79},
  {83.69,137.28},{48.91,86.74},{32.35,65.27},{29.44,75.42},
  {30.65,77.48},{85.56,144.14},{90.86,150.13},{81.44,126.15},
  {47.80,89.15},{13.73,41.35},{25.43,79.82},{58.07,92.59},
  {22.91,42.48},{49.14,87.71},{55.98,114.64},{ 8.82,45.15},
  {90.55,153.99},{81.55,138.12},{82.55,136.84},{51.00,101.17},
  {27.58,74.60},{37.31,77.27},{ 1.12,44.83},{88.58,143.95},
  {22.77,75.64},{97.08,157.64},{66.49,108.31},{98.86,156.70},
  {45.64,88.45},{89.75,139.02},{30.57,69.62},{36.48,85.01},
  {98.72,154.40},{30.12,76.32},{73.34,117.76},{16.37,52.48},
  {69.14,134.69},{98.21,174.31},{80.43,120.96},{56.01,100.03},
  { 1.48,43.90},{30.68,56.24},{65.36,121.74},{ 4.45,30.83}
};
```

Table of comparing running time of regression in original and multithread



	original	Multithread
1 time	0.185495735	0.594347646
2 time	0.263625642	0.575018792
3 time	0.186059904	0.768456743
4 time	0.198139012	0.608714755
5 time	0.210376657	0.579868326
6 time	0.200452651	1.471493032
7 time	0.230106994	0.702499545
8 time	0.189638428	0.863140864
9 time	0.233792324	0.583260398
10 time	0.224554748	0.585802211
Mean running time	0.21222421	0.733260231

Bar chart showing the variance of time between original and multithread of linear regression



From above we can clearly see that the mean time running of multithread is 0.733 and original is 0.212 which shows the original is faster than the multithread. What happened here is in the multithread there is two thread which are doing work simultaneously and the output need to merge again. During which it consumes fraction of time than the original so in this case the multithread takes more time.

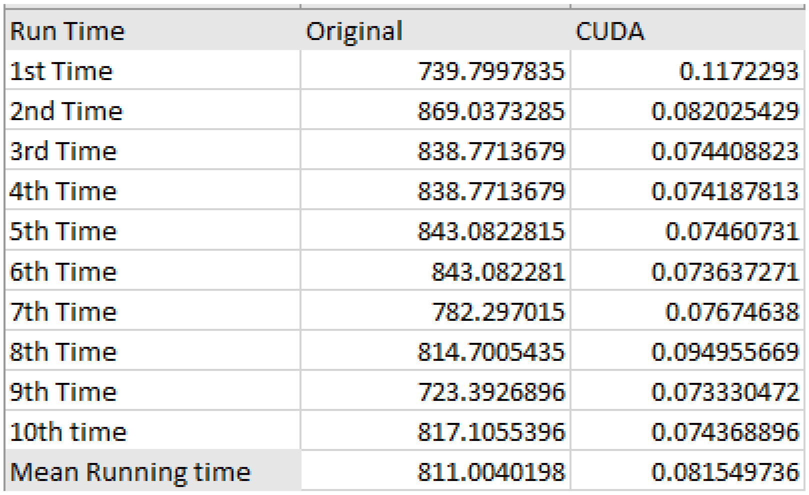
2 CUDA

2.1 Password Cracking

```
#include <stdio.h>
#include <cuda_runtime_api.h>
#include <time.h>
/**************************************************************
**************
  To Compile:
    nvcc -o Question_3_1_a Question_3_1_a.cu

  To run:
    ./Question_3_1_a

***************************************************************
**************/

__device__ int is_a_match(char *attempt) {
  char plain_password_1[] = "SA1234";
  char plain_password_2[] = "BI5678";
  char plain_password_3[] = "NG1246";
  char plain_password_4[] = "TI3579";


  char *a = attempt;
  char *b = attempt;
  char *c = attempt;
  char *d = attempt;
  char *p1 = plain_password_1;
  char *p2 = plain_password_2;
  char *p3 = plain_password_3;
  char *p4 = plain_password_4;

  while(*a == *p1) {
   if(*a == '\0')
    {
     printf("Password found is: %s\n",plain_password_1);
      break;
    }

    a++;
    p1++;
  }

  while(*b == *p2) {
   if(*b == '\0')
```

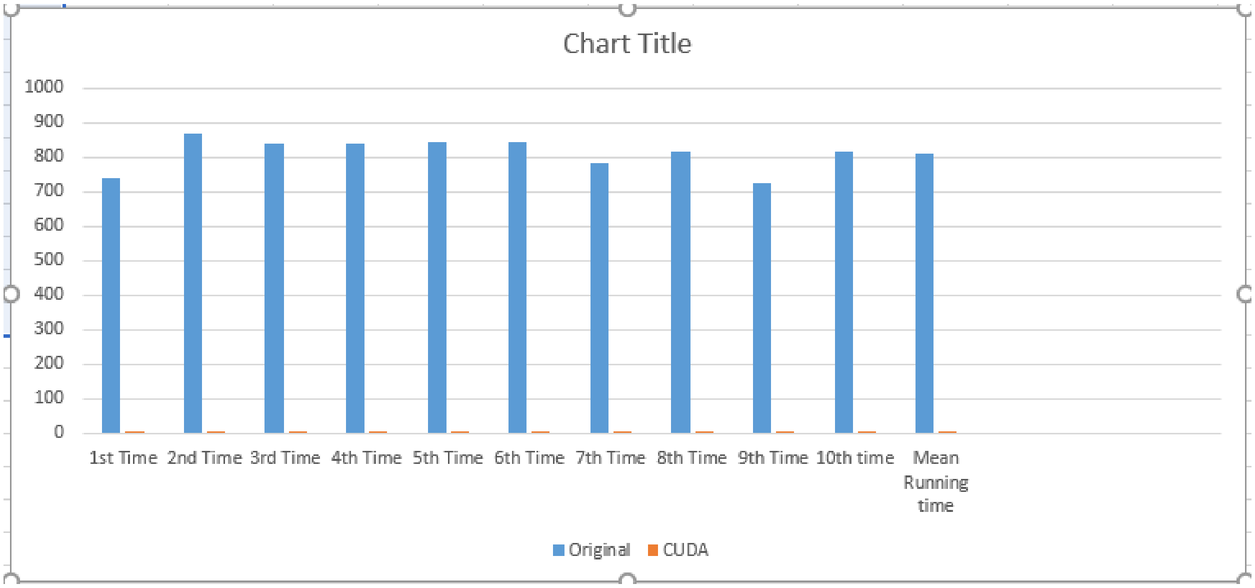
```
    {
     printf("Password found is: %s\n",plain_password_2);
      break;
    }

    b++;
    p2++;
  }

  while(*c == *p3) {
   if(*c == '\0')
    {
     printf("Password found is: %s\n",plain_password_3);
      break;
    }

    c++;
    p3++;
  }

  while(*d == *p4) {
   if(*d == '\0')
    {
     printf("Password found is: %s\n",plain_password_4);
      return 1;
    }

    d++;
    p4++;
  }
  return 0;
}
__global__ void  kernel() {
char w,x,y,z;

  char password[7];
  password[6] = '\0';

int i = blockIdx.x+65;
int j = threadIdx.x+65;
char firstValue = i;
char secondValue = j;

password[0] = firstValue;
password[1] = secondValue;
     for(w='0'; w<='9'; w++){
       for(x='0'; x<='9'; x++){
        for(y='0'; y<='9'; y++){
          for(z='0'; z<='9'; z++){
             password[2] = w;
             password[3] = x;
             password[4] = y;
             password[5] = z;
           if(is_a_match(password)) {

           }
             else {

           }
        }
     }
     }
     }

}

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;
  clock_gettime(CLOCK_MONOTONIC, &start);

  kernel <<<26,26>>>();
  cudaThreadSynchronize();

  clock_gettime(CLOCK_MONOTONIC, &finish);
  time_difference(&start, &finish, &time_elapsed);
  printf("Time elapsed was %lldns or %0.9lfs\n", time_elapsed,
(time_elapsed/1.0e9));

  return 0;
}
```

Table showing the running times for the original and CUDA version



Run Time	Original	CUDA
1st Time	739.7997835	0.1172293
2nd Time	869.0373285	0.082025429
3rd Time	838.7713679	0.074408823
4th Time	838.7713679	0.074187813
5th Time	843.0822815	0.07460731
6th Time	843.082281	0.073637271
7th Time	782.297015	0.07674638
8th Time	814.7005435	0.094955669
9th Time	723.3926896	0.073330472
10th time	817.1055396	0.074368896
Mean Running time	811.0040198	0.081549736

Graph showing running time of original and CUDA version for running password cracking



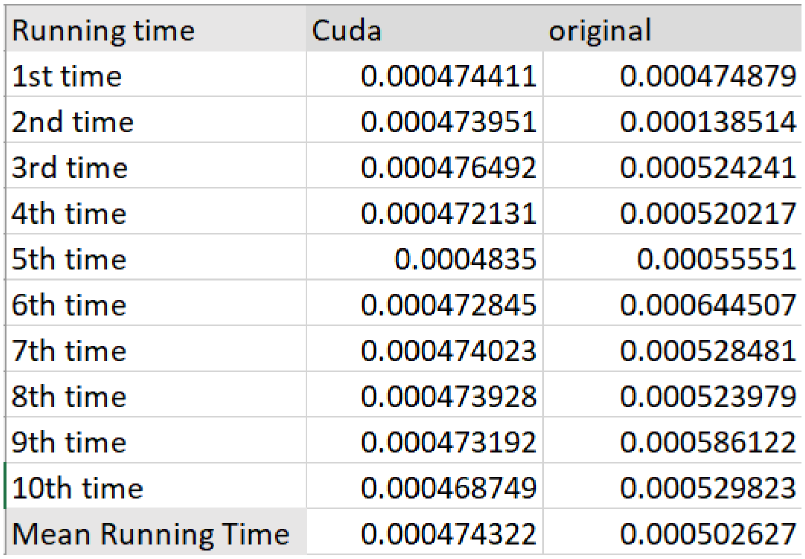
Form the above data we can clearly see the time taken by the CUDA to crack the password is way faster than original. In the original cracking we have 1 thread in 1 block. But in CUDA we have 26 blocks and in each in 1 block there is 26 threads. Hence there are total 676 threads to process a task. By the CUDA programming it takes nearly 1 sec to crack password. Our CUDA program takes 0.081549736s which is nearly to 1 sec.

2.2 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 <cuda_runtime_api.h>



/*************************************************************
****************

  To compile:
   nvcc -o imagecuda imagecuda_033.cu -lglut -lGL -lm

  Dr Kevan Buckley, University of Wolverhampton, 2018
*************************************************************
***************/
#define width 100
#define height 72

unsigned char image[] =
{255,255,255,255,0,0,255,255,255,255,255,255,255,0,255,255,0,0,

255,0,0,255,255,255,255,0,0,0,0,255,255,255,255,255,0,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,0,0,0,0,255,255,255,255
,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,0,255,255,255,255,255,0,0,255,255,255,255,255,2
55,255,
  0,0,255,0,0,255,255,0,255,255,255,255,255,0,0,0,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,0,255,255,0,0,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,0,0,255,255,255,255,255,0,2
55,255,
  255,255,0,0,255,255,0,0,0,0,255,255,0,0,255,255,255,255,255,

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
,0,0,255,

255,255,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,2
55,255,
  255,255,0,0,0,0,0,0,0,255,255,0,0,0,0,255,255,255,0,

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
,0,255,255,

255,255,0,0,255,255,255,0,0,255,255,0,0,0,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,0,0,255,255,255,255,255,0,0,0,255,255,255,255,255,255,0,0,

0,255,255,255,0,0,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,0,
  255,255,0,0,255,255,255,0,0,255,255,255,255,0,0,0,0,0,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,0,0,0,255,255,255,255,0,0,255,255,255,2
55,

255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,0,0,255,0,0,255,255,255,0,0,255,255,255,255,0,

0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,

255,255,255,255,255,255,255,255,255,255,255,0,255,0,255,255,255
,255,255,

0,255,255,255,255,0,0,255,255,0,255,255,255,255,255,255,255,255
,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,0,0,255,255,255,255,0,0,0,255,255,0,0,0,0,0,0,

0,0,255,255,255,0,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,0,
  0,255,255,255,255,0,0,0,0,0,0,0,255,255,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,255,255,0,
  0,0,0,255,255,255,0,0,255,255,0,0,255,255,0,0,0,0,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,0,0,255,255,255,255,0,0,0,0,255,255,255,255,255
,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,0,255,255,255,255,255,255,255,255,255,255,0,0,0,255
,255,
  0,0,255,255,255,0,0,255,255,255,255,0,0,255,255,0,0,0,0,

0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,

255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,0,0,0,0,0,0,255,255,255,255,255,255,255,255
,
  255,0,0,255,255,0,0,255,255,255,0,0,255,255,255,255,0,0,255,

255,255,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,255
,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,0,0,255,255,255,0,0,255
,255,

255,255,255,255,255,255,255,0,0,255,0,0,255,255,255,255,0,0,255
,

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
,255,255,0,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,0,0,0,255,255,255,0,0,255,255,2
55,
255,255,0,255,255,255,255,255,255,255,255,255,255,0,0,0,0,255,2
55,

255,255,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,255,

255,0,0,255,255,255,255,0,0,255,255,255,255,255,255,255,255,255
,0,

0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

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,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0
,0,255,

255,255,0,0,0,255,0,0,255,255,255,255,255,0,255,255,255,255,255
,

255,255,255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,2
55,255,255,

255,0,0,255,255,255,255,255,0,0,255,0,0,255,255,255,255,255,0,
0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,0,0,0,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,0,255,255,255,255,0,0,255,255,255,255,255,255,0,0,255,0,0,

255,255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,0,0,255,255,0,0,255,255,255,255
,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,0,0,255,255,255,0,0,255,255,255,255,255
,255,

0,0,255,255,0,255,255,255,255,255,0,0,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,255,255
,255,0,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,0,255,255,255,2
55,0,0,

255,255,255,255,255,255,0,255,255,0,0,255,255,255,255,255,0,255
,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,0,255,0,0,255,255,255,2
55,0,

0,255,255,255,0,0,255,0,0,0,0,255,255,255,255,255,255,255,255,

255,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,0,0,

255,255,255,0,0,255,255,255,255,255,0,0,255,255,0,0,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,255,255,0
,
  0,255,255,255,0,0,255,255,255,0,0,0,0,0,255,255,255,255,255,

255,255,255,255,255,0,0,0,255,255,0,0,255,255,255,255,255,255,2
55,
  255,255,255,0,0,0,0,255,255,0,0,255,255,255,255,0,0,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0
,255,

255,255,255,255,0,0,255,255,0,0,0,255,255,255,255,0,0,255,255,

255,255,255,255,255,255,0,255,255,255,0,0,255,255,255,255,0,0,2
55,
  255,255,255,255,255,255,255,255,0,0,0,0,0,255,255,0,0,0,0,

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
,0,0,255,
  255,0,0,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,255,

255,0,0,255,255,255,255,255,255,255,255,0,0,0,255,0,0,255,255,
255,255,0,0,255,255,255,255,255,255,255,255,255,0,255,255,255,0
,0,

255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,0,0,255,255,0,0,255,255,255,255,255,255,255,0,0,0,255,
  255,255,255,0,0,255,255,0,0,255,0,0,0,0,255,255,255,0,0,

255,255,255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,2
55,0,

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,
  255,255,0,255,255,255,255,255,0,0,255,255,0,0,255,0,0,0,0,
  255,255,0,0,255,255,255,255,0,0,255,255,0,0,0,0,0,255,255,

255,255,255,255,0,0,255,255,255,0,0,0,255,255,255,255,255,255,2
55,

255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,0,0,255,255,255,255,0,0,255,255,0,0
,

0,0,0,255,255,255,255,0,0,255,255,255,255,0,0,255,255,255,255,

255,255,255,255,255,255,255,255,0,0,255,255,255,0,0,0,255,255,2
55,

255,255,255,255,255,255,255,255,255,0,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
255,255,0,0,0,255,255,255,255,255,255,255,0,0,255,255,255,255,0
,
  0,0,255,255,0,0,255,255,255,255,255,255,255,0,0,0,0,0,0,

255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,255,255,2
55,0,

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,

255,255,255,255,255,0,0,0,0,0,0,255,255,255,255,255,255,255,0,
  0,0,0,0,0,0,0,255,255,0,0,255,255,255,255,255,255,255,0,

0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0
,0,

255,255,255,255,255,0,0,255,255,0,0,255,255,255,255,255,255,255
,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,0,0,255,255,255,255,0,0,255
,255,
  255,255,255,255,0,0,0,0,0,255,255,0,0,255,255,0,0,255,0,

0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,0,255,255,255,0,0,0,0,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,0,0,255,255,255,0,0,255,255
,255,

255,0,0,255,255,255,255,255,255,255,0,0,255,255,255,255,0,0,255
,

255,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,2
55,
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,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,2
55,255,

0,0,255,255,255,0,0,0,255,255,255,255,255,255,255,0,0,255,255,

255,255,255,0,0,255,255,0,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,0,255,255,0,0,255,2
55,255,

0,0,0,255,255,0,0,255,255,255,255,255,255,255,255,255,255,255,2
55,

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
,255,255,0,

0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,255,255,0
,0,255,

255,0,255,255,255,0,0,0,0,255,0,0,255,255,255,255,255,255,255,

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
,255,0,0,
  255,255,0,0,255,255,0,0,255,255,0,0,255,0,255,255,0,0,255,

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
,255,0,255,

255,255,255,0,0,255,255,255,0,0,255,255,0,255,255,0,255,255,0,

0,255,0,0,255,255,255,255,255,0,0,255,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  0,255,255,0,0,255,255,255,0,0,255,255,255,0,0,255,255,0,0,
  0,0,255,255,255,0,255,255,0,0,0,0,0,0,0,255,255,255,255,

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,

255,255,255,255,0,0,255,255,0,0,255,255,255,0,0,255,255,255,255
,
  0,0,255,0,0,0,0,255,255,255,0,0,255,255,255,0,0,0,0,

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,

255,255,255,255,0,255,255,255,255,0,0,255,255,255,0,0,0,255,0,

0,255,255,255,255,0,0,255,255,0,0,0,255,255,255,255,0,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,255
,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,0,255,255,0,0,255,255,255,0,0,255,255,2
55,
  0,0,0,0,0,0,255,255,255,255,255,0,0,255,0,0,0,255,255,

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,

0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,

255,255,255,255,255,255,255,255,255,255,255,0,255,255,0,0,255,2
55,255,
  0,0,255,255,255,255,0,0,0,0,0,0,255,255,255,255,0,0,255,

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
,255,255,0,
0,255,255,255,255,0,0,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,0,0,255,255,255,255,0,0,0,2
55,
  255,0,0,0,0,0,0,0,0,255,255,0,0,255,255,0,0,0,0,

255,255,255,0,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,0,0,255,255,0,0,255,255,255,0,0,255,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0
,0,0,
  0,0,0,0,255,255,0,0,0,0,255,255,0,0,0,255,255,0,0,

255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,0,255,255,255,0,0,0,255,255,0,0,0,255,255,0,0,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,0,0,255,255,255
,255,255,
  255,0,0,0,0,0,255,0,0,255,255,255,0,0,255,255,255,255,0,

0,255,255,0,0,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,0,0,255,0,0,255,255,255,0,0,0,0,255,255,0,0,0,

0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,0,
  0,255,255,255,255,255,255,0,0,0,255,255,0,0,255,255,255,0,0,

255,255,255,255,0,0,255,255,255,0,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,0,255,255,0,0,255,255,0,0,255,255,0,0,0,0,

255,255,0,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255
,

255,255,255,255,0,0,255,255,255,255,255,255,255,0,0,255,255,0,0
,

255,255,255,255,0,0,255,0,0,0,0,255,255,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,0,0,255,255,0,0,255,0,0,255
,
  255,0,0,255,0,0,255,255,0,0,255,0,0,0,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,255,255,
  0,0,255,0,0,255,255,255,255,0,0,0,0,0,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,0,0,255,255,255,255,255,0,255,2
55,0,
  0,255,255,0,0,255,0,0,255,255,0,255,255,0,0,255,255,255,0,

0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,2
55,

255,255,0,0,255,255,0,0,0,0,255,255,255,255,255,0,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,255,2
55,255,255,
  255,0,0,255,255,0,0,255,255,0,0,0,0,255,255,0,0,255,255,

0,0,255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,2
55,255,
255,255,0,0,255,255,255,255,0,255,255,255,0,0,0,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,255,255
,255,255,
  0,0,255,255,255,255,0,0,255,255,0,0,255,255,0,0,0,0,255,

255,255,0,255,255,0,0,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,0,0,255,255,255,0,0,0,255,255,255,0
,

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
,255,255,0,
  0,255,255,255,255,255,0,0,255,0,0,0,0,0,255,255,0,0,255,

0,0,0,0,255,255,255,0,0,255,255,255,255,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,255,0,0
,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,
  255,255,255,255,255,0,0,255,255,255,255,0,0,0,0,0,255,0,0,

255,255,0,0,255,255,0,0,0,255,255,255,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,0,

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
,255,0,0,

0,255,255,255,255,255,255,255,255,255,0,0,255,255,255,255,255,0
,0,

255,255,255,255,0,0,255,255,0,0,255,0,0,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,0,0,2
55,

255,255,255,0,0,255,255,255,255,0,0,255,255,0,0,255,255,255,255
,

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
,255,255,0,
  0,0,255,0,0,255,255,255,0,0,0,0,255,255,255,255,255,255,255,

255,255,0,0,255,255,255,255,0,0,255,255,255,255,255,0,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,0,0,0,0,0,255,255,255,0,0,255,255,0,0,0,0,255,255,
255,255,255,255,255,255,255,255,0,0,255,255,255,255,0,0,255,255
,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,

255,255,255,255,255,0,0,0,255,255,255,255,255,255,0,0,255,255,0
,
  0,255,0,0,255,255,255,255,255,255,255,255,255,0,0,0,0,0,255,

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,

255,255,255,255,255,0,255,255,255,255,255,0,0,255,255,255,255,2
55,255,

255,0,0,255,0,0,255,255,0,255,255,255,255,255,255,255,255,0,0,

0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,0,0,0,255,255,255,255,0,0,2
55,

255,255,255,255,255,255,0,0,0,0,0,255,255,0,0,255,255,255,255,

255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,0,0,0,0,0,0,0,255,255,255,255,255,255,255,255,255,

255,255,255,255,255,255,255,255,0,0,255,255,255,255,0,0,0,0,255
,
  255,255,255,0,0,0,0,0,0,0,255,255,0,0,0,0,255,255,0,

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,255,255,255,2
55,
  0,0,0,0,0,255,255,255,0,0,0,0,0,255,255,255,255,0,0,

0,255,255,255,255,0,0,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,0,

0,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,255
,255,

255,255,255,255,255,0,0,255,255,0,0,255,255,255,0,0,255,255,255
,

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,
  255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255
,

0,0,255,255,255,255,255,255,255,255,255,0,255,255,255,0,0,255,2
55,
0,0,255,255,255,255,0,0,255,0,0,255,255,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,255,255,255,255,255,255,255,255,

255,255,255,255,255,255,0,255,255,255,255,255,255,255,255,255,0
,0,255,
  255,255,0,0,255,255,0,0,0,0,0,0,0,255,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,255,255,255,

0,255,255,255,255,255,255,255,255,255,255,0,0,255,0,0,0,0,255,
  255,255,0,0,255,255,255,0,0,0,255,0,0,0,0,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0
,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,255,255,255,0,

0,0,255,255,255,0,0,255,255,255,255,255,255,255,255,255,0,0,0,

0,0,255,255,255,255,255,0,0,255,255,255,255,0,0,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0
,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

255,255,255,255,0,0,0,255,255,255,255,0,255,255,255,255,255,255
,255,

255,255,255,0,0,255,255,255,255,255,255,255,0,0,255,255,255,255
,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

255,255,0,0,0,255,255,255,255,255,0,255,255,255,255,255,0,0,255
,

255,255,255,255,255,255,255,255,0,0,255,255,255,255,255,0,255,0
,0,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,255,255,255,0,0,0,255,255,255,255,255,0,255,255,255,

255,255,255,0,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0
,

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,
  255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,255,255,255,0,0,255,255,255,0,0,0,255,255,255,255,

255,0,255,255,255,255,255,255,0,0,255,255,255,255,255,255,255,2
55,255,

0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,255,255,255,0,255,255,255,255,0,255,255,255,255,0,

0,255,255,255,255,255,255,0,255,255,255,255,255,255,0,255,255,2
55,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,255,255,255,255,0,255,255,255,255,0,

255,255,255,255,0,255,255,255,255,255,255,255,0,255,255,255,255
,255,255,

0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,255,255,0,0,0,255,255,255,255,0,0,

255,255,255,255,0,0,255,255,255,0,255,255,255,255,255,255,255,0
,255,

0,255,255,255,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,255,255,255,0,0,0,255,

255,255,255,0,0,0,255,255,255,0,0,255,255,255,255,255,255,255,2
55,

255,255,255,0,255,0,0,255,0,0,0,255,255,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,255,255,255,0,0,255,255,

255,255,0,0,255,255,255,255,0,0,0,255,255,255,255,0,255,255,255
,

255,255,255,255,0,255,255,255,255,255,0,0,255,255,255,255,255,2
55,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,255,255,

255,0,0,255,255,255,255,0,0,255,255,255,255,0,0,0,255,255,255,

255,0,0,255,255,255,255,255,255,0,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,255,255,255,255,

0,0,255,255,255,255,0,0,255,255,255,255,255,255,255,255,255,255
,0,

0,0,255,255,255,255,0,0,255,255,255,255,255,255,0,0,255,255,255
};


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);
}

unsigned char results[width * height];
//static void key_pressed(unsigned char key, int x, int y);
//void stgint callback(int signal_number);
//static void display();
//void tidy_and_exit();

__global__ void detect_edges(unsigned char *in, unsigned char
*out) {

  unsigned int i = blockIdx.x ;
```

