6CS005 Learning Journal - Semester 1 2018/19
Ashish Shrestha, 1828427

Table of Contents

1 POSIX Threads

1.1 Password Cracking

a. Insert a table of 10 running times and the mean running time.

Calculating running times for the original program.	
	Time for original program.
1st	852.086086160 Sec
2nd	852.851117573 Sec
3rd	866.621660790 Sec
4th	841.839096854 Sec
5th	870.891680057 Sec
6th	825.242112556 Sec
7th	864.673123279 Sec
8th	877.859196113 Sec
9th	871.069347481 Sec
10th	854.746969314 Sec
Total time:	8577.88039018 Sec
Mean Time in sec:	857.788039018 Sec
Mean Time in Min :	14.296467317 Min

From above table, we know that the average time taken to crack password is 14.296467317 minute.

b. Insert a paragraph that hypothesis's how long it would take to run if the number of initials were to be increased to 3. Include your calculations.

```
#include <stdio.h>
#include <string.h>
#include <stdlib.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-3initials CrackAZ99-With-3initials.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-3initials > results.txt

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

char *encrypted_passwords[] = {

"$6$KB$.OnsM0.pEomFoalYh1Tz0da3BPcVeBs6fn3fn6Aq8XMjLMc5aImRqL9hBnjFpCVLi
gSq9dYwAelFBufipopxY1",

"$6$KB$a3I0PZzxK72KmUEF9BwCDgavNTcKF7qMyisIX92DAGRCeiU4eAngEL7TSC/
v.vAix7d0j1jFV3stlwZUohpEo0",

"$6$KB$hn2ozvmpb8pvNu.G7pewcZb6xu.iQBuE9MywKWKPgsALV.gpJnzo8NQqJBs7yvs13
jPV2dficoIfumz04V5uH/",

"$6$KB$o1ZCfk2QE6ENjSr9AViCaiPy6Ln1.pkNEorytYnS5gEAAk4dYSTUomM5YGbXSs6Zg
1ObJ3HiraiyI0ONH1oAb0"
};

/**
 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 w, 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(w='A'; w<='Z'; w++){
  for(x='A'; x<='Z'; x++){
    for(y='A'; y<='Z'; y++){
       for(z=0; z<=99; z++){
        sprintf(plain, "%c%c%c%02d", w, 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);
}

// Calculate the difference between two times. Returns zero on
// success and the time difference through an argument. It will
// be unsuccessful if the start time is after the end 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[]){
  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 or %0.9lfmin\n",
time_elapsed,(time_elapsed/1.0e9),((time_elapsed/1.0e9)/60));


  return 0;
}


/*****************************************************************************/
```

Ans:
From 2 initial number program we know that, it took approximately 14.296 minute time to run program. So, if we increase the number of initials to 3, which means (A-Z) = 26 character, then the time should increase by 26 * 14.296 = 371.696 min = 6.1949 hours.

c. Explain your results of running your 3 initial password cracker with relation to your earlier hypothesis.

Ans:
From above (question number b) estimation it was found that it should take 6.1949 hours to run 3 initial password cracking program but actually it took 337.457580566 min = 5.624293009 hours ≈ 6 hours.

Multi-thread Password Cracking Program.

```
#include <stdlib.h>
#include <stdio.h>
#include <ctype.h>
#include <sys/stat.h>
#include <string.h>
#include <time.h>
#include <pthread.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-multi-threads CrackAZ99-With-Data-multi-
threads.c -lcrypt -pthread

  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-multi-threads > results.txt

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

int n_passwords = 4;
```

```
char *encrypted_passwords[] = {

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

"$6$KB$1vCJQRsKAizZvuTjEqLE./
zeAqZ7AdqzTdx5Tob.bQDRGlXOEVkNhuqMorBSvCp4fP0eoNm2wULs3DRABbrLQ/",

"$6$KB$VKxGV/w3/
XgUswkstexjD72Q5jKKSTyxSfIQlkwtApaQlnYvPfZA4O38d1fc4QvWxDc93T4HzfF/
ksMKgfzFM/",

"$6$KB$xJHmCu7jpWBZlq2AAXZUB5ZF8NcJgZkydHbCyfPwy3nE.djIlSmUxHDLAZdnwz8ys
5L8L9.61NMdYXsyN32r11"
};

/**
 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_block() {
  int i;
  pthread_t th1, th2;

  void *first_block_crack();
  void *second_block_crack();

  for(i=0;i<n_passwords;i<i++) {
    pthread_create(&th1, NULL, first_block_crack, encrypted_passwords[i]
);
    pthread_create(&th2, NULL, second_block_crack,
encrypted_passwords[i]);
```

```
  pthread_join(th1, NULL);
  pthread_join(th2, NULL);
}
}


void *first_block_crack(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++){
        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);
        }
       }
    }
  }
}

void *second_block_crack(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++){
        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;

   clock_gettime(CLOCK_MONOTONIC, &start);


     crack_block();


   clock_gettime(CLOCK_MONOTONIC, &finish);
   time_difference(&start, &finish, &time_elapsed);
   printf("Time elapsed was %lldns or %0.9lfs or %0.9lfmin\n",
time_elapsed,(time_elapsed/1.0e9),((time_elapsed/1.0e9)/60));


   return 0;
}


/*******************************************************************************/
```