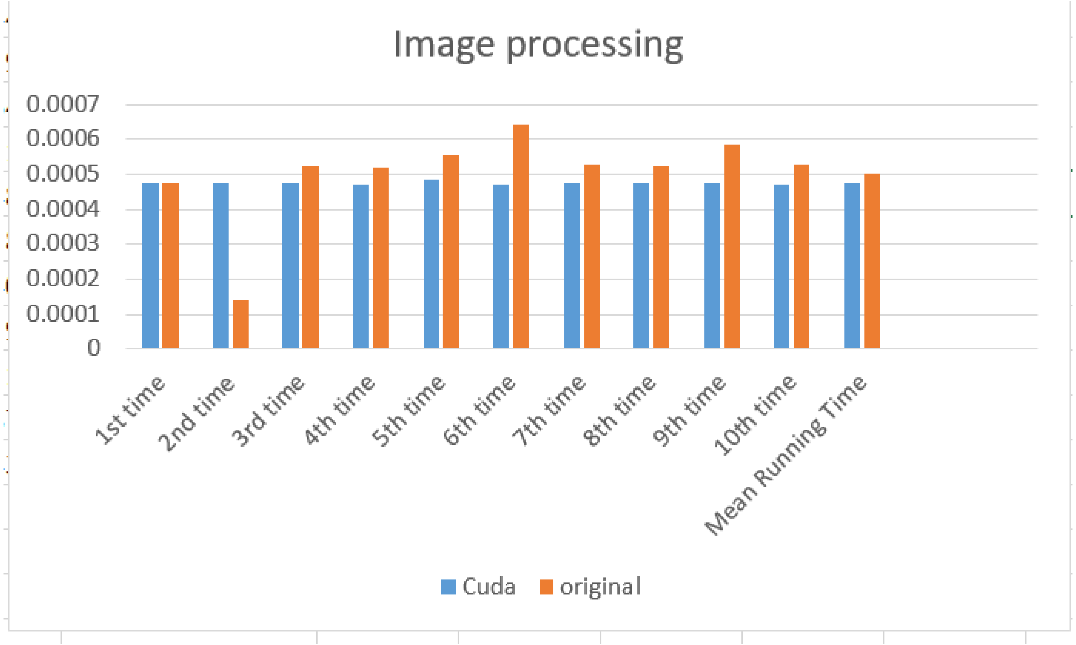
```
    int x;
    int y;
    int b;
    int d;
    int f;
    int h;
    int r;

    y = i / 100;
    x = i - (100 * y);

    if (x == 0 || y == 0 || x == width - 1 || y == height - 1)
{
      out[i] = 0;
    } else {
      b = i + 100;
      d = i - 1;
      f = i + 1;
      h = i - 100;

      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;
      }
    }
  }




void tidy_and_exit() {
  exit(0);
}

void sigint_callback(int signal_number){
  printf("\nInterrupt from keyboard\n");
  tidy_and_exit();
}

static void display() {
  glClear(GL_COLOR_BUFFER_BIT);
  glRasterPos4i(-1, -1, 0, 1);
  glDrawPixels(width, height, GL_LUMINANCE, GL_UNSIGNED_BYTE,
image);
  glRasterPos4i(0, -1, 0, 1);
  glDrawPixels(width, height, GL_LUMINANCE, GL_UNSIGNED_BYTE,
results);
  glFlush();
}

static void key_pressed(unsigned char key, int x, int y) {
  switch(key){
    case 27: // escape
      tidy_and_exit();
      break;
    default:
      printf("\nPress escape to exit\n");
      break;
  }
}


int main(int argc, char **argv) {

  signal(SIGINT, sigint_callback);

  printf("image dimensions %dx%d\n", width, height);


  unsigned char *d_results;
  unsigned char *d_image;

  cudaMalloc((void**)&d_results, sizeof(unsigned char) * (width
* height));
  cudaMalloc((void**)&d_image, sizeof(unsigned char) * (width *
height) );
  cudaMemcpy(d_image, &image, sizeof(unsigned char) * (width *
height), cudaMemcpyHostToDevice);
  cudaMemcpy(&d_results, &results, sizeof(unsigned char) *
(width * height), cudaMemcpyHostToDevice);

  struct timespec start, finish;
  long long int time_elapsed;
  clock_gettime(CLOCK_MONOTONIC, &start);

  detect_edges <<<7200, 1>>>(d_image, d_results);
  cudaThreadSynchronize();
  cudaMemcpy(&results, d_results, sizeof(unsigned char) *
(width * height), cudaMemcpyDeviceToHost);
  cudaMemcpy(&image, &d_image, sizeof(unsigned char) * (width *
height), cudaMemcpyDeviceToHost);

  cudaFree(&d_image);
  cudaFree(&d_results);

  clock_gettime(CLOCK_MONOTONIC, &finish);
  time_difference(&start, &finish, &time_elapsed);
  printf("Time elapsed was %lldns or %0.9lfs\n", time_elapsed,
(time_elapsed/1.0e9));

  glutInit(&argc, argv);
  glutInitWindowSize(width * 2,height);
  glutInitDisplayMode(GLUT_SINGLE | GLUT_LUMINANCE);

  glutCreateWindow("6CS005 Image Progessing Courework");
  glutDisplayFunc(display);
  glutKeyboardFunc(key_pressed);
  glClearColor(0.0, 1.0, 0.0, 1.0);

  glutMainLoop();

  tidy_and_exit();

  return 0;
}
```

Table showing the running time of original and CUDA version of image processing.



Running time	Cuda	original
1st time	0.000474411	0.000474879
2nd time	0.000473951	0.000138514
3rd time	0.000476492	0.000524241
4th time	0.000472131	0.000520217
5th time	0.0004835	0.00055551
6th time	0.000472845	0.000644507
7th time	0.000474023	0.000528481
8th time	0.000473928	0.000523979
9th time	0.000473192	0.000586122
10th time	0.000468749	0.000529823
Mean Running Time	0.000474322	0.000502627

Bar chart showing time difference of original and CUDA



In our original program there is no threads the program does the image processing in one single process. But in CUDA there are numerous threads to divide the work. As, our image is small the pixels are divided into different threads to process which takes time. After the work is done again then again merging the pixels and showing the output is time consuming that's why CUDA takes longer time to execute than original.

2.3 Linear Regression

```
#include <stdio.h>
#include <math.h>
#include <time.h>
#include <unistd.h>
#include <cuda_runtime_api.h>
#include <unistd.h>


/***************************************************************
****************
 * This program takes an initial estimate of m and c and finds
the associated
 * rms error. It is then as a base to generate and evaluate 8
new estimates,
 * which are steps in different directions in m-c space. The
best estimate is
 * then used as the base for another iteration of "generate and
evaluate". This
 * continues until none of the new estimates are better than
the base. This is
 * a gradient search for a minimum in mc-space.
 *
 * To compile:
 *   nvcc -o lr_coursework_002_cuda lr_coursework_002_cuda.cu -
lm
 *
 * To run:
 *   ./lr_coursework_002_cuda
 *
 * Dr Kevan Buckley, University of Wolverhampton, 2018

****************************************************************
**************/


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

int n_data = 1000;
__device__ int d_n_data = 1000;

point_t data[] = {
  {77.96,128.03},{65.68,95.52},{82.85,133.97},{87.44,126.26},
  {65.25,115.66},{65.81,109.29},{78.07,137.65},{73.81,119.15},
  {83.67,130.20},{83.29,130.15},{65.45,92.82},{14.28,69.92},
  {59.72,113.57},{51.21,105.39},{18.33,70.24},{17.57,42.36},
  {19.28,86.46},{61.11,110.70},{ 3.06,32.92},{66.61,132.11},
  {50.28,86.85},{16.14,56.57},{71.80,131.85},{76.53,143.87},
  {22.16,49.15},{30.64,87.42},{58.97,98.66},{10.46,46.54},
  {65.53,106.21},{91.93,143.24},{73.47,128.02},{20.95,64.18},
  {62.50,110.38},{18.18,54.18},{21.48,57.86},{28.50,60.11},
  { 7.54,48.39},{27.41,78.05},{49.86,91.89},{27.28,85.08},
  {53.79,104.68},{21.85,57.80},{35.55,66.67},{95.93,155.19},
  {57.39,104.01},{42.54,96.87},{67.66,103.79},{82.65,134.83},
  {56.97,100.17},{64.07,112.48},{87.60,146.67},{36.95,90.02},
  {24.78,62.26},{65.78,106.87},{72.22,123.40},{85.91,138.57},
  {22.21,58.65},{23.45,65.71},{34.59,66.36},{40.13,82.01},
  {19.99,73.66},{47.56,101.54},{ 8.38,26.03},{61.23,96.47},
  {52.33,115.23},{61.95,116.68},{84.06,132.97},{47.14,96.40},
  {10.24,56.26},{42.03,87.61},{12.97,52.18},{82.86,150.30},
  {30.40,76.91},{66.49,114.01},{42.05,78.38},{68.17,120.08},
  {91.94,142.49},{66.60,97.85},{38.59,106.14},{67.52,114.10},
  { 0.68,39.30},{ 5.86,47.99},{87.24,138.51},{40.64,79.63},
  {85.67,145.03},{ 1.15,34.90},{57.79,100.48},{28.04,48.02},
  {79.74,149.06},{39.70,84.37},{47.29,113.66},{70.42,141.72},
  {17.33,66.95},{79.96,142.75},{38.66,63.02},{34.16,64.26},
  {82.46,122.55},{39.01,78.63},{36.46,78.62},{28.80,66.12},
  {39.23,75.54},{63.28,97.79},{20.99,74.67},{94.94,136.48},
  {65.50,132.78},{60.50,108.85},{36.39,90.05},{15.92,58.69},
  {56.87,106.49},{79.49,133.03},{65.81,119.26},{61.67,111.30},
  {13.16,62.03},{28.13,76.33},{33.71,71.92},{16.93,67.53},
  { 8.77,48.89},{ 3.03,25.86},{24.15,64.71},{78.16,138.89},
  {66.89,93.13},{70.14,133.04},{69.61,115.60},{60.69,92.18},
  {24.13,40.17},{ 0.62,44.00},{21.36,73.46},{79.55,141.25},
  {37.98,93.03},{ 9.27,51.73},{51.18,88.24},{24.94,78.03},
  {40.98,76.92},{55.01,103.17},{66.86,133.25},{ 6.21,39.08},
  {95.42,145.04},{91.66,159.55},{74.85,129.96},{33.35,82.30},
  {99.98,153.62},{ 7.78,57.71},{43.91,106.75},{85.56,141.85},
  {99.30,154.24},{76.92,135.95},{23.31,71.94},{83.06,124.53},
  {20.73,48.34},{ 7.61,64.85},{ 4.14,62.97},{48.41,93.44},
  {11.08,54.65},{23.22,66.51},{86.51,146.66},{63.93,96.90},
  {62.18,140.01},{54.58,92.40},{14.68,55.65},{74.08,152.67},
  {29.16,78.88},{34.39,70.74},{53.47,111.77},{79.47,115.22},
  {60.26,115.73},{ 4.35,53.66},{44.74,98.91},{36.52,77.68},
  {12.57,46.94},{ 9.35,29.42},{67.14,132.38},{94.48,155.46},
  {60.56,126.66},{82.58,148.39},{40.20,81.60},{97.03,152.57},
  {37.79,88.69},{92.35,131.07},{73.56,141.49},{60.68,89.01},
  {50.87,102.46},{80.10,134.53},{20.10,63.39},{56.11,85.56},
  {17.12,57.21},{43.41,79.14},{66.00,99.99},{55.44,104.05},
  {65.87,112.98},{87.30,140.25},{40.73,84.93},{35.28,76.61},
  {93.55,139.77},{51.48,73.31},{10.10,57.74},{65.11,111.70},
  {16.16,44.28},{62.35,90.61},{ 3.11,21.44},{97.40,152.47},
  {31.24,68.96},{20.00,58.73},{83.50,123.60},{72.10,109.50},
  {18.70,39.72},{80.72,123.39},{71.65,122.57},{47.08,90.29},
  {15.62,61.31},{39.27,88.61},{73.84,133.29},{48.34,93.10},
  {71.71,131.19},{ 4.44,49.00},{88.42,150.90},{34.22,92.45},
  {36.37,63.52},{22.69,60.34},{81.39,132.60},{61.47,104.34},
  {24.19,69.92},{28.64,65.00},{83.10,134.44},{55.87,97.39},
  {91.48,164.21},{86.12,158.13},{18.45,61.30},{70.00,115.55},
  {29.97,70.25},{ 7.80,50.03},{93.35,152.67},{98.77,166.10},
  {76.52,123.74},{76.27,140.22},{79.94,142.14},{ 9.42,26.12},
  {25.14,66.01},{42.29,84.73},{ 7.16,34.35},{34.35,79.77},
  {30.24,60.18},{97.12,158.57},{ 3.77,32.31},{18.59,71.26},
  {23.68,65.65},{77.69,134.56},{74.78,132.51},{38.54,79.86},
  { 1.41,29.22},{34.20,81.76},{46.56,83.08},{21.55,54.54},
  {36.12,80.97},{89.12,143.03},{74.69,139.63},{72.38,134.98},
  {90.52,126.38},{33.45,77.65},{14.45,48.11},{ 2.19,69.22},
  {34.24,68.57},{38.58,69.10},{31.53,76.53},{95.11,149.75},
  {82.77,134.49},{17.54,48.12},{27.36,80.42},{74.70,120.86},
  {18.22,61.57},{23.51,62.04},{43.08,64.64},{37.54,75.77},
  {40.53,96.29},{63.39,122.09},{57.60,116.76},{70.50,133.52},
  {51.09,93.63},{19.81,58.37},{16.62,33.92},{40.21,71.13},
  {83.14,140.67},{34.51,67.66},{51.98,88.01},{57.57,125.27},
  { 7.08,39.25},{41.83,82.22},{95.35,152.79},{27.18,68.47},
  {84.15,140.31},{28.04,72.94},{97.86,152.05},{76.82,124.28},
  {95.52,159.59},{14.04,60.42},{ 6.68,25.03},{46.02,85.90},
  {25.07,67.74},{12.15,59.67},{79.23,130.51},{67.33,124.69},
  {19.51,60.84},{65.77,111.26},{51.20,100.38},{79.65,128.45},
  {41.88,68.63},{97.73,169.94},{96.29,170.47},{69.36,138.89},
  {76.03,140.16},{71.63,129.66},{14.73,73.72},{30.55,69.36},
  {90.98,153.28},{15.44,40.55},{94.05,155.37},{95.58,146.36},
  {30.50,50.53},{40.93,82.23},{53.19,108.18},{49.46,103.72},
  {84.84,146.27},{56.25,92.68},{18.60,69.02},{12.41,57.91},
  { 2.98,46.08},{25.75,63.79},{97.67,138.01},{82.61,133.12},
  {15.31,50.96},{68.86,122.15},{96.44,143.45},{ 5.69,64.11},
  {78.34,119.62},{53.71,99.43},{13.96,37.77},{27.91,61.52},
  {56.45,105.71},{87.10,147.15},{64.17,102.53},{50.38,90.39},
  {58.45,94.56},{97.31,134.92},{93.78,145.98},{79.53,128.34},
  {55.63,107.26},{56.69,90.85},{17.24,58.16},{94.33,138.14},
  {84.76,142.18},{46.67,100.15},{10.99,44.86},{27.50,55.39},
  { 0.45,37.66},{10.59,44.13},{16.45,52.52},{33.86,76.86},
  {94.86,138.84},{91.84,172.00},{42.33,93.16},{74.66,119.60},
  {33.78,87.28},{ 7.40,28.44},{24.19,82.22},{96.11,160.10},
  {74.05,125.18},{71.22,109.69},{97.56,140.56},{ 8.84,47.04},
  {69.56,117.00},{ 8.41,42.56},{34.89,64.19},{85.11,115.74},
  {37.56,58.13},{12.59,48.39},{ 2.68,46.73},{62.78,99.69},
  {67.09,109.26},{87.49,126.32},{79.12,136.70},{ 7.17,57.74},
  {26.72,50.76},{35.61,88.60},{27.05,73.51},{71.28,140.53},
  {32.32,78.74},{ 0.41, 2.12},{79.39,135.08},{57.97,100.72},
  {60.76,116.50},{93.08,128.87},{78.11,120.73},{85.78,136.17},
  {53.83,96.46},{26.24,48.05},{78.45,133.71},{97.53,140.60},
  {70.22,129.62},{68.04,119.83},{57.92,97.63},{59.04,86.68},
  {91.06,140.68},{32.87,67.78},{96.49,155.45},{23.99,63.38},
  {83.65,136.74},{73.11,125.63},{12.25,55.87},{79.42,121.84},
  {57.58,111.96},{20.97,68.56},{70.31,136.14},{28.32,78.99},
  {32.13,84.83},{43.86,94.95},{58.82,95.92},{ 2.14,49.54},
  {32.16,76.00},{76.61,126.87},{86.98,117.12},{66.97,108.44},
  {69.58,124.78},{ 7.97,40.98},{86.41,129.47},{17.79,38.59},
  {25.39,62.60},{50.76,90.81},{59.80,104.48},{38.07,70.55},
  {35.73,85.25},{17.46,51.93},{71.16,124.51},{71.80,115.02},
  { 1.62,16.54},{91.26,155.89},{ 3.06,50.64},{29.66,64.84},
  { 7.09,53.98},{49.85,109.93},{14.36,45.56},{24.68,53.75},
  {51.44,105.05},{25.25,65.94},{40.55,78.79},{26.87,72.33},
  {27.84,74.47},{40.26,85.74},{62.00,110.62},{10.52,40.68},
  {40.51,76.33},{ 4.88,52.94},{28.29,81.65},{27.36,83.38},
  {79.84,125.13},{26.59,50.55},{51.70,108.24},{49.35,90.31},
  {32.62,69.09},{15.94,45.41},{69.23,121.30},{ 7.46,45.89},
  {84.07,129.79},{90.25,124.15},{47.88,79.40},{96.61,163.10},
  {15.78,73.16},{80.96,131.56},{31.45,62.49},{ 9.29,50.43},
  {35.18,84.87},{30.50,76.96},{89.17,143.70},{90.85,146.92},
  {59.81,96.39},{25.35,70.99},{ 5.98,55.09},{37.00,74.37},
  {80.19,144.98},{94.28,135.99},{86.46,150.28},{34.03,75.79},
  { 4.75,34.83},{70.46,110.49},{94.34,167.80},{70.80,115.88},
  {16.07,57.87},{68.62,112.90},{95.65,146.35},{ 1.02,57.34},
  { 1.78,41.23},{21.44,60.15},{73.24,114.19},{44.80,89.07},
  {84.42,139.57},{98.01,138.74},{ 9.17,33.87},{87.40,133.66},
  {52.38,86.57},{61.45,112.93},{65.82,127.01},{83.91,158.51},
  {93.63,158.06},{61.78,120.56},{24.30,59.86},{28.54,80.27},
  {95.60,143.52},{61.49,117.24},{10.29,60.58},{11.44,50.16},
  {51.35,108.25},{ 7.62,50.81},{69.37,114.61},{50.44,97.38},
  {11.35,39.19},{17.17,43.10},{61.58,132.61},{28.84,69.42},
  {30.47,70.96},{98.22,151.09},{86.43,136.81},{53.19,118.09},
  {95.29,155.91},{98.84,148.83},{ 2.22,26.13},{47.59,85.31},
  {73.84,109.83},{17.56,47.04},{26.58,73.75},{56.82,105.65},
  {79.98,134.87},{68.84,112.36},{88.87,142.96},{62.43,119.22},
  {76.15,135.84},{43.70,103.87},{16.29,58.86},{53.38,96.60},
  {15.83,57.27},{51.27,120.72},{47.59,64.87},{71.91,129.93},
  {98.51,152.48},{71.10,109.44},{77.99,127.90},{23.87,62.56},
  { 8.34,51.65},{79.64,132.34},{31.04,72.40},{44.82,81.69},
  {59.45,109.71},{84.15,130.99},{82.20,127.12},{31.22,83.20},
  {86.18,141.01},{75.15,134.96},{44.27,95.34},{95.97,144.24},
  {14.43,49.55},{34.57,82.98},{63.67,104.58},{66.22,95.67},
  {88.43,134.55},{63.52,123.11},{77.81,136.16},{ 8.71,34.79},
  {29.07,60.32},{19.49,62.23},{ 9.98,58.80},{78.34,117.98},
  {38.96,76.73},{55.21,87.60},{ 5.69,48.63},{74.40,128.08},
  {61.88,107.77},{65.91,111.44},{78.58,122.76},{39.78,91.82},
  {74.49,116.43},{45.47,99.10},{45.39,95.83},{76.30,120.24},
  {10.79,42.01},{52.12,85.62},{62.14,114.29},{77.15,134.46},
  {78.85,143.47},{22.68,53.16},{83.14,126.70},{ 7.41,29.99},
  {82.26,132.70},{72.14,126.79},{99.55,147.42},{85.72,134.53},
  {64.30,139.77},{38.81,51.80},{40.39,87.73},{25.10,72.42},
  {25.08,66.21},{ 5.81,56.34},{14.18,41.56},{94.84,167.41},
  { 8.54,69.86},{63.85,112.79},{57.40,109.06},{21.58,42.54},
  {43.64,78.24},{77.65,141.23},{81.50,142.13},{67.89,111.36},
  {72.87,120.96},{27.00,62.02},{61.38,127.54},{83.46,136.99},
  {54.76,108.99},{15.45,47.88},{60.05,100.26},{22.10,72.44},
  {28.35,65.66},{96.71,151.35},{86.94,116.64},{65.73,130.87},
  {85.94,126.58},{38.88,96.56},{69.21,127.86},{43.69,61.97},
  {69.87,113.70},{82.25,150.00},{94.99,151.04},{25.88,83.24},
  {81.60,115.26},{73.98,117.58},{43.63,88.14},{70.31,100.08},
  {55.26,100.57},{25.59,71.33},{34.63,76.64},{ 2.50,41.37},
  {54.69,95.27},{78.95,129.89},{67.28,112.36},{89.06,136.95},
  {64.28,113.88},{85.81,145.44},{25.24,57.88},{80.48,135.92},
  {92.34,143.08},{46.56,90.96},{88.74,134.15},{53.17,104.03},
  {31.02,59.35},{ 5.52,52.83},{80.57,120.09},{98.37,169.48},
  {12.81,29.56},{21.78,49.52},{42.46,97.45},{75.97,122.99},
  {32.01,99.32},{15.05,60.83},{75.69,123.02},{36.09,68.64},
  {57.21,106.14},{91.60,134.65},{73.77,118.23},{ 0.03,39.93},
  {55.27,107.03},{52.33,89.48},{53.54,107.47},{49.96,74.23},
  {37.17,74.84},{84.24,130.66},{14.91,60.54},{65.04,130.72},
  { 9.44,41.64},{81.34,129.14},{71.28,100.66},{43.43,93.58},
  {75.01,113.17},{88.82,151.61},{29.93,71.33},{65.46,116.24},
  {22.97,61.19},{51.00,96.99},{60.59,109.94},{ 7.00,41.67},
  {25.77,71.27},{75.31,124.33},{ 4.36,40.75},{87.14,156.88},
  {17.37,61.50},{46.75,82.19},{13.05,42.65},{78.95,128.33},
  {73.10,134.15},{74.36,135.48},{98.65,146.97},{88.12,132.71},
  {28.60,82.30},{15.10,49.96},{49.63,97.09},{87.28,135.25},
  {61.89,111.58},{34.90,93.67},{49.99,100.83},{94.71,170.51},
  {25.88,73.00},{61.84,110.97},{55.36,102.57},{67.98,132.12},
  {21.37,64.84},{13.94,48.72},{43.71,90.80},{91.48,146.34},
  {31.14,59.33},{10.31,48.50},{94.61,149.76},{71.41,120.34},
  {21.14,64.98},{32.06,73.94},{66.20,105.95},{33.15,61.90},
  {86.19,131.68},{83.78,114.83},{54.33,96.58},{33.55,80.32},
  {96.60,170.21},{11.97,28.29},{63.59,124.56},{55.26,100.67},
  {63.14,123.76},{32.40,83.07},{18.12,63.35},{87.59,126.65},
  {50.10,97.98},{65.93,124.39},{62.49,103.95},{ 6.82,27.69},
  {46.80,85.68},{62.95,104.80},{45.74,78.09},{61.82,126.51},
  {92.80,134.13},{57.48,114.43},{74.40,109.61},{60.92,108.78},
  {30.89,79.66},{94.62,139.00},{49.83,98.73},{ 6.11,31.51},
  {45.54,88.00},{19.50,42.96},{25.52,66.51},{73.81,138.08},
  {29.34,88.98},{ 8.18,27.09},{30.62,65.44},{46.31,113.87},
  {41.02,86.84},{90.96,164.15},{22.01,45.41},{ 4.89,39.78},
  {95.35,142.65},{71.94,110.75},{63.35,105.06},{ 4.09,40.40},
  {88.54,139.61},{62.07,112.54},{27.70,56.09},{76.66,137.20},
  {83.12,143.20},{67.15,120.22},{60.00,99.28},{87.13,145.36},
  {94.71,148.17},{25.37,60.43},{64.45,125.47},{41.69,82.97},
  {39.60,86.26},{23.70,59.91},{95.30,155.36},{56.25,96.91},
  {10.09,46.74},{ 0.76,40.83},{59.30,94.67},{94.01,140.36},
  {88.44,141.46},{50.91,92.29},{42.26,99.49},{31.75,63.82},
  {48.52,104.01},{ 5.91,42.20},{80.60,128.78},{25.14,63.22},
  {74.28,124.28},{15.63,74.24},{97.86,149.97},{79.77,140.75},
  {65.69,118.02},{73.56,134.34},{17.85,53.07},{63.86,91.88},
  {13.97,53.39},{32.44,72.72},{ 8.17,56.60},{58.57,116.54},
  {37.35,65.08},{98.78,158.02},{70.98,130.54},{ 6.81,35.93},
  {85.39,160.17},{34.97,80.61},{ 2.64,43.13},{56.77,104.59},
  {81.91,112.09},{31.48,60.20},{34.81,61.30},{11.12,49.56},
  {71.51,128.10},{73.49,124.35},{72.99,112.97},{31.50,83.11},
  {99.98,147.74},{ 2.81,53.17},{42.82,98.70},{59.16,100.75},
  {23.89,72.61},{81.97,159.88},{46.85,94.22},{36.55,93.76},
  {64.96,95.23},{15.11,53.48},{65.91,113.75},{69.19,107.31},
  {28.06,63.30},{58.54,114.26},{89.15,153.48},{42.06,77.31},
  {40.50,76.81},{86.00,146.02},{ 3.20,48.79},{58.69,97.33},
  {35.28,78.08},{ 9.10,46.61},{25.91,66.15},{57.01,103.41},
  {14.91,73.97},{96.76,161.54},{99.67,149.40},{72.54,137.18},
  {30.50,74.61},{42.00,93.95},{59.93,109.89},{66.51,120.16},
  {80.06,138.60},{10.36,51.63},{ 1.60,27.94},{30.78,62.07},
  {66.55,102.64},{61.32,120.77},{91.03,150.60},{53.45,99.29},
  {42.37,78.63},{62.46,111.62},{66.04,112.54},{93.06,151.16},
  {51.95,88.51},{43.30,113.42},{64.13,99.00},{45.53,94.25},
  {39.47,87.77},{29.37,80.52},{92.61,162.12},{21.69,47.65},
  { 1.05,64.71},{40.01,92.37},{97.62,155.20},{70.10,106.73},
  {50.52,80.33},{11.96,51.24},{26.52,76.52},{77.52,132.84},
  {30.52,78.72},{30.52,78.19},{38.11,88.26},{76.86,128.87},
  {28.28,56.37},{30.49,65.71},{59.67,108.95},{89.64,157.43},
  {30.25,81.36},{22.29,69.28},{35.55,84.75},{68.06,113.95},
  {51.60,84.29},{10.09,43.39},{76.55,134.77},{71.33,115.11},
  {51.60,75.80},{54.79,109.48},{96.69,155.10},{14.77,49.52},
  {95.02,148.05},{96.72,141.18},{63.72,98.83},{91.93,140.47},
  {64.34,123.35},{88.86,137.02},{64.18,98.25},{70.04,110.77},
  {94.17,140.07},{75.24,124.76},{44.64,98.74},{87.41,153.95},
  {92.46,144.38},{19.06,66.13},{57.71,112.93},{ 7.45,39.86},
  {27.55,73.56},{56.19,108.21},{ 6.01,40.37},{62.74,89.47},
  {16.17,59.02},{ 8.79,30.74},{83.08,130.52},{24.23,54.52},
  {85.16,149.10},{24.81,90.03},{25.92,54.58},{54.33,82.03},
  {63.90,102.12},{68.66,100.77},{ 5.95,56.49},{42.26,93.76},
  {31.60,90.12},{44.62,89.67},{89.70,139.94},{24.77,52.46},
  {74.16,113.75},{85.36,142.27},{ 2.84,76.16},{91.59,150.81},
  {70.65,125.99},{26.70,76.22},{95.56,152.00},{ 0.44,27.70},
  {20.27,67.29},{ 5.23,37.86},{10.10,40.67},{74.38,130.22},
  {89.36,158.51},{ 4.21,42.73},{42.73,69.67},{72.56,111.50},
  {53.35,114.22},{28.76,96.68},{61.84,110.08},{16.56,62.79},
  {83.69,137.28},{48.91,86.74},{32.35,65.27},{29.44,75.42},
  {30.65,77.48},{85.56,144.14},{90.86,150.13},{81.44,126.15},
  {47.80,89.15},{13.73,41.35},{25.43,79.82},{58.07,92.59},
  {22.91,42.48},{49.14,87.71},{55.98,114.64},{ 8.82,45.15},
  {90.55,153.99},{81.55,138.12},{82.55,136.84},{51.00,101.17},
  {27.58,74.60},{37.31,77.27},{ 1.12,44.83},{88.58,143.95},
  {22.77,75.64},{97.08,157.64},{66.49,108.31},{98.86,156.70},
  {45.64,88.45},{89.75,139.02},{30.57,69.62},{36.48,85.01},
  {98.72,154.40},{30.12,76.32},{73.34,117.76},{16.37,52.48},
  {69.14,134.69},{98.21,174.31},{80.43,120.96},{56.01,100.03},
  { 1.48,43.90},{30.68,56.24},{65.36,121.74},{ 4.45,30.83}
};


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);
}

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);
}

__device__ double d_residual_error(double x, double y, double
m, double c) {
  double e = (m * x) + c - y;
  return e * e;
}
__global__ void d_rms_error(double *m, double *c, double
*error_sum_arr, point_t *d_data) {

     int i = threadIdx.x + blockIdx.x * blockDim.x;

  error_sum_arr[i] = d_residual_error(d_data[i].x, d_data[i].y,
*m, *c);
}



int main() {
  int i;
  double bm = 1.3;
  double bc = 10;
  double be;
  double dm[8];
  double dc[8];
  double e[8];
  double step = 0.01;
  double best_error = 999999999;
  int best_error_i;
  int minimum_found = 0;

  double om[] = {0,1,1, 1, 0,-1,-1,-1};
  double oc[] = {1,1,0,-1,-1,-1, 0, 1};

     struct timespec start, finish;
  long long int time_elapsed;

  clock_gettime(CLOCK_MONOTONIC, &start);

     cudaError_t error;


     double *d_dm;
  double *d_dc;
```