d. Write a paragraph that compares the original results with those of your multi-thread password cracker.

Calculating running times for the original program and for multi-thread version.		
	Time for original program.	Time for multi-thread program.
1st	852.086086160 Sec	424.101145135 Sec
2nd	852.851117573 Sec	428.928177644 Sec
3rd	866.621660790 Sec	428.028800344 Sec
4th	841.839096854 Sec	426.398617337 Sec
5th	870.891680057 Sec	432.247245291 Sec
6th	825.242112556 Sec	430.344344965 Sec
7th	864.673123279 Sec	430.50855403 Sec
8th	877.859196113 Sec	431.871656769 Sec
9th	871.069347481 Sec	433.916258052 Sec
10th	854.746969314 Sec	475.311364531 Sec
Total time:	8577.88039018 Sec	4341.656164098 Sec
Mean Time in sec:	857.788039018 Sec	434.1656164098 Sec
Mean Time in Min :	14.296467317 Min	7.23609360683 Min

From above table, we know that the average time taken to crack password by using multi-threading is 7.2360 minute. So, we can say that by using multi-thread in password cracking can reduce time by half or more according to the processor of laptop or desktop.

1.2 Image Processing
a. Insert the image displayed by your program

Image Processing program using multi-threaded edge detector.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <GL/glut.h>
#include <GL/gl.h>
#include <malloc.h>
#include <signal.h>
#include <pthread.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_multi_thread_014
ip_coursework_multi_thread_014.c -lglut -lGL -lm -pthread
  To run this program:
   ./ip_coursework_multi_thread_014

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

```
**********************************************************************
******/
#define width 100
#define height 72

unsigned char image[], results[width * height];
typedef struct arguments_t {
  int start;
  int stride;
  unsigned char *input;
  unsigned char *output;
} arguments_t;

void edge_detect(unsigned char *image,unsigned char *results) {
  pthread_t th1, th2, th3, th4;

  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_image_edges();

  pthread_create(&th1, NULL, detect_image_edges, &t1_arguments);
  pthread_create(&th2, NULL, detect_image_edges, &t2_arguments);
  pthread_create(&th3, NULL, detect_image_edges, &t3_arguments);
  pthread_create(&th4, NULL, detect_image_edges, &t4_arguments);

  pthread_join(th1, NULL);
  pthread_join(th2, NULL);
  pthread_join(th3, NULL);
  pthread_join(th4, NULL);
}
```

```
void *detect_image_edges(arguments_t *args) {
  int i;
  int n_pixels = width * height;
  unsigned char *in=args->input;
  unsigned char *out=args->output;

  for(i=args->start;i<n_pixels;i+=args->stride){
    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);
  //detect_edges(image, results);

  struct timespec start, finish;
  long long int time_elapsed;

  clock_gettime(CLOCK_MONOTONIC, &start);

  edge_detect(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[] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,255,255,255,

255,255,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,

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

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

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

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,
```

```
255,255,255,255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,25
5,

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

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

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

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

255,255,255,255,255,255,255,255,255,255,255,255,0,255,255,255,255,255,25
5,

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

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

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

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

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

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

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

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

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

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,

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,
  255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,
  255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,
  255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,

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

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

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

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

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

```
0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,
  255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,
  255,255,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,
  255,255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,

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

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

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

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

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

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

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

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

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

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

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,
```

```
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,

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

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

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

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

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

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

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

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

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

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

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

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

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,
```

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

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

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

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

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,

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

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

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

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

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

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

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

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,
```

```
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,

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

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

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

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

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

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

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

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

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

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

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

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,
```

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

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

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

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

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

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

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

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

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

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

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

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

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,
```

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

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

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

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

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,
  255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,255,255,

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

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

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

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

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

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,

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

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

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

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

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

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

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

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

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

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

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

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

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

/*****************************************************************************/
```

b. Insert a table that has columns containing running times for the original program and your multi-thread version. Mean running times should be included at the bottom of the columns.