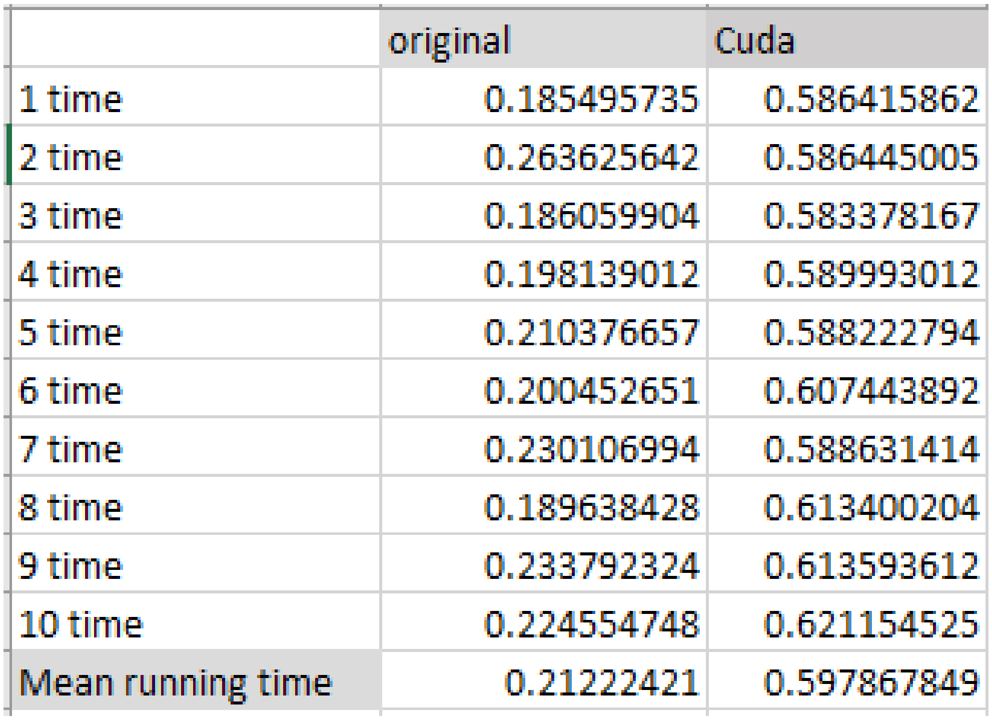
```
     double *d_error_sum_arr;
     point_t *d_data;

  be = rms_error(bm, bc);

     //Allocate memory for d_dm
     error = cudaMalloc(&d_dm, (sizeof(double) * 8));
     if(error){
     fprintf(stderr, "cudaMalloc on d_dm returned %d %s\n",
error,
     cudaGetErrorString(error));
     exit(1);
     }

     //Allocate memory for d_dc
     error = cudaMalloc(&d_dc, (sizeof(double) * 8));
     if(error){
     fprintf(stderr, "cudaMalloc on d_dc returned %d %s\n",
error,
       cudaGetErrorString(error));
     exit(1);
     }

     //Allocate memory for d_error_sum_arr
     error = cudaMalloc(&d_error_sum_arr, (sizeof(double) *
1000));
     if(error){
     fprintf(stderr, "cudaMalloc on d_error_sum_arr returned %d
%s\n", error,
       cudaGetErrorString(error));
     exit(1);
     }

     //Allocate memory for d_data
     error = cudaMalloc(&d_data, sizeof(data));
     if(error){
     fprintf(stderr, "cudaMalloc on d_data returned %d %s\n",
error,
       cudaGetErrorString(error));
     exit(1);
     }

  while(!minimum_found) {
    for(i=0;i<8;i++) {
      dm[i] = bm + (om[i] * step);
      dc[i] = bc + (oc[i] * step);
    }
```

```
          //Copy memory for dm to d_dm
     error = cudaMemcpy(d_dm, dm, (sizeof(double) * 8),
cudaMemcpyHostToDevice);
     if(error){
     fprintf(stderr, "cudaMemcpy to d_dm returned %d %s\n",
error,
      cudaGetErrorString(error));
     }

          //Copy memory for dc to d_dc
     error = cudaMemcpy(d_dc, dc, (sizeof(double) * 8),
cudaMemcpyHostToDevice);
     if(error){
     fprintf(stderr, "cudaMemcpy to d_dc returned %d %s\n",
error,
      cudaGetErrorString(error));
     }

          //Copy memory for data to d_data
     error = cudaMemcpy(d_data, data, sizeof(data),
cudaMemcpyHostToDevice);
     if(error){
     fprintf(stderr, "cudaMemcpy to d_data returned %d %s\n",
error,
      cudaGetErrorString(error));
     }

    for(i=0;i<8;i++) {
               //Host variable storing the array returned from
the kernel function.
               double h_error_sum_arr[1000];

               //Stores the total sum of the values from the
error sum array.
               double error_sum_total;

               //Stores the mean of the total sum of the error
sums.
               double error_sum_mean;

               //Call the rms_error function using 100 blocks
and 10 threads.
               d_rms_error <<<100,10>>>(&d_dm[i], &d_dc[i],
d_error_sum_arr, d_data);
               cudaThreadSynchronize();
               //Copy memory for d_error_sum_arr
            error = cudaMemcpy(&h_error_sum_arr,
d_error_sum_arr, (sizeof(double) * 1000),
cudaMemcpyDeviceToHost);
            if(error){
         fprintf(stderr, "cudaMemcpy to error_sum returned %d
%s\n", error,
           cudaGetErrorString(error));
            }

               //Loop through the error sum array returned from
the kernel function
               for(int j=0; j<n_data; j++) {
                    //Add each error sum to the error sum
total.
          error_sum_total += h_error_sum_arr[j];
          }

               //Calculate the mean for the error sum.
               error_sum_mean = error_sum_total / n_data;

               //Calculate the square root for the error sum
mean.
               e[i] = sqrt(error_sum_mean);

      if(e[i] < best_error) {
        best_error = e[i];
        best_error_i = i;
      }

               //Reset the error sum total.
               error_sum_total = 0;
    }

     //printf("best m,c is %lf,%lf with error %lf in direction
%d\n",
     //dm[best_error_i], dc[best_error_i], best_error,
best_error_i);


    if(best_error < be) {
      be = best_error;
      bm = dm[best_error_i];
      bc = dc[best_error_i];
    } else {
      minimum_found = 1;
    }
```

```
  }

     //Free memory for d_dm
     error = cudaFree(d_dm);
     if(error){
          fprintf(stderr, "cudaFree on d_dm returned %d %s\n",
error,
          cudaGetErrorString(error));
          exit(1);
     }

     //Free memory for d_dc
     error = cudaFree(d_dc);
     if(error){
          fprintf(stderr, "cudaFree on d_dc returned %d %s\n",
error,
               cudaGetErrorString(error));
          exit(1);
     }

     //Free memory for d_data
     error = cudaFree(d_data);
     if(error){
          fprintf(stderr, "cudaFree on d_data returned %d
%s\n", error,
          cudaGetErrorString(error));
          exit(1);
     }

     //Free memory for d_error_sum_arr
     error = cudaFree(d_error_sum_arr);
     if(error){
          fprintf(stderr, "cudaFree on d_error_sum_arr returned
%d %s\n", error,
          cudaGetErrorString(error));
          exit(1);
     }

  printf("minimum m,c is %lf,%lf with error %lf\n", bm, bc,
be);

     //Get the system time after we have run the linear
regression function.
     clock_gettime(CLOCK_MONOTONIC, &finish);

     //Calculate the time spent between the start time and end
time.
```

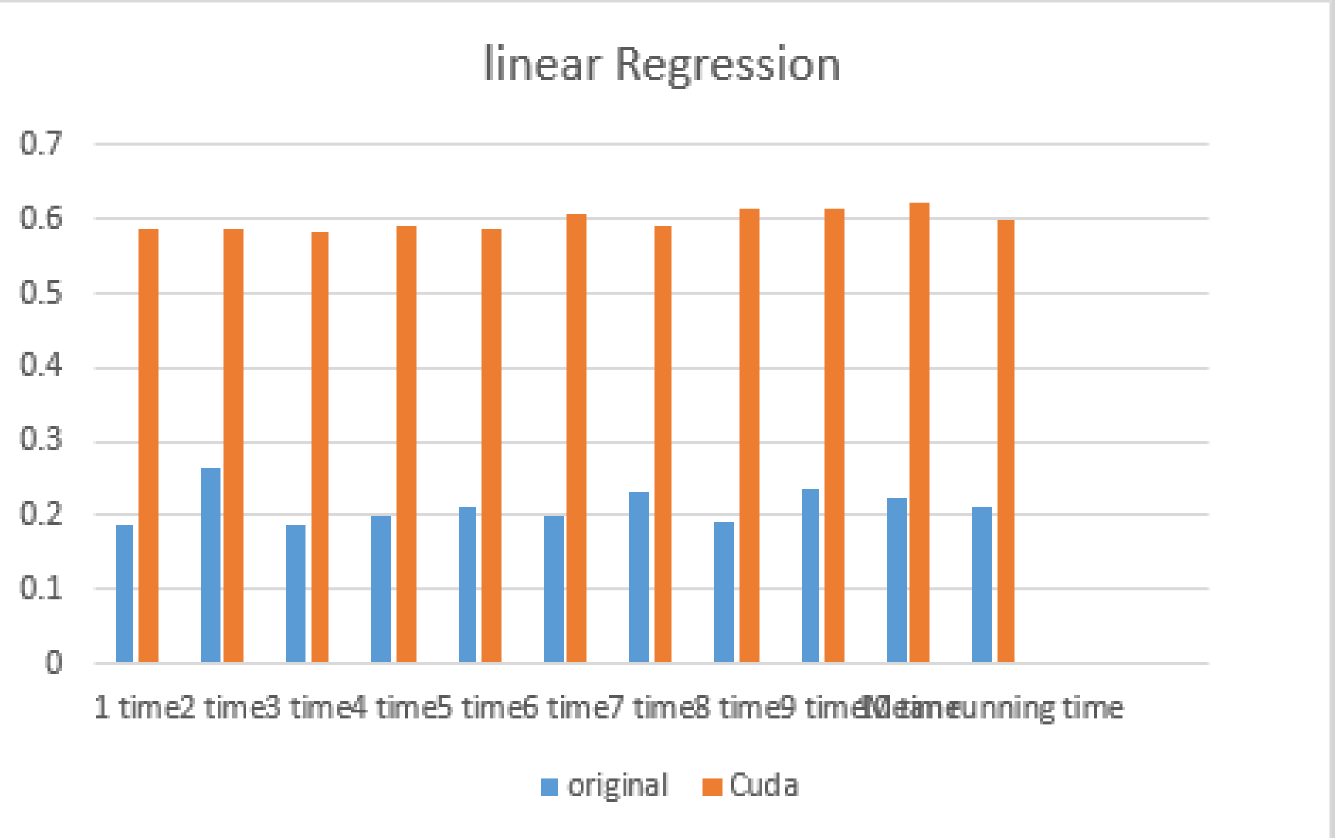
```
  time_difference(&start, &finish, &time_elapsed);

     //Output the time spent running the program.
  printf("Time elapsed was %lldns or %0.9lfs\n", time_elapsed,
         (time_elapsed/1.0e9));

  return 0;
}
```

Table showing the running times of original and CUDA version of Linear Regression.



	original	Cuda
1 time	0.185495735	0.586415862
2 time	0.263625642	0.586445005
3 time	0.186059904	0.583378167
4 time	0.198139012	0.589993012
5 time	0.210376657	0.588222794
6 time	0.200452651	0.607443892
7 time	0.230106994	0.588631414
8 time	0.189638428	0.613400204
9 time	0.233792324	0.613593612
10 time	0.224554748	0.621154525
Mean running time	0.21222421	0.597867849

Bar chart showing the running time of original and CUDA version.



From the above chart and the table we can see the time taken by the CUDA is marginally more than the original. The output shown the mean time taken by the original is 0.2122 and the time taken by the CUDA is 0.5978 which is nearly 3 time more in rate. It is especially caused the programming pattern in CUDA. In CUDA we make more threads to complete task in faster rate. But in every case it doesn’t go in that way which can be seen in the above program. Having numerous threads it takes fraction of time to divide the tasks for each threads, run and process. Which output must be merge later to give desire output which especially consumes more time? But in original it does the work in one process. Hence the CUDA is time consuming.

3 MPI

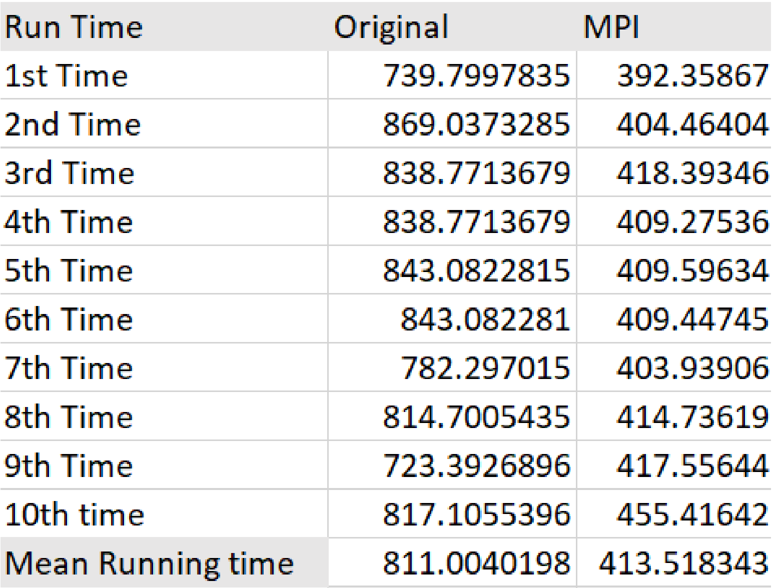
3.1 Password Cracking

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <crypt.h>
#include "time_diff.h"
#include <stdio.h>
#include <mpi.h>
#include <unistd.h>
/***********************************************************************
*******
  Demonstrates how to crack an encrypted password using a simple
  "brute force" algorithm. Works on passwords that consist only of 2
uppercase
  letters and a 2 digit integer. Your personalised data set is included
in the
  code.

  Compile with:
    cc -o CrackAZ99-With-Data CrackAZ99-With-Data.c -lcrypt

  If you want to analyse the results then use the redirection operator
to send
  output to a file that you can view using an editor or the less
utility:

    ./CrackAZ99-With-Data > results.txt

  Dr Kevan Buckley, University of Wolverhampton, 2018
************************************************************************
******/
int n_passwords = 4;

char *encrypted_passwords[] = {

"$6$KB$3MiAO5oLs/.coZCPQ2QYOy8Ozo3v7QzGdwBEv3N7E0pJen3CJ63DmYXIZz6KEsykH
mGsu3Dh1KCNe0niN0wvx/",

"$6$KB$J4IvaQyaBTaxqgcXSRy1ekd5MJe8ZEwlgiHVGz1lBlN7E3IoHmL6crz9230xUd9ci
GjXbTf60drGnuUg9mFm20",

"$6$KB$nTF3p1cNAysQAF/gFwaXB.7L2YEguvQi4qLJi/aSkN1MP1vXUGgt36FEFbdD8tElu
sQda178XE.kGMBoZgLiB0",

"$6$KB$rmRaZq1xk.DPrBiN3K2XH5mnujGY51hILP8v4RYIllaVDgqr3in5hKnpm52i9VS/s
gJ0OBjD5u62k0sSleQwD/"
};
```

```
/**
 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 function_1(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
  char *enc;       // Pointer to the encrypted password
  int count = 0;   // The number of combinations explored so far

  substr(salt, salt_and_encrypted, 0, 6);

  for(x='A'; x<='M'; x++){
    for(y='A'; y<='Z'; y++){
      for(z=0; z<=99; z++){
 printf("Instance 1");
        sprintf(plain, "%c%c%02d", x, y, z);
        enc = (char *) crypt(plain, salt);
        count++;
        if(strcmp(salt_and_encrypted, enc) == 0){
          printf("#%-8d%s %s\n", count, plain, enc);
        } else {
          printf(" %-8d%s %s\n", count, plain, enc);
        }
      }
    }
  }
  printf("%d solutions explored\n", count);
}

void function_2(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
  char *enc;       // Pointer to the encrypted password
  int count = 0;   // The number of combinations explored so far

  substr(salt, salt_and_encrypted, 0, 6);

  for(x='N'; x<='Z'; x++){
    for(y='A'; y<='Z'; y++){
      for(z=0; z<=99; z++){
        printf("Instance 2");
        sprintf(plain, "%c%c%02d", x, y, z);
        enc = (char *) crypt(plain, salt);
        count++;
        if(strcmp(salt_and_encrypted, enc) == 0){
          printf("#%-8d%s %s\n", count, plain, enc);
        } else {
          printf(" %-8d%s %s\n", count, plain, enc);
        }
      }
    }
  }
  printf("%d solutions explored\n", count);
}

int main(int argc, char *argv[]){

  struct timespec start, finish;
  long long int difference;
  int account = 0;
  clock_gettime(CLOCK_MONOTONIC, &start);



int size, rank;

  MPI_Init(NULL, NULL);
  MPI_Comm_size(MPI_COMM_WORLD, &size);
  MPI_Comm_rank(MPI_COMM_WORLD, &rank);
  if(size != 3) {
    if(rank == 0) {
      printf("This program needs to run on exactly 3 processes\n");
    }
  } else {
    if(rank ==0){
      int x;
      int y;
      int i;
       MPI_Send(&x, 1, MPI_INT, 1, 0, MPI_COMM_WORLD);
      MPI_Send(&y, 1, MPI_INT, 2, 0, MPI_COMM_WORLD);

    } else {
      if(rank == 1){
       int i;
```

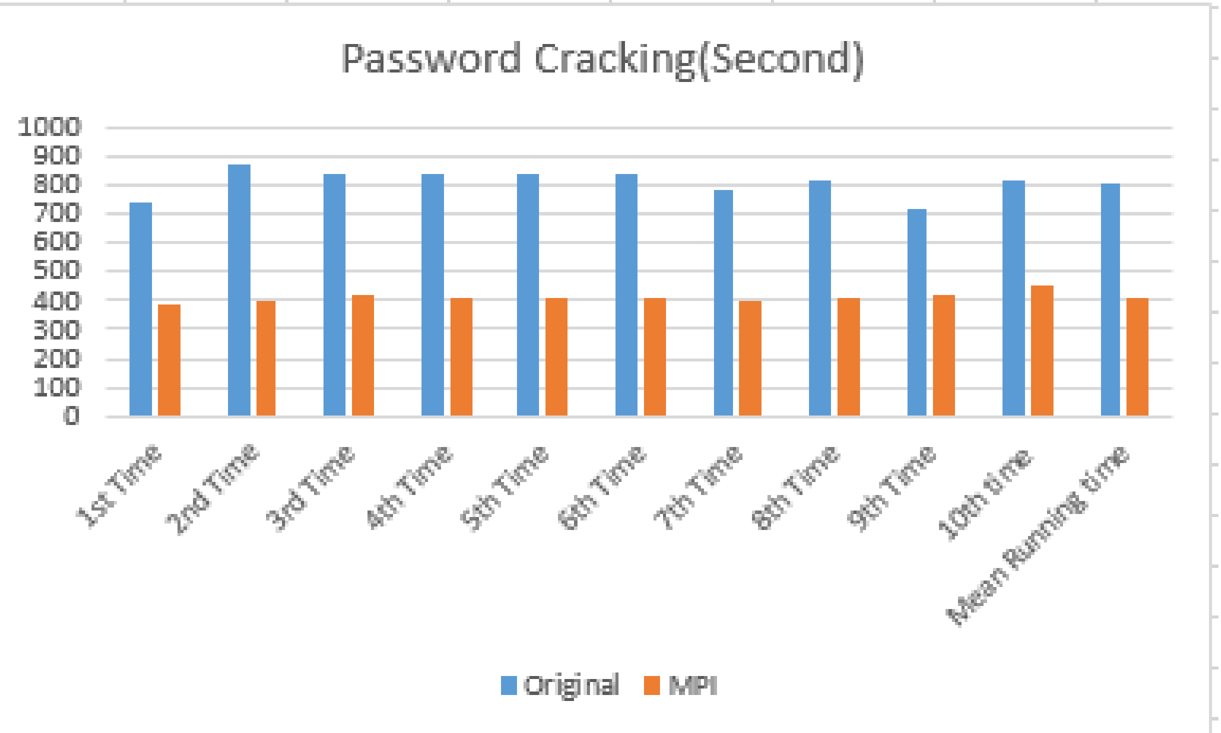
```
         int number = rank + 10;
        MPI_Recv(&number, 1, MPI_INT, 0, 0,
MPI_COMM_WORLD,MPI_STATUS_IGNORE);
        for ( i = 0; i<n_passwords;i<i++){
                function_1(encrypted_passwords[i]);
        }
       }
        else if(rank == 2){
        int i;
        int number = rank + 10;
        MPI_Recv(&number, 1, MPI_INT, 0, 0,
MPI_COMM_WORLD,MPI_STATUS_IGNORE);
        for ( i = 0; i<n_passwords;i<i++){
                function_2(encrypted_passwords[i]);
        }
        }
     }
  }
  MPI_Finalize();

  clock_gettime(CLOCK_MONOTONIC, &finish);
  time_difference(&start, &finish, &difference);

  printf("Elapsed Time: %9.5lfs\n", difference/1000000000.0);


  return 0;
}
```

Table showing the running times of original and MPI versions.



Run Time	Original	MPI
1st Time	739.7997835	392.35867
2nd Time	869.0373285	404.46404
3rd Time	838.7713679	418.39346
4th Time	838.7713679	409.27536
5th Time	843.0822815	409.59634
6th Time	843.082281	409.44745
7th Time	782.297015	403.93906
8th Time	814.7005435	414.73619
9th Time	723.3926896	417.55644
10th time	817.1055396	455.41642
Mean Running time	811.0040198	413.518343

Bar chart showing running of password cracking of MPI and original.



The above data shows the running time of MPI to crack the password is less than original. In the MPI programming it uses 2 instance to work parallel with master instance which is running different cores along with it. It is using the same processor but different core in a single processes which lowers the execution which also shows the number of cores increase the execution time. MPI Is designed to run on different PC to reduce the execution time by huge margin. But due to lack of resources our program is run in a same pc which can cause the overall performance of MPI. So that the execution time of MPI is less.

3.2 Image processing

Code of MPI 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 <ctype.h>
#include <errno.h>
#include <sys/stat.h>
#include <string.h>
#include <mpi.h>
#include <math.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 -o image_processing_mpi image_processing_mpi.c -lm -
lglut -lGL
```

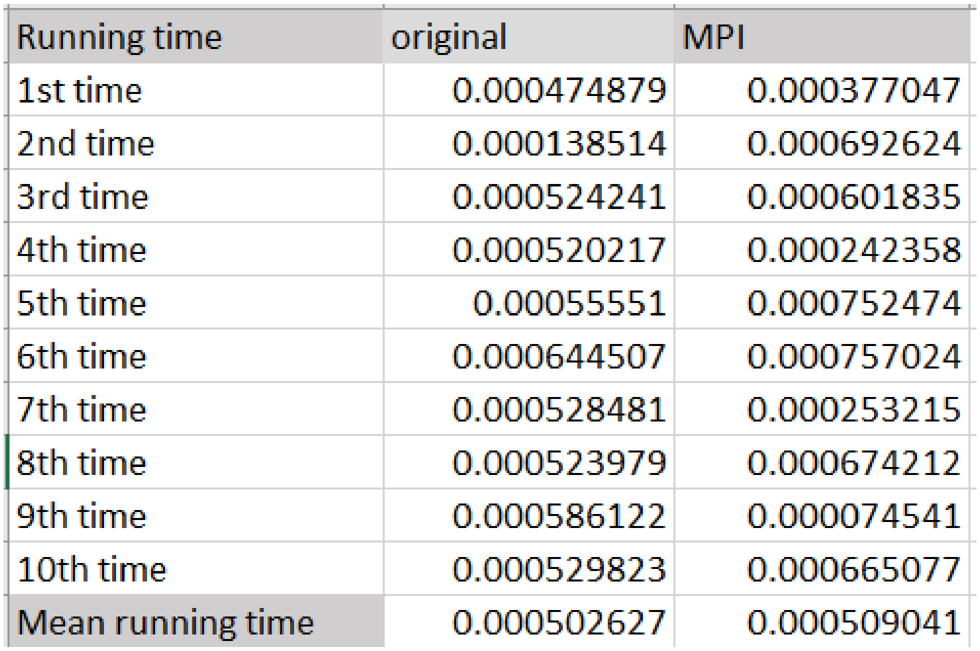
```
  To run:
  mpirun -n 5 ./image_processing_mpi

  Dr Kevan Buckley, University of Wolverhampton, 2018
***************************************************************
***************/
#define width 100
#define height 72

unsigned char image[], results[width * height];
int stIndex, endIndex;
int time_difference(struct timespec *start,struct timespec
*finish, long long int *difference) ;
void detect_edges(unsigned char *in, unsigned char *out);
void sigint_callback(int signal_number);
static void display();
void tidy_and_exit();


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;
      }
    }
  }
}




void tidy_and_exit() {
  exit(0);
}

void sigint_callback(int signal_number){
  printf("\nInterrupt from keyboard\n");
  tidy_and_exit();
}

static void display() {
  glClear(GL_COLOR_BUFFER_BIT);
  glRasterPos4i(-1, -1, 0, 1);
  glDrawPixels(width, height, GL_LUMINANCE, GL_UNSIGNED_BYTE,
image);
  glRasterPos4i(0, -1, 0, 1);
  glDrawPixels(width, height, GL_LUMINANCE, GL_UNSIGNED_BYTE,
results);
  glFlush();
}

static void key_pressed(unsigned char key, int x, int y) {
  switch(key){
    case 27: // escape
      tidy_and_exit();
      break;
    default:
      printf("\nPress escape to exit\n");
      break;
  }
}

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(int argc, char **argv) {
  signal(SIGINT, sigint_callback);

  printf("image dimensions %dx%d\n", width, height);

  int size, rank;

  MPI_Init(NULL, NULL);
  MPI_Comm_size(MPI_COMM_WORLD, &size);
  MPI_Comm_rank(MPI_COMM_WORLD, &rank);

  if (size != 5 ){
  if(rank != 0) {
      printf("This program needs to run on exactly 4
processes\n");
      exit(-1);
    }
  }
  if (rank ==0){
     //stIndex = 0;
     //endIndex = 1799;
     struct timespec start, finish;
     long long int time_elapsed;
     clock_gettime(CLOCK_MONOTONIC, &start);


  /*struct timespec start, finish;
  long long int time_elapsed;
  clock_gettime(CLOCK_MONOTONIC, &start);*/

     //detect_edges(image, results);


     MPI_Recv(&results[0], 1800, MPI_UNSIGNED_CHAR, 1, 0,
MPI_COMM_WORLD,MPI_STATUS_IGNORE);
     MPI_Recv(&results[1800], 1800, MPI_UNSIGNED_CHAR, 2, 0,
MPI_COMM_WORLD,MPI_STATUS_IGNORE);
```