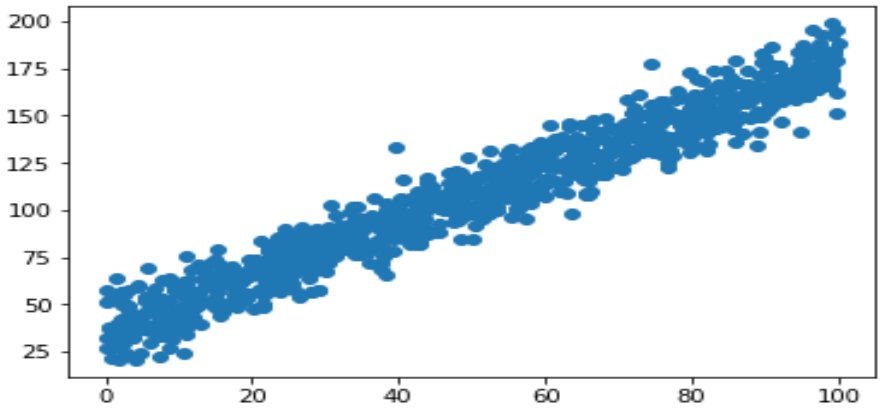
Calculating running times for the original program and for multi-thread version.		
	Time for original program in second.	Time for multi-thread program in second.
1st	0.000244189	0.000276838
2nd	0.000158769	0.000386129
3rd	0.000134126	0.000563675
4th	0.000124196	0.000390902
5th	0.000320527	0.000624942
6th	0.000182322	0.00058715
7th	0.000297749	0.000371939
8th	0.000330665	0.000629257
9th	0.000338829	0.000600857
10th	0.000286728	0.000544998
Total time:	0.0024181	0.004976687
Mean Time:	0.00024181	0.0004976687

c. Insert an explanation of the results presented in the above table.
Ans:

From above table the mean time to run an original program is 0.00024181 sec and for multi-thread program mean time is 0.0004976687 sec. Which shows that the multi-thread program took more time to execute compared to original program because in multi-thread program there are four pixels in parallel which processes each thread. So, it took more time.

1.3 Linear Regression

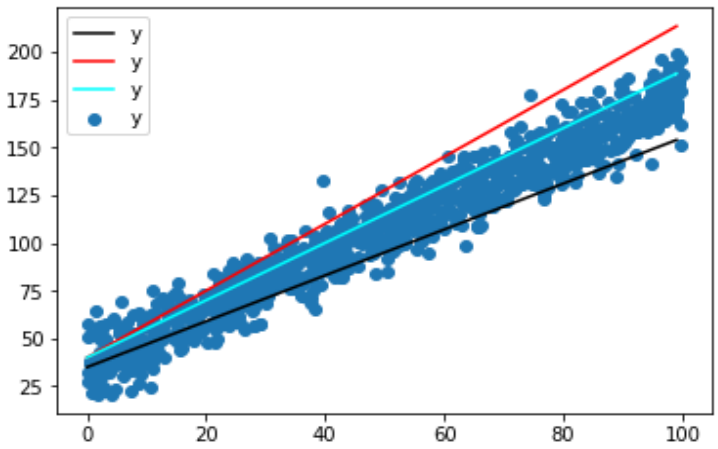
a. Insert a scatter plot of your data.



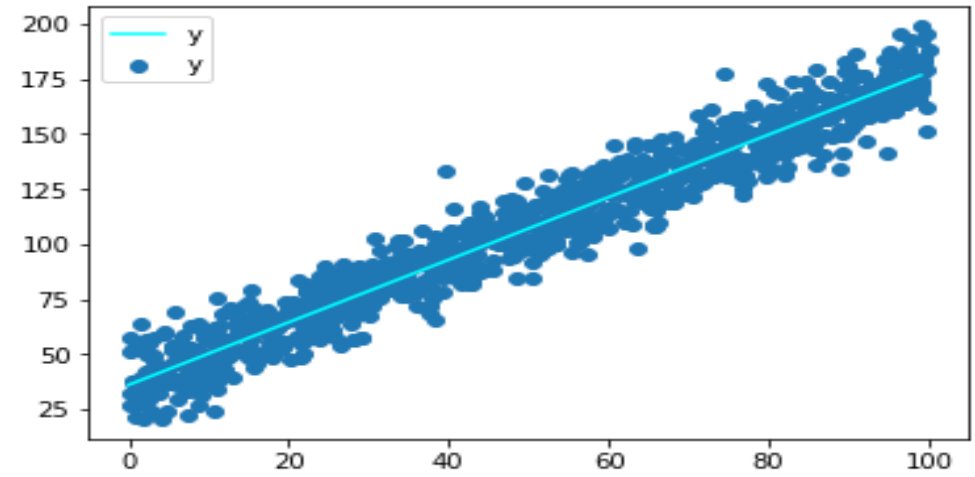
b. Have 3 guesses at the optimum values for m and c and present them in a graph that overlays your data.