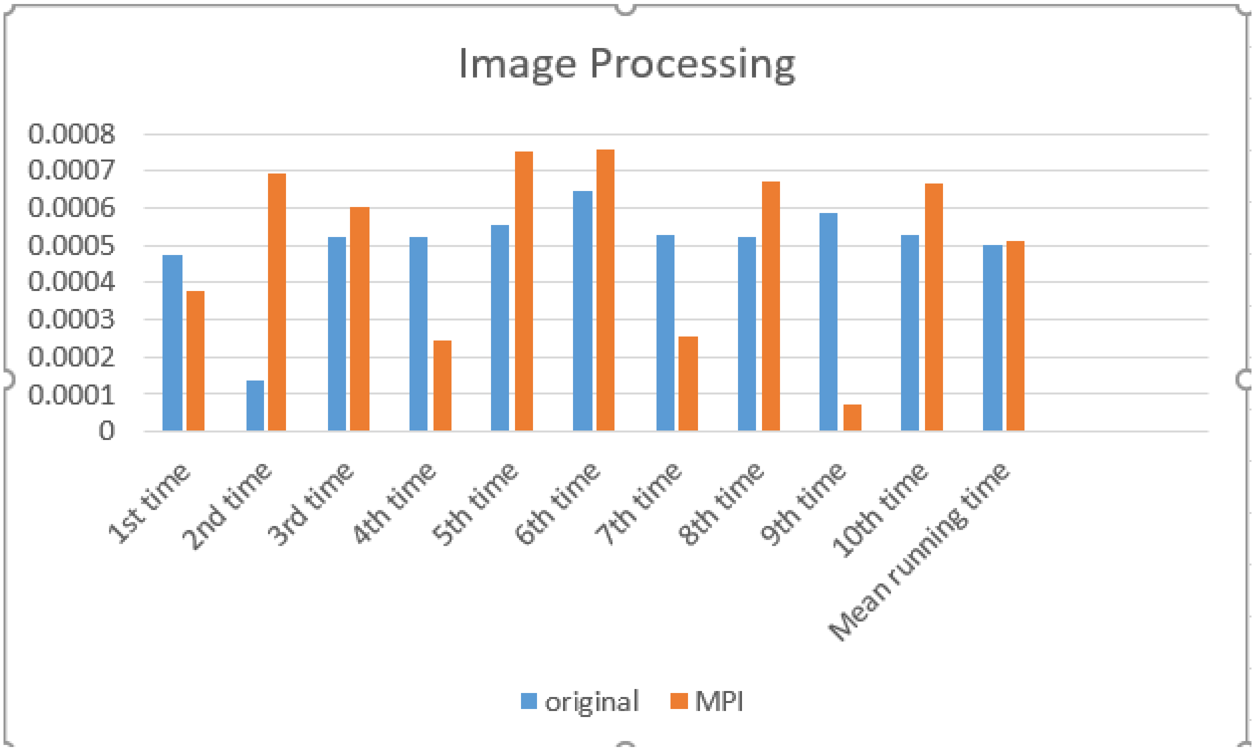
```
     MPI_Recv(&results[3600], 1800, MPI_UNSIGNED_CHAR, 3, 0,
MPI_COMM_WORLD,MPI_STATUS_IGNORE);
     MPI_Recv(&results[5400], 1800, MPI_UNSIGNED_CHAR, 4, 0,
MPI_COMM_WORLD,MPI_STATUS_IGNORE);

     clock_gettime(CLOCK_MONOTONIC, &finish);
     time_difference(&start, &finish, &time_elapsed);
     printf("Time elapsed was %lldns or %0.9lfs\n",
time_elapsed, (time_elapsed/1.0e9));


  glutInit(&argc, argv);
  glutInitWindowSize(width * 2,height);
  glutInitDisplayMode(GLUT_SINGLE | GLUT_LUMINANCE);

  glutCreateWindow("6CS005 Image Progessing Courework");
  glutDisplayFunc(display);
  glutKeyboardFunc(key_pressed);
  glClearColor(0.0, 1.0, 0.0, 1.0);

  glutMainLoop();

  tidy_and_exit();

  return 0;
}
  else if (rank == 1){
     stIndex = 0;
     endIndex =1799;

     detect_edges(image, results);
     MPI_Send(&results[0], 1800, MPI_UNSIGNED_CHAR, 0, 0,
MPI_COMM_WORLD);
}

  else if (rank == 2){
     stIndex = 1800;
     endIndex =3599;

     detect_edges(image, results);
     MPI_Send(&results[1800], 1800, MPI_UNSIGNED_CHAR, 0, 0,
MPI_COMM_WORLD);
}


 else if (rank == 3){
     stIndex = 3600;
     endIndex =5399;

     detect_edges(image, results);
     MPI_Send(&results[3600], 1800, MPI_UNSIGNED_CHAR, 0, 0,
MPI_COMM_WORLD);
}

 else if (rank == 4){
     stIndex = 5400;
     endIndex =7199;

     detect_edges(image, results);
     MPI_Send(&results[5400], 1800, MPI_UNSIGNED_CHAR, 0, 0,
MPI_COMM_WORLD);
}

MPI_Finalize();
return 0;
}





unsigned char image[]
={255,255,255,255,0,0,255,255,255,255,255,255,255,0,255,255,0,0
,

255,0,0,255,255,255,255,0,0,0,0,255,255,255,255,255,0,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,0,0,0,0,255,255,255,255
,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,0,255,255,255,255,255,0,0,255,255,255,255,255,2
55,255,
  0,0,255,0,0,255,255,0,255,255,255,255,255,0,0,0,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,0,255,255,0,0,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,0,0,255,255,255,255,255,0,2
55,255,
  255,255,0,0,255,255,0,0,0,0,255,255,0,0,255,255,255,255,255,

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
,0,0,255,

255,255,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,2
55,255,
  255,255,0,0,0,0,0,0,0,255,255,0,0,0,0,255,255,255,0,

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
,0,255,255,

255,255,0,0,255,255,255,0,0,255,255,0,0,0,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,0,0,255,255,255,255,255,0,0,0,255,255,255,255,255,255,0,0,

0,255,255,255,0,0,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,0,
  255,255,0,0,255,255,255,0,0,255,255,255,255,0,0,0,0,0,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
255,255,255,255,255,255,0,0,0,255,255,255,255,0,0,255,255,255,2
55,

255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,0,0,255,0,0,255,255,255,0,0,255,255,255,255,0,

0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,

255,255,255,255,255,255,255,255,255,255,255,0,255,0,255,255,255
,255,255,

0,255,255,255,255,0,0,255,255,0,255,255,255,255,255,255,255,255
,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,0,0,255,255,255,255,0,0,0,255,255,0,0,0,0,0,0,

0,0,255,255,255,0,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,0,
  0,255,255,255,255,0,0,0,0,0,0,0,255,255,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,255,255,0,
  0,0,0,255,255,255,0,0,255,255,0,0,255,255,0,0,0,0,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,0,0,255,255,255,255,0,0,0,0,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
255,255,255,0,255,255,255,255,255,255,255,255,255,255,0,0,0,255
,255,
  0,0,255,255,255,0,0,255,255,255,255,0,0,255,255,0,0,0,0,

0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,

255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,0,0,0,0,0,0,255,255,255,255,255,255,255,255
,
  255,0,0,255,255,0,0,255,255,255,0,0,255,255,255,255,0,0,255,

255,255,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,255
,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,0,0,255,255,255,0,0,255
,255,

255,255,255,255,255,255,255,0,0,255,0,0,255,255,255,255,0,0,255
,

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
,255,255,0,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,0,0,0,255,255,255,0,0,255,255,2
55,

255,255,0,255,255,255,255,255,255,255,255,255,255,0,0,0,0,255,2
55,
255,255,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,255,

255,0,0,255,255,255,255,0,0,255,255,255,255,255,255,255,255,255
,0,

0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

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,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0
,0,255,

255,255,0,0,0,255,0,0,255,255,255,255,255,0,255,255,255,255,255
,

255,255,255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,2
55,255,255,

255,0,0,255,255,255,255,255,0,0,255,0,0,255,255,255,255,255,0,

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,0,0,0,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,0,255,255,255,255,0,0,255,255,255,255,255,255,0,0,255,0,0,

255,255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,0,0,255,255,0,0,255,255,255,255
,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,0,0,255,255,255,0,0,255,255,255,255,255
,255,

0,0,255,255,0,255,255,255,255,255,0,0,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,255,255
,255,0,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,0,255,255,255,2
55,0,0,

255,255,255,255,255,255,0,255,255,0,0,255,255,255,255,255,0,255
,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
255,255,255,255,255,255,255,255,255,255,0,255,0,0,255,255,255,2
55,0,

0,255,255,255,0,0,255,0,0,0,0,255,255,255,255,255,255,255,255,

255,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,0,0,

255,255,255,0,0,255,255,255,255,255,0,0,255,255,0,0,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,255,255,0
,
  0,255,255,255,0,0,255,255,255,0,0,0,0,0,255,255,255,255,255,

255,255,255,255,255,0,0,0,255,255,0,0,255,255,255,255,255,255,2
55,
  255,255,255,0,0,0,0,255,255,0,0,255,255,255,255,0,0,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0
,255,

255,255,255,255,0,0,255,255,0,0,0,255,255,255,255,0,0,255,255,

255,255,255,255,255,255,0,255,255,255,0,0,255,255,255,255,0,0,2
55,
  255,255,255,255,255,255,255,255,0,0,0,0,0,255,255,0,0,0,0,

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
,0,0,255,
  255,0,0,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,255,

255,0,0,255,255,255,255,255,255,255,255,0,0,0,255,0,0,255,255,

255,255,0,0,255,255,255,255,255,255,255,255,255,0,255,255,255,0
,0,
255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,0,0,255,255,0,0,255,255,255,255,255,255,255,0,0,0,255,
  255,255,255,0,0,255,255,0,0,255,0,0,0,0,255,255,255,0,0,

255,255,255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,2
55,0,

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,
  255,255,0,255,255,255,255,255,0,0,255,255,0,0,255,0,0,0,0,
  255,255,0,0,255,255,255,255,0,0,255,255,0,0,0,0,0,255,255,

255,255,255,255,0,0,255,255,255,0,0,0,255,255,255,255,255,255,2
55,

255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,0,0,255,255,255,255,0,0,255,255,0,0
,

0,0,0,255,255,255,255,0,0,255,255,255,255,0,0,255,255,255,255,

255,255,255,255,255,255,255,255,0,0,255,255,255,0,0,0,255,255,2
55,

255,255,255,255,255,255,255,255,255,0,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,0,0,0,255,255,255,255,255,255,255,0,0,255,255,255,255,0
,
  0,0,255,255,0,0,255,255,255,255,255,255,255,0,0,0,0,0,0,
255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,255,255,2
55,0,

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,

255,255,255,255,255,0,0,0,0,0,0,255,255,255,255,255,255,255,0,
  0,0,0,0,0,0,0,255,255,0,0,255,255,255,255,255,255,255,0,

0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0
,0,

255,255,255,255,255,0,0,255,255,0,0,255,255,255,255,255,255,255
,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,0,0,255,255,255,255,0,0,255
,255,
  255,255,255,255,0,0,0,0,0,255,255,0,0,255,255,0,0,255,0,

0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,0,255,255,255,0,0,0,0,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,0,0,255,255,255,0,0,255,255
,255,

255,0,0,255,255,255,255,255,255,255,0,0,255,255,255,255,0,0,255
,

255,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,2
55,

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,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,2
55,255,

0,0,255,255,255,0,0,0,255,255,255,255,255,255,255,0,0,255,255,

255,255,255,0,0,255,255,0,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,0,255,255,0,0,255,2
55,255,

0,0,0,255,255,0,0,255,255,255,255,255,255,255,255,255,255,255,2
55,

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
,255,255,0,

0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,255,255,0
,0,255,

255,0,255,255,255,0,0,0,0,255,0,0,255,255,255,255,255,255,255,

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
,255,0,0,
  255,255,0,0,255,255,0,0,255,255,0,0,255,0,255,255,0,0,255,

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
,255,0,255,

255,255,255,0,0,255,255,255,0,0,255,255,0,255,255,0,255,255,0,

0,255,0,0,255,255,255,255,255,0,0,255,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  0,255,255,0,0,255,255,255,0,0,255,255,255,0,0,255,255,0,0,
  0,0,255,255,255,0,255,255,0,0,0,0,0,0,0,255,255,255,255,

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,

255,255,255,255,0,0,255,255,0,0,255,255,255,0,0,255,255,255,255
,
  0,0,255,0,0,0,0,255,255,255,0,0,255,255,255,0,0,0,0,
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,

255,255,255,255,0,255,255,255,255,0,0,255,255,255,0,0,0,255,0,

0,255,255,255,255,0,0,255,255,0,0,0,255,255,255,255,0,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,255
,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,0,255,255,0,0,255,255,255,0,0,255,255,2
55,
  0,0,0,0,0,0,255,255,255,255,255,0,0,255,0,0,0,255,255,

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,

0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,

255,255,255,255,255,255,255,255,255,255,255,0,255,255,0,0,255,2
55,255,
  0,0,255,255,255,255,0,0,0,0,0,0,255,255,255,255,0,0,255,

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
,255,255,0,
0,255,255,255,255,0,0,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,0,0,255,255,255,255,0,0,0,2
55,
  255,0,0,0,0,0,0,0,0,255,255,0,0,255,255,0,0,0,0,

255,255,255,0,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,0,0,255,255,0,0,255,255,255,0,0,255,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0
,0,0,
  0,0,0,0,255,255,0,0,0,0,255,255,0,0,0,255,255,0,0,

255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,0,255,255,255,0,0,0,255,255,0,0,0,255,255,0,0,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,0,0,255,255,255
,255,255,
  255,0,0,0,0,0,255,0,0,255,255,255,0,0,255,255,255,255,0,

0,255,255,0,0,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,0,0,255,0,0,255,255,255,0,0,0,0,255,255,0,0,0,

0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,0,
  0,255,255,255,255,255,255,0,0,0,255,255,0,0,255,255,255,0,0,

255,255,255,255,0,0,255,255,255,0,255,255,255,255,255,255,255,2
55,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,0,255,255,0,0,255,255,0,0,255,255,0,0,0,0,

255,255,0,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255
,

255,255,255,255,0,0,255,255,255,255,255,255,255,0,0,255,255,0,0
,

255,255,255,255,0,0,255,0,0,0,0,255,255,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,0,0,255,255,0,0,255,0,0,255
,
  255,0,0,255,0,0,255,255,0,0,255,0,0,0,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,255,255,
  0,0,255,0,0,255,255,255,255,0,0,0,0,0,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,0,0,255,255,255,255,255,0,255,2
55,0,
  0,255,255,0,0,255,0,0,255,255,0,255,255,0,0,255,255,255,0,

0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,2
55,

255,255,0,0,255,255,0,0,0,0,255,255,255,255,255,0,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,255,2
55,255,255,
  255,0,0,255,255,0,0,255,255,0,0,0,0,255,255,0,0,255,255,

0,0,255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,2
55,255,
255,255,0,0,255,255,255,255,0,255,255,255,0,0,0,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,255,255
,255,255,
  0,0,255,255,255,255,0,0,255,255,0,0,255,255,0,0,0,0,255,

255,255,0,255,255,0,0,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,0,0,255,255,255,0,0,0,255,255,255,0
,

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
,255,255,0,
  0,255,255,255,255,255,0,0,255,0,0,0,0,0,255,255,0,0,255,

0,0,0,0,255,255,255,0,0,255,255,255,255,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,255,0,0
,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,
  255,255,255,255,255,0,0,255,255,255,255,0,0,0,0,0,255,0,0,

255,255,0,0,255,255,0,0,0,255,255,255,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,0,

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
,255,0,0,

0,255,255,255,255,255,255,255,255,255,0,0,255,255,255,255,255,0
,0,

255,255,255,255,0,0,255,255,0,0,255,0,0,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,0,0,2
55,

255,255,255,0,0,255,255,255,255,0,0,255,255,0,0,255,255,255,255
,

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
,255,255,0,
  0,0,255,0,0,255,255,255,0,0,0,0,255,255,255,255,255,255,255,

255,255,0,0,255,255,255,255,0,0,255,255,255,255,255,0,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,0,0,0,0,0,255,255,255,0,0,255,255,0,0,0,0,255,255,
255,255,255,255,255,255,255,255,0,0,255,255,255,255,0,0,255,255
,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,

255,255,255,255,255,0,0,0,255,255,255,255,255,255,0,0,255,255,0
,
  0,255,0,0,255,255,255,255,255,255,255,255,255,0,0,0,0,0,255,

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,

255,255,255,255,255,0,255,255,255,255,255,0,0,255,255,255,255,2
55,255,

255,0,0,255,0,0,255,255,0,255,255,255,255,255,255,255,255,0,0,

0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,0,0,0,255,255,255,255,0,0,2
55,

255,255,255,255,255,255,0,0,0,0,0,255,255,0,0,255,255,255,255,

255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,0,0,0,0,0,0,0,255,255,255,255,255,255,255,255,255,

255,255,255,255,255,255,255,255,0,0,255,255,255,255,0,0,0,0,255
,
  255,255,255,0,0,0,0,0,0,0,255,255,0,0,0,0,255,255,0,

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,255,255,255,2
55,
  0,0,0,0,0,255,255,255,0,0,0,0,0,255,255,255,255,0,0,

0,255,255,255,255,0,0,255,255,255,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,0,

0,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,255
,255,

255,255,255,255,255,0,0,255,255,0,0,255,255,255,0,0,255,255,255
,

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,
  255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255
,

0,0,255,255,255,255,255,255,255,255,255,0,255,255,255,0,0,255,2
55,
0,0,255,255,255,255,0,0,255,0,0,255,255,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,255,255,255,255,255,255,255,255,

255,255,255,255,255,255,0,255,255,255,255,255,255,255,255,255,0
,0,255,
  255,255,0,0,255,255,0,0,0,0,0,0,0,255,255,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,255,255,255,

0,255,255,255,255,255,255,255,255,255,255,0,0,255,0,0,0,0,255,
  255,255,0,0,255,255,255,0,0,0,255,0,0,0,0,255,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0
,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,255,255,255,0,

0,0,255,255,255,0,0,255,255,255,255,255,255,255,255,255,0,0,0,

0,0,255,255,255,255,255,0,0,255,255,255,255,0,0,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0
,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

255,255,255,255,0,0,0,255,255,255,255,0,255,255,255,255,255,255
,255,

255,255,255,0,0,255,255,255,255,255,255,255,0,0,255,255,255,255
,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

255,255,0,0,0,255,255,255,255,255,0,255,255,255,255,255,0,0,255
,

255,255,255,255,255,255,255,255,0,0,255,255,255,255,255,0,255,0
,0,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,255,255,255,0,0,0,255,255,255,255,255,0,255,255,255,

255,255,255,0,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0
,

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,
  255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,255,255,255,0,0,255,255,255,0,0,0,255,255,255,255,

255,0,255,255,255,255,255,255,0,0,255,255,255,255,255,255,255,2
55,255,

0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,255,255,255,0,255,255,255,255,0,255,255,255,255,0,

0,255,255,255,255,255,255,0,255,255,255,255,255,255,0,255,255,2
55,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,255,255,255,255,0,255,255,255,255,0,

255,255,255,255,0,255,255,255,255,255,255,255,0,255,255,255,255
,255,255,

0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,2
55,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,255,255,0,0,0,255,255,255,255,0,0,

255,255,255,255,0,0,255,255,255,0,255,255,255,255,255,255,255,0
,255,

0,255,255,255,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,255,255,255,0,0,0,255,

255,255,255,0,0,0,255,255,255,0,0,255,255,255,255,255,255,255,2
55,

255,255,255,0,255,0,0,255,0,0,0,255,255,255,255,255,255,255,255
,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,255,255,255,0,0,255,255,

255,255,0,0,255,255,255,255,0,0,0,255,255,255,255,0,255,255,255
,

255,255,255,255,0,255,255,255,255,255,0,0,255,255,255,255,255,2
55,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,
  255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,255,255,

255,0,0,255,255,255,255,0,0,255,255,255,255,0,0,0,255,255,255,

255,0,0,255,255,255,255,255,255,0,255,255,255,255,255,255,255,2
55,255,

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255
,255,255,255,

255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,255,255,255,255,

0,0,255,255,255,255,0,0,255,255,255,255,255,255,255,255,255,255
,0,

0,0,255,255,255,255,0,0,255,255,255,255,255,255,0,0,255,255,255
};
```

Table showing the image processing of MPI and original



Running time	original	MPI
1st time	0.000474879	0.000377047
2nd time	0.000138514	0.000692624
3rd time	0.000524241	0.000601835
4th time	0.000520217	0.000242358
5th time	0.00055551	0.000752474
6th time	0.000644507	0.000757024
7th time	0.000528481	0.000253215
8th time	0.000523979	0.000674212
9th time	0.000586122	0.000074541
10th time	0.000529823	0.000665077
Mean running time	0.000502627	0.000509041

Bar chart showing the running time variation of original and MPI of image processing



Usually MPI is faster comparatively to other in this case it takes more time than the original. The mean time of MPI is slightly more than MPI this is because as the image is small its pixel are must be divided and passed to the different core to be processed which can create load in a core. Which can result time consuming time consuming. And sending and receiving form the cores can cause communication overheat so in this case MPI takes more time than original.

3.3 Linear Regression

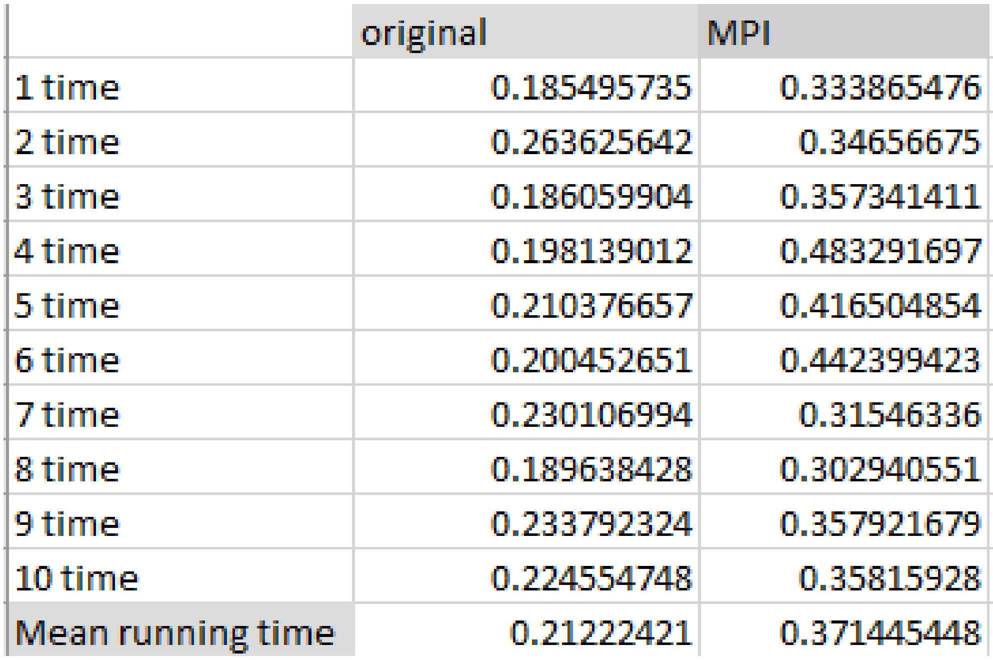
```
#include <stdio.h>
#include <math.h>
#include <time.h>
#include <mpi.h>
#include <malloc.h>
#include <sys/stat.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

/**************************************************************
****************
 * This program takes an initial estimate of m and c and finds
the associated
 * rms error. It is then as a base to generate and evaluate 8
new estimates,
 * which are steps in different directions in m-c space. The
best estimate is
 * then used as the base for another iteration of "generate and
evaluate". This
 * continues until none of the new estimates are better than
the base. This is
 * a gradient search for a minimum in mc-space.
 *
 * To compile:
 *   cc -o lr_coursework_53 lr_coursework_53.c -lm
 *
 * To run:
 *   ./lr_coursework_53
 *
 * Dr Kevan Buckley, University of Wolverhampton, 2018

***************************************************************
**************/

typedef struct point_t
{
   double a;
   double b;
} point_t;

int n_data = 1000;
point_t data[];
```

```
double residual_error (double a, double b, double m, double c)
{
   double e = (m * a) + c - b;
   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].a, data[i].b, 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;
   clock_gettime(CLOCK_MONOTONIC, &start);

   int rank, size;
   int i;
   double bm = 1.3;
   double bc = 10;
   double be;
   double dm[8];
   double dc[8];
   double e[8];
   double step = 0.01;
   double best_error = 999999999;
   int best_error_i;
   int minimum_found = 0;
   double pError = 0;
   double baseMC[2];

   double om[] = { 0, 1, 1, 1, 0, -1, -1, -1 };
   double oc[] = { 1, 1, 0, -1, -1, -1, 0, 1 };


   MPI_Init (NULL, NULL);
   MPI_Comm_size (MPI_COMM_WORLD, &size);
   MPI_Comm_rank (MPI_COMM_WORLD, &rank);

   be = rms_error (bm, bc);

   if (size != 9)
   {
      if (rank == 0)
      {
         printf
            ("This program is needs to run with exactly 9 
processes.\n");
         return 0;
      }
   }

   while (!minimum_found)
   {

      if (rank != 0)
      {
         i = rank - 1;
         dm[i] = bm + (om[i] * step);
         dc[i] = bc + (oc[i] * step);
         pError = rms_error (dm[i], dc[i]);

         MPI_Send (&pError, 1, MPI_DOUBLE, 0, 0, 
MPI_COMM_WORLD);
         MPI_Send (&dm[i], 1, MPI_DOUBLE, 0, 0, 
MPI_COMM_WORLD);
```

```
         MPI_Send (&dc[i], 1, MPI_DOUBLE, 0, 0, 
MPI_COMM_WORLD);


         MPI_Recv (&bm, 1, MPI_DOUBLE, 0, 0, MPI_COMM_WORLD, 
MPI_STATUS_IGNORE);
         MPI_Recv (&bc, 1, MPI_DOUBLE, 0, 0, MPI_COMM_WORLD, 
MPI_STATUS_IGNORE);
         MPI_Recv (&minimum_found, 1, MPI_INT, 0, 0, 
MPI_COMM_WORLD, MPI_STATUS_IGNORE);
      }
      else
      {
         for (i = 1; i < size; i++)
         {
            MPI_Recv (&pError, 1, MPI_DOUBLE, i, 0, 
MPI_COMM_WORLD, MPI_STATUS_IGNORE);
            MPI_Recv (&dm[i-1], 1, MPI_DOUBLE, i, 0, 
MPI_COMM_WORLD, MPI_STATUS_IGNORE);
            MPI_Recv (&dc[i-1], 1, MPI_DOUBLE, i, 0, 
MPI_COMM_WORLD, MPI_STATUS_IGNORE);
            if (pError < best_error)
            {
               best_error = pError;
               best_error_i = i - 1;

            }
         }
         // printf ("best m,c is %lf,%lf with error %lf in 
direction %d\n",
         // dm[best_error_i], dc[best_error_i], best_error, 
best_error_i);
         if (best_error < be)
         {
            be = best_error;
            bm = dm[best_error_i];
            bc = dc[best_error_i];
         }
         else
         {
            minimum_found = 1;
         }

         for (i = 1; i < size; i++)
         {
            MPI_Send (&bm, 1, MPI_DOUBLE, i, 0, 
MPI_COMM_WORLD);
```