The three initial guesses are:

1. m = 1.2 , c = 35
2. m = 1.75 , c = 40
3. m = 1.5 , c = 40



c. Insert a graph that presents your data with the solution overlaid.
My data,
m = 1.42 , c = 36.18



d. Insert a comment that compares your guesses with the solution found.

Linear regression multi-thread program.

```
#include <stdio.h>
#include <math.h>
#include <pthread.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 by using thread.
 *
 * To compile:
 *   cc -o lr_coursework_multi-thread 'lr_coursework _multi-thread.c' -
lm -pthread
 *
```

```
 * To run:
 *   ./lr_coursework_multi-thread
 *
 * Dr Kevan Buckley, University of Wolverhampton, 2018

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



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 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};

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 *function_threads(void *args){
```

```
   int *arg = args;
   int i = *arg;
      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() {
  int i;
  pthread_t th[8];

  be = rms_error(bm, bc);

  struct timespec start, finish;
  long long int time_elapsed;

  clock_gettime(CLOCK_MONOTONIC, &start);

  while(!minimum_found) {
    for(i=0;i<8;i++) {
       pthread_create(&th[i], NULL,(void*)function_threads , &i);
       pthread_join(th[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;
    }
  }

 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));

  return 0;
}

 point_t data[] = {
  {77.23,150.76},{79.80,130.99},{72.07,129.56},{68.54,135.57},
  {83.81,149.30},{85.64,146.39},{76.45,131.33},{84.98,160.16},
  {65.01,144.98},{78.23,138.68},{80.13,138.82},{14.69,60.34},
  {15.27,56.88},{42.63,99.46},{49.53,108.22},{46.79,114.60},
  {74.02,138.49},{68.93,143.70},{29.07,78.24},{22.66,57.23},
  {13.55,60.50},{24.42,89.84},{69.63,141.88},{68.20,121.97},
  {13.52,67.63},{98.18,171.90},{96.84,167.57},{15.31,79.26},
  {81.15,155.58},{ 7.24,42.23},{92.34,171.28},{ 7.59,35.72},
  {86.80,140.15},{94.64,164.89},{55.48,117.90},{94.96,162.89},
  {27.86,70.22},{58.08,132.89},{66.77,128.47},{74.02,138.16},
  { 0.44,37.42},{24.84,61.83},{36.74,93.20},{ 6.16,36.15},
  { 8.67,46.12},{62.84,120.85},{91.14,160.36},{38.19,92.96},
  {38.61,86.03},{ 4.54,41.42},{ 7.66,45.15},{90.28,151.89},
  {66.28,136.36},{14.97,53.68},{99.09,181.15},{ 4.32,37.34},
  {28.45,70.87},{51.71,106.37},{72.04,154.43},{92.30,157.78},
  {43.08,107.99},{68.79,133.45},{21.91,66.64},{ 0.87,21.60},
  {37.19,90.34},{34.04,84.70},{31.02,84.13},{44.61,88.35},
  {14.38,51.01},{65.07,131.80},{25.28,83.50},{57.93,103.46},
  {63.70,133.50},{61.95,135.18},{25.10,71.75},{ 7.30,35.50},
  {27.96,56.29},{56.46,101.30},{64.09,138.41},{84.80,156.70},
  {82.47,148.81},{65.74,115.48},{ 2.20,50.31},{79.73,146.57},
  {99.84,195.89},{75.87,157.39},{99.03,169.23},{ 1.04,37.54},
  {90.77,169.12},{44.44,112.84},{53.09,117.78},{29.20,74.15},
  { 8.14,48.53},{75.40,151.29},{ 0.36,51.74},{44.53,109.50},
  {17.21,69.95},{82.10,153.97},{10.85,75.38},{11.03,37.76},
  {43.62,95.03},{ 4.17,37.63},{42.83,86.76},{94.99,162.04},