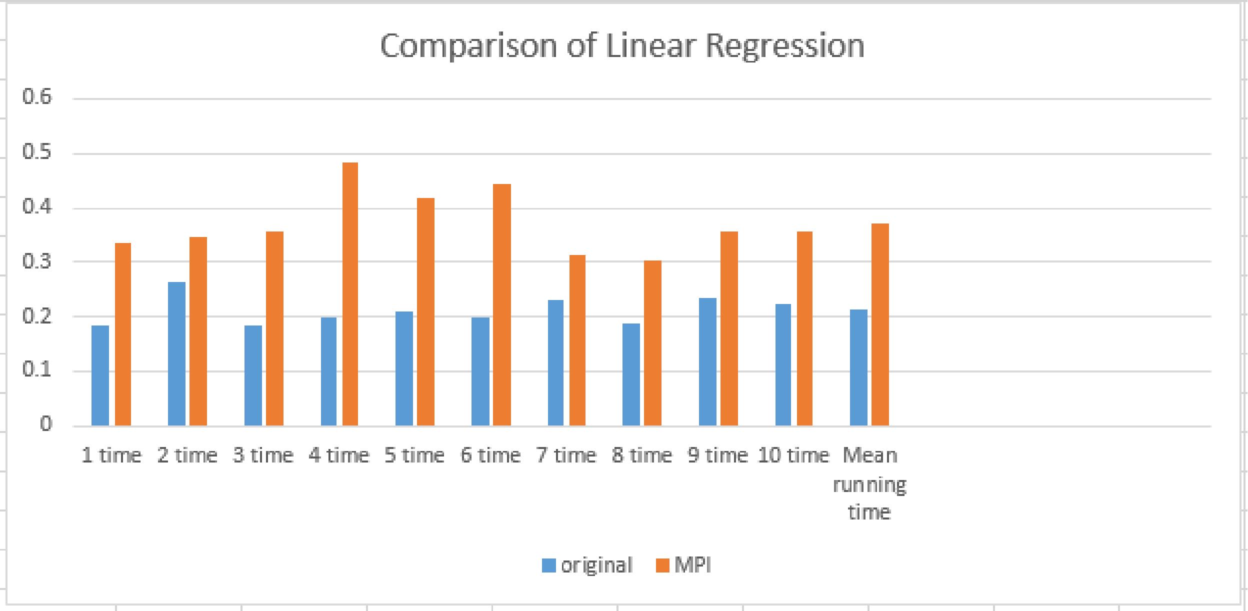
```
            MPI_Send (&bc, 1, MPI_DOUBLE, i, 0,
MPI_COMM_WORLD);
            MPI_Send (&minimum_found, 1, MPI_INT, i, 0,
MPI_COMM_WORLD);
         }
      }
   }

   if(rank==0) {
      printf ("minimum m,c is %lf,%lf with error %lf\n", bm,
bc, be);
      clock_gettime(CLOCK_MONOTONIC, &finish);
      time_difference(&start, &finish, &time_elapsed);
      printf("Time elapsed was %lldns or %0.9lfs\n",
time_elapsed,
         (time_elapsed/1.0e9));
   }

   MPI_Finalize();
   return 0;
}

point_t data[] = {
  {77.96,128.03},{65.68,95.52},{82.85,133.97},{87.44,126.26},
  {65.25,115.66},{65.81,109.29},{78.07,137.65},{73.81,119.15},
  {83.67,130.20},{83.29,130.15},{65.45,92.82},{14.28,69.92},
  {59.72,113.57},{51.21,105.39},{18.33,70.24},{17.57,42.36},
  {19.28,86.46},{61.11,110.70},{ 3.06,32.92},{66.61,132.11},
  {50.28,86.85},{16.14,56.57},{71.80,131.85},{76.53,143.87},
  {22.16,49.15},{30.64,87.42},{58.97,98.66},{10.46,46.54},
  {65.53,106.21},{91.93,143.24},{73.47,128.02},{20.95,64.18},
  {62.50,110.38},{18.18,54.18},{21.48,57.86},{28.50,60.11},
  { 7.54,48.39},{27.41,78.05},{49.86,91.89},{27.28,85.08},
  {53.79,104.68},{21.85,57.80},{35.55,66.67},{95.93,155.19},
  {57.39,104.01},{42.54,96.87},{67.66,103.79},{82.65,134.83},
  {56.97,100.17},{64.07,112.48},{87.60,146.67},{36.95,90.02},
  {24.78,62.26},{65.78,106.87},{72.22,123.40},{85.91,138.57},
  {22.21,58.65},{23.45,65.71},{34.59,66.36},{40.13,82.01},
  {19.99,73.66},{47.56,101.54},{ 8.38,26.03},{61.23,96.47},
  {52.33,115.23},{61.95,116.68},{84.06,132.97},{47.14,96.40},
  {10.24,56.26},{42.03,87.61},{12.97,52.18},{82.86,150.30},
  {30.40,76.91},{66.49,114.01},{42.05,78.38},{68.17,120.08},
  {91.94,142.49},{66.60,97.85},{38.59,106.14},{67.52,114.10},
  { 0.68,39.30},{ 5.86,47.99},{87.24,138.51},{40.64,79.63},
  {85.67,145.03},{ 1.15,34.90},{57.79,100.48},{28.04,48.02},
  {79.74,149.06},{39.70,84.37},{47.29,113.66},{70.42,141.72},
  {17.33,66.95},{79.96,142.75},{38.66,63.02},{34.16,64.26},
  {82.46,122.55},{39.01,78.63},{36.46,78.62},{28.80,66.12},
  {39.23,75.54},{63.28,97.79},{20.99,74.67},{94.94,136.48},
  {65.50,132.78},{60.50,108.85},{36.39,90.05},{15.92,58.69},
  {56.87,106.49},{79.49,133.03},{65.81,119.26},{61.67,111.30},
  {13.16,62.03},{28.13,76.33},{33.71,71.92},{16.93,67.53},
  { 8.77,48.89},{ 3.03,25.86},{24.15,64.71},{78.16,138.89},
  {66.89,93.13},{70.14,133.04},{69.61,115.60},{60.69,92.18},
  {24.13,40.17},{ 0.62,44.00},{21.36,73.46},{79.55,141.25},
  {37.98,93.03},{ 9.27,51.73},{51.18,88.24},{24.94,78.03},
  {40.98,76.92},{55.01,103.17},{66.86,133.25},{ 6.21,39.08},
  {95.42,145.04},{91.66,159.55},{74.85,129.96},{33.35,82.30},
  {99.98,153.62},{ 7.78,57.71},{43.91,106.75},{85.56,141.85},
  {99.30,154.24},{76.92,135.95},{23.31,71.94},{83.06,124.53},
  {20.73,48.34},{ 7.61,64.85},{ 4.14,62.97},{48.41,93.44},
  {11.08,54.65},{23.22,66.51},{86.51,146.66},{63.93,96.90},
  {62.18,140.01},{54.58,92.40},{14.68,55.65},{74.08,152.67},
  {29.16,78.88},{34.39,70.74},{53.47,111.77},{79.47,115.22},
  {60.26,115.73},{ 4.35,53.66},{44.74,98.91},{36.52,77.68},
  {12.57,46.94},{ 9.35,29.42},{67.14,132.38},{94.48,155.46},
  {60.56,126.66},{82.58,148.39},{40.20,81.60},{97.03,152.57},
  {37.79,88.69},{92.35,131.07},{73.56,141.49},{60.68,89.01},
  {50.87,102.46},{80.10,134.53},{20.10,63.39},{56.11,85.56},
  {17.12,57.21},{43.41,79.14},{66.00,99.99},{55.44,104.05},
  {65.87,112.98},{87.30,140.25},{40.73,84.93},{35.28,76.61},
  {93.55,139.77},{51.48,73.31},{10.10,57.74},{65.11,111.70},
  {16.16,44.28},{62.35,90.61},{ 3.11,21.44},{97.40,152.47},
  {31.24,68.96},{20.00,58.73},{83.50,123.60},{72.10,109.50},
  {18.70,39.72},{80.72,123.39},{71.65,122.57},{47.08,90.29},
  {15.62,61.31},{39.27,88.61},{73.84,133.29},{48.34,93.10},
  {71.71,131.19},{ 4.44,49.00},{88.42,150.90},{34.22,92.45},
  {36.37,63.52},{22.69,60.34},{81.39,132.60},{61.47,104.34},
  {24.19,69.92},{28.64,65.00},{83.10,134.44},{55.87,97.39},
  {91.48,164.21},{86.12,158.13},{18.45,61.30},{70.00,115.55},
  {29.97,70.25},{ 7.80,50.03},{93.35,152.67},{98.77,166.10},
  {76.52,123.74},{76.27,140.22},{79.94,142.14},{ 9.42,26.12},
  {25.14,66.01},{42.29,84.73},{ 7.16,34.35},{34.35,79.77},
  {30.24,60.18},{97.12,158.57},{ 3.77,32.31},{18.59,71.26},
  {23.68,65.65},{77.69,134.56},{74.78,132.51},{38.54,79.86},
  { 1.41,29.22},{34.20,81.76},{46.56,83.08},{21.55,54.54},
  {36.12,80.97},{89.12,143.03},{74.69,139.63},{72.38,134.98},
  {90.52,126.38},{33.45,77.65},{14.45,48.11},{ 2.19,69.22},
  {34.24,68.57},{38.58,69.10},{31.53,76.53},{95.11,149.75},
  {82.77,134.49},{17.54,48.12},{27.36,80.42},{74.70,120.86},
  {18.22,61.57},{23.51,62.04},{43.08,64.64},{37.54,75.77},
  {40.53,96.29},{63.39,122.09},{57.60,116.76},{70.50,133.52},
  {51.09,93.63},{19.81,58.37},{16.62,33.92},{40.21,71.13},
  {83.14,140.67},{34.51,67.66},{51.98,88.01},{57.57,125.27},
  { 7.08,39.25},{41.83,82.22},{95.35,152.79},{27.18,68.47},
  {84.15,140.31},{28.04,72.94},{97.86,152.05},{76.82,124.28},
  {95.52,159.59},{14.04,60.42},{ 6.68,25.03},{46.02,85.90},
  {25.07,67.74},{12.15,59.67},{79.23,130.51},{67.33,124.69},
  {19.51,60.84},{65.77,111.26},{51.20,100.38},{79.65,128.45},
  {41.88,68.63},{97.73,169.94},{96.29,170.47},{69.36,138.89},
  {76.03,140.16},{71.63,129.66},{14.73,73.72},{30.55,69.36},
  {90.98,153.28},{15.44,40.55},{94.05,155.37},{95.58,146.36},
  {30.50,50.53},{40.93,82.23},{53.19,108.18},{49.46,103.72},
  {84.84,146.27},{56.25,92.68},{18.60,69.02},{12.41,57.91},
  { 2.98,46.08},{25.75,63.79},{97.67,138.01},{82.61,133.12},
  {15.31,50.96},{68.86,122.15},{96.44,143.45},{ 5.69,64.11},
  {78.34,119.62},{53.71,99.43},{13.96,37.77},{27.91,61.52},
  {56.45,105.71},{87.10,147.15},{64.17,102.53},{50.38,90.39},
  {58.45,94.56},{97.31,134.92},{93.78,145.98},{79.53,128.34},
  {55.63,107.26},{56.69,90.85},{17.24,58.16},{94.33,138.14},
  {84.76,142.18},{46.67,100.15},{10.99,44.86},{27.50,55.39},
  { 0.45,37.66},{10.59,44.13},{16.45,52.52},{33.86,76.86},
  {94.86,138.84},{91.84,172.00},{42.33,93.16},{74.66,119.60},
  {33.78,87.28},{ 7.40,28.44},{24.19,82.22},{96.11,160.10},
  {74.05,125.18},{71.22,109.69},{97.56,140.56},{ 8.84,47.04},
  {69.56,117.00},{ 8.41,42.56},{34.89,64.19},{85.11,115.74},
  {37.56,58.13},{12.59,48.39},{ 2.68,46.73},{62.78,99.69},
  {67.09,109.26},{87.49,126.32},{79.12,136.70},{ 7.17,57.74},
  {26.72,50.76},{35.61,88.60},{27.05,73.51},{71.28,140.53},
  {32.32,78.74},{ 0.41, 2.12},{79.39,135.08},{57.97,100.72},
  {60.76,116.50},{93.08,128.87},{78.11,120.73},{85.78,136.17},
  {53.83,96.46},{26.24,48.05},{78.45,133.71},{97.53,140.60},
  {70.22,129.62},{68.04,119.83},{57.92,97.63},{59.04,86.68},
  {91.06,140.68},{32.87,67.78},{96.49,155.45},{23.99,63.38},
  {83.65,136.74},{73.11,125.63},{12.25,55.87},{79.42,121.84},
  {57.58,111.96},{20.97,68.56},{70.31,136.14},{28.32,78.99},
  {32.13,84.83},{43.86,94.95},{58.82,95.92},{ 2.14,49.54},
  {32.16,76.00},{76.61,126.87},{86.98,117.12},{66.97,108.44},
  {69.58,124.78},{ 7.97,40.98},{86.41,129.47},{17.79,38.59},
  {25.39,62.60},{50.76,90.81},{59.80,104.48},{38.07,70.55},
  {35.73,85.25},{17.46,51.93},{71.16,124.51},{71.80,115.02},
  { 1.62,16.54},{91.26,155.89},{ 3.06,50.64},{29.66,64.84},
  { 7.09,53.98},{49.85,109.93},{14.36,45.56},{24.68,53.75},
  {51.44,105.05},{25.25,65.94},{40.55,78.79},{26.87,72.33},
  {27.84,74.47},{40.26,85.74},{62.00,110.62},{10.52,40.68},
  {40.51,76.33},{ 4.88,52.94},{28.29,81.65},{27.36,83.38},
  {79.84,125.13},{26.59,50.55},{51.70,108.24},{49.35,90.31},
  {32.62,69.09},{15.94,45.41},{69.23,121.30},{ 7.46,45.89},
  {84.07,129.79},{90.25,124.15},{47.88,79.40},{96.61,163.10},
  {15.78,73.16},{80.96,131.56},{31.45,62.49},{ 9.29,50.43},
  {35.18,84.87},{30.50,76.96},{89.17,143.70},{90.85,146.92},
  {59.81,96.39},{25.35,70.99},{ 5.98,55.09},{37.00,74.37},
  {80.19,144.98},{94.28,135.99},{86.46,150.28},{34.03,75.79},
  { 4.75,34.83},{70.46,110.49},{94.34,167.80},{70.80,115.88},
  {16.07,57.87},{68.62,112.90},{95.65,146.35},{ 1.02,57.34},
  { 1.78,41.23},{21.44,60.15},{73.24,114.19},{44.80,89.07},
  {84.42,139.57},{98.01,138.74},{ 9.17,33.87},{87.40,133.66},
  {52.38,86.57},{61.45,112.93},{65.82,127.01},{83.91,158.51},
  {93.63,158.06},{61.78,120.56},{24.30,59.86},{28.54,80.27},
  {95.60,143.52},{61.49,117.24},{10.29,60.58},{11.44,50.16},
  {51.35,108.25},{ 7.62,50.81},{69.37,114.61},{50.44,97.38},
  {11.35,39.19},{17.17,43.10},{61.58,132.61},{28.84,69.42},
  {30.47,70.96},{98.22,151.09},{86.43,136.81},{53.19,118.09},
  {95.29,155.91},{98.84,148.83},{ 2.22,26.13},{47.59,85.31},
  {73.84,109.83},{17.56,47.04},{26.58,73.75},{56.82,105.65},
  {79.98,134.87},{68.84,112.36},{88.87,142.96},{62.43,119.22},
  {76.15,135.84},{43.70,103.87},{16.29,58.86},{53.38,96.60},
  {15.83,57.27},{51.27,120.72},{47.59,64.87},{71.91,129.93},
  {98.51,152.48},{71.10,109.44},{77.99,127.90},{23.87,62.56},
  { 8.34,51.65},{79.64,132.34},{31.04,72.40},{44.82,81.69},
  {59.45,109.71},{84.15,130.99},{82.20,127.12},{31.22,83.20},
  {86.18,141.01},{75.15,134.96},{44.27,95.34},{95.97,144.24},
  {14.43,49.55},{34.57,82.98},{63.67,104.58},{66.22,95.67},
  {88.43,134.55},{63.52,123.11},{77.81,136.16},{ 8.71,34.79},
  {29.07,60.32},{19.49,62.23},{ 9.98,58.80},{78.34,117.98},
  {38.96,76.73},{55.21,87.60},{ 5.69,48.63},{74.40,128.08},
  {61.88,107.77},{65.91,111.44},{78.58,122.76},{39.78,91.82},
  {74.49,116.43},{45.47,99.10},{45.39,95.83},{76.30,120.24},
  {10.79,42.01},{52.12,85.62},{62.14,114.29},{77.15,134.46},
  {78.85,143.47},{22.68,53.16},{83.14,126.70},{ 7.41,29.99},
  {82.26,132.70},{72.14,126.79},{99.55,147.42},{85.72,134.53},
  {64.30,139.77},{38.81,51.80},{40.39,87.73},{25.10,72.42},
  {25.08,66.21},{ 5.81,56.34},{14.18,41.56},{94.84,167.41},
  { 8.54,69.86},{63.85,112.79},{57.40,109.06},{21.58,42.54},
  {43.64,78.24},{77.65,141.23},{81.50,142.13},{67.89,111.36},
  {72.87,120.96},{27.00,62.02},{61.38,127.54},{83.46,136.99},
  {54.76,108.99},{15.45,47.88},{60.05,100.26},{22.10,72.44},
  {28.35,65.66},{96.71,151.35},{86.94,116.64},{65.73,130.87},
  {85.94,126.58},{38.88,96.56},{69.21,127.86},{43.69,61.97},
  {69.87,113.70},{82.25,150.00},{94.99,151.04},{25.88,83.24},
  {81.60,115.26},{73.98,117.58},{43.63,88.14},{70.31,100.08},
  {55.26,100.57},{25.59,71.33},{34.63,76.64},{ 2.50,41.37},
  {54.69,95.27},{78.95,129.89},{67.28,112.36},{89.06,136.95},
  {64.28,113.88},{85.81,145.44},{25.24,57.88},{80.48,135.92},
  {92.34,143.08},{46.56,90.96},{88.74,134.15},{53.17,104.03},
  {31.02,59.35},{ 5.52,52.83},{80.57,120.09},{98.37,169.48},
  {12.81,29.56},{21.78,49.52},{42.46,97.45},{75.97,122.99},
  {32.01,99.32},{15.05,60.83},{75.69,123.02},{36.09,68.64},
  {57.21,106.14},{91.60,134.65},{73.77,118.23},{ 0.03,39.93},
  {55.27,107.03},{52.33,89.48},{53.54,107.47},{49.96,74.23},
  {37.17,74.84},{84.24,130.66},{14.91,60.54},{65.04,130.72},
  { 9.44,41.64},{81.34,129.14},{71.28,100.66},{43.43,93.58},
  {75.01,113.17},{88.82,151.61},{29.93,71.33},{65.46,116.24},
  {22.97,61.19},{51.00,96.99},{60.59,109.94},{ 7.00,41.67},
  {25.77,71.27},{75.31,124.33},{ 4.36,40.75},{87.14,156.88},
  {17.37,61.50},{46.75,82.19},{13.05,42.65},{78.95,128.33},
  {73.10,134.15},{74.36,135.48},{98.65,146.97},{88.12,132.71},
  {28.60,82.30},{15.10,49.96},{49.63,97.09},{87.28,135.25},
  {61.89,111.58},{34.90,93.67},{49.99,100.83},{94.71,170.51},
  {25.88,73.00},{61.84,110.97},{55.36,102.57},{67.98,132.12},
  {21.37,64.84},{13.94,48.72},{43.71,90.80},{91.48,146.34},
  {31.14,59.33},{10.31,48.50},{94.61,149.76},{71.41,120.34},
  {21.14,64.98},{32.06,73.94},{66.20,105.95},{33.15,61.90},
  {86.19,131.68},{83.78,114.83},{54.33,96.58},{33.55,80.32},
  {96.60,170.21},{11.97,28.29},{63.59,124.56},{55.26,100.67},
  {63.14,123.76},{32.40,83.07},{18.12,63.35},{87.59,126.65},
  {50.10,97.98},{65.93,124.39},{62.49,103.95},{ 6.82,27.69},
  {46.80,85.68},{62.95,104.80},{45.74,78.09},{61.82,126.51},
  {92.80,134.13},{57.48,114.43},{74.40,109.61},{60.92,108.78},
  {30.89,79.66},{94.62,139.00},{49.83,98.73},{ 6.11,31.51},
  {45.54,88.00},{19.50,42.96},{25.52,66.51},{73.81,138.08},
  {29.34,88.98},{ 8.18,27.09},{30.62,65.44},{46.31,113.87},
  {41.02,86.84},{90.96,164.15},{22.01,45.41},{ 4.89,39.78},
  {95.35,142.65},{71.94,110.75},{63.35,105.06},{ 4.09,40.40},
  {88.54,139.61},{62.07,112.54},{27.70,56.09},{76.66,137.20},
  {83.12,143.20},{67.15,120.22},{60.00,99.28},{87.13,145.36},
  {94.71,148.17},{25.37,60.43},{64.45,125.47},{41.69,82.97},
  {39.60,86.26},{23.70,59.91},{95.30,155.36},{56.25,96.91},
  {10.09,46.74},{ 0.76,40.83},{59.30,94.67},{94.01,140.36},
  {88.44,141.46},{50.91,92.29},{42.26,99.49},{31.75,63.82},
  {48.52,104.01},{ 5.91,42.20},{80.60,128.78},{25.14,63.22},
  {74.28,124.28},{15.63,74.24},{97.86,149.97},{79.77,140.75},
  {65.69,118.02},{73.56,134.34},{17.85,53.07},{63.86,91.88},
  {13.97,53.39},{32.44,72.72},{ 8.17,56.60},{58.57,116.54},
  {37.35,65.08},{98.78,158.02},{70.98,130.54},{ 6.81,35.93},
  {85.39,160.17},{34.97,80.61},{ 2.64,43.13},{56.77,104.59},
  {81.91,112.09},{31.48,60.20},{34.81,61.30},{11.12,49.56},
  {71.51,128.10},{73.49,124.35},{72.99,112.97},{31.50,83.11},
  {99.98,147.74},{ 2.81,53.17},{42.82,98.70},{59.16,100.75},
  {23.89,72.61},{81.97,159.88},{46.85,94.22},{36.55,93.76},
  {64.96,95.23},{15.11,53.48},{65.91,113.75},{69.19,107.31},
  {28.06,63.30},{58.54,114.26},{89.15,153.48},{42.06,77.31},
  {40.50,76.81},{86.00,146.02},{ 3.20,48.79},{58.69,97.33},
  {35.28,78.08},{ 9.10,46.61},{25.91,66.15},{57.01,103.41},
  {14.91,73.97},{96.76,161.54},{99.67,149.40},{72.54,137.18},
  {30.50,74.61},{42.00,93.95},{59.93,109.89},{66.51,120.16},
  {80.06,138.60},{10.36,51.63},{ 1.60,27.94},{30.78,62.07},
  {66.55,102.64},{61.32,120.77},{91.03,150.60},{53.45,99.29},
  {42.37,78.63},{62.46,111.62},{66.04,112.54},{93.06,151.16},
  {51.95,88.51},{43.30,113.42},{64.13,99.00},{45.53,94.25},
  {39.47,87.77},{29.37,80.52},{92.61,162.12},{21.69,47.65},
  { 1.05,64.71},{40.01,92.37},{97.62,155.20},{70.10,106.73},
  {50.52,80.33},{11.96,51.24},{26.52,76.52},{77.52,132.84},
  {30.52,78.72},{30.52,78.19},{38.11,88.26},{76.86,128.87},
  {28.28,56.37},{30.49,65.71},{59.67,108.95},{89.64,157.43},
  {30.25,81.36},{22.29,69.28},{35.55,84.75},{68.06,113.95},
  {51.60,84.29},{10.09,43.39},{76.55,134.77},{71.33,115.11},
  {51.60,75.80},{54.79,109.48},{96.69,155.10},{14.77,49.52},
  {95.02,148.05},{96.72,141.18},{63.72,98.83},{91.93,140.47},
  {64.34,123.35},{88.86,137.02},{64.18,98.25},{70.04,110.77},
  {94.17,140.07},{75.24,124.76},{44.64,98.74},{87.41,153.95},
  {92.46,144.38},{19.06,66.13},{57.71,112.93},{ 7.45,39.86},
  {27.55,73.56},{56.19,108.21},{ 6.01,40.37},{62.74,89.47},
  {16.17,59.02},{ 8.79,30.74},{83.08,130.52},{24.23,54.52},
  {85.16,149.10},{24.81,90.03},{25.92,54.58},{54.33,82.03},
  {63.90,102.12},{68.66,100.77},{ 5.95,56.49},{42.26,93.76},
  {31.60,90.12},{44.62,89.67},{89.70,139.94},{24.77,52.46},
  {74.16,113.75},{85.36,142.27},{ 2.84,76.16},{91.59,150.81},
  {70.65,125.99},{26.70,76.22},{95.56,152.00},{ 0.44,27.70},
  {20.27,67.29},{ 5.23,37.86},{10.10,40.67},{74.38,130.22},
  {89.36,158.51},{ 4.21,42.73},{42.73,69.67},{72.56,111.50},
  {53.35,114.22},{28.76,96.68},{61.84,110.08},{16.56,62.79},
  {83.69,137.28},{48.91,86.74},{32.35,65.27},{29.44,75.42},
  {30.65,77.48},{85.56,144.14},{90.86,150.13},{81.44,126.15},
  {47.80,89.15},{13.73,41.35},{25.43,79.82},{58.07,92.59},
  {22.91,42.48},{49.14,87.71},{55.98,114.64},{ 8.82,45.15},
  {90.55,153.99},{81.55,138.12},{82.55,136.84},{51.00,101.17},
  {27.58,74.60},{37.31,77.27},{ 1.12,44.83},{88.58,143.95},
  {22.77,75.64},{97.08,157.64},{66.49,108.31},{98.86,156.70},
  {45.64,88.45},{89.75,139.02},{30.57,69.62},{36.48,85.01},
  {98.72,154.40},{30.12,76.32},{73.34,117.76},{16.37,52.48},
  {69.14,134.69},{98.21,174.31},{80.43,120.96},{56.01,100.03},
  { 1.48,43.90},{30.68,56.24},{65.36,121.74},{ 4.45,30.83}
};
```

Table showing the running time for the original and MPI version of linear regression



	original	MPI
1 time	0.185495735	0.333865476
2 time	0.263625642	0.34656675
3 time	0.186059904	0.357341411
4 time	0.198139012	0.483291697
5 time	0.210376657	0.416504854
6 time	0.200452651	0.442399423
7 time	0.230106994	0.31546336
8 time	0.189638428	0.302940551
9 time	0.233792324	0.357921679
10 time	0.224554748	0.35815928
Mean running time	0.21222421	0.371445448

Bar chart showing the running time of original and MPI



By the above table and bar chart it shows the time taken by the linear regression is more than the original file. MPI is actually faster comparative to other. MPI is designed to process to run in to different PC so that the execution time is less. In the above data the running time of original is 0.2122 and the MPI is 0.37144. It was expected to run in 9 different PCs where 1 PC used as a master PC to send and receive message. But in this case the programming is done in a same PC which can cause the overheating of a cores which can takes time. So in this case MPI takes more time than the original one.

4. Verbose Repository Log

```
commit 173b3994f618391657f45d11bdeaceea8b9b0745
Author: SavinGhatani <sa.bin.exe@gmail.com>
Date:   Sat Jan 5 23:35:39 2019 +0545

    journal is modifiedfinal time

commit fc25be5f70029fc2831a96bf874769929cd22c75
Author: SavinGhatani <sa.bin.exe@gmail.com>
Date:   Sat Jan 5 22:24:43 2019 +0545

    new managed file is added and the journal is modified

commit 1f87c91cda9a2761be70470ce998ebfe18def1fe
Author: SavinGhatani <sa.bin.exe@gmail.com>
Date:   Sat Jan 5 20:17:03 2019 +0545

    lr of cuda is added

commit 1d59aefb00ebe3af56fbad6382f05241dca4a8ae
Author: SavinGhatani <sa.bin.exe@gmail.com>
Date:   Sat Jan 5 19:33:42 2019 +0545

    journal is being modified

commit 523f37eed55dcebdcb24a1e0775d13b4d5720ecf
Author: SavinGhatani <sa.bin.exe@gmail.com>
Date:   Sat Jan 5 17:50:14 2019 +0545

    lr from cuda and journal is added journal is incomplete

commit 3524c17e4b29dc4f7e584fef57841fea4226a66d
Author: SavinGhatani <sa.bin.exe@gmail.com>
Date:   Sat Jan 5 16:57:16 2019 +0545

    Linear regresson of MPI is added with SS

commit 01d1dfbc78ab4da6ffdb2212d28f170e9f119fa1
Author: SavinGhatani <sa.bin.exe@gmail.com>
Date:   Sat Jan 5 16:36:10 2019 +0545

    first
```

```
commit 4e0630da16759faccf29a2e7a5e87fc1fae8ac1d
Author: Sabin Ghatani <sa.bin.exe@gmail.com>
Date:   Sat Jan 5 10:15:37 2019 +0000

    Initial commit
```