  {38.27,103.19},{ 8.92,31.66},{ 5.91,47.62},{72.11,147.98},
  {21.67,50.12},{73.32,150.38},{89.72,162.56},{21.43,63.83},
  {38.87,87.30},{57.55,108.29},{95.32,160.39},{50.51,102.08},
  {40.26,105.93},{ 1.42,64.12},{88.47,165.02},{60.40,117.44},
  {23.16,61.15},{31.47,97.11},{24.66,65.60},{15.71,66.86},
  {53.25,100.04},{25.62,75.47},{65.53,127.89},{20.46,68.16},
  {17.91,67.45},{58.56,110.59},{72.98,137.58},{41.63,81.83},
  {52.35,131.31},{42.70,81.87},{18.82,63.87},{82.37,135.43},
  {50.63,118.03},{26.69,86.74},{38.28,96.89},{79.50,149.35},
  {91.93,176.52},{27.69,87.12},{27.88,71.18},{ 7.50,33.40},
```

```
{53.18,106.68},{42.09,92.69},{12.12,47.44},{55.74,126.06},
{12.73,71.40},{47.39,102.06},{66.37,128.93},{11.89,56.14},
{25.50,61.42},{60.54,145.01},{25.38,80.41},{51.70,119.33},
{78.77,138.83},{54.80,113.53},{67.44,128.89},{21.41,55.06},
{87.89,161.48},{30.29,84.74},{ 4.13,39.77},{73.32,136.32},
{30.56,102.27},{13.64,66.47},{46.86,119.54},{43.94,113.97},
{84.97,148.64},{32.25,79.23},{51.04,117.43},{ 7.29,40.56},
{65.26,144.83},{17.88,52.10},{23.29,75.11},{10.81,24.38},
{ 2.63,24.55},{80.69,147.49},{51.33,107.01},{87.47,164.31},
{55.04,131.68},{88.24,149.22},{55.56,108.54},{99.68,179.53},
{94.33,175.50},{32.96,85.35},{60.91,134.47},{22.09,82.19},
{25.30,78.38},{96.33,195.65},{77.88,138.48},{ 8.50,39.79},
{78.23,148.86},{69.41,135.14},{30.54,77.78},{46.23,102.68},
{89.37,178.40},{29.83,79.88},{71.53,129.01},{94.35,171.86},
{45.41,88.06},{51.08,106.12},{39.44,133.07},{82.63,164.06},
{22.72,64.40},{34.75,89.84},{23.32,63.71},{59.84,129.53},
{21.32,73.28},{95.18,170.16},{ 2.38,31.84},{96.41,173.78},
{31.43,83.49},{99.53,151.57},{73.88,138.10},{80.01,155.21},
{79.83,159.75},{ 2.96,57.46},{25.73,66.61},{79.91,157.52},
{55.94,118.26},{73.29,141.97},{80.19,151.77},{94.29,173.05},
{51.57,109.38},{49.58,128.08},{24.75,58.95},{ 7.00,33.41},
{70.50,130.74},{17.67,55.34},{37.63,68.25},{26.82,91.27},
{79.86,146.65},{10.04,56.42},{71.33,128.72},{64.98,121.51},
{32.35,87.64},{64.04,122.89},{45.95,103.56},{94.60,141.55},
{98.90,177.75},{40.96,86.52},{73.50,142.09},{25.03,62.52},
{52.53,98.14},{48.42,120.10},{87.03,156.45},{36.30,95.35},
{79.61,138.84},{20.66,67.92},{62.61,125.21},{ 2.90,23.00},
{98.68,165.89},{ 6.96,50.75},{39.79,104.21},{79.05,137.55},
{52.58,111.78},{80.68,141.45},{59.72,108.11},{16.10,67.64},
{34.04,88.73},{34.17,101.36},{19.48,64.44},{75.91,151.92},
{11.10,33.79},{77.63,158.42},{36.73,92.59},{75.41,146.27},
{23.88,69.67},{52.25,103.51},{85.09,149.96},{87.95,157.52},
{92.24,157.91},{43.90,87.27},{ 1.94,37.20},{43.01,83.71},
{65.13,116.23},{ 3.01,46.90},{22.24,68.17},{29.85,79.88},
{16.25,63.22},{43.43,99.49},{52.04,104.99},{ 2.45,29.99},
{52.39,110.43},{56.02,122.25},{96.73,169.43},{80.39,161.18},
{75.89,140.06},{83.85,143.76},{85.20,166.70},{ 7.88,35.14},
{80.19,134.04},{85.45,166.57},{39.31,102.76},{46.99,106.95},
{ 8.61,26.49},{75.77,151.27},{ 9.79,41.97},{58.14,121.89},
{95.44,180.82},{88.45,160.82},{55.43,109.57},{33.69,88.65},
{ 6.82,58.64},{10.68,62.46},{76.72,148.78},{61.42,131.59},
{74.44,177.54},{65.51,108.40},{64.69,119.39},{52.46,104.86},
{67.71,125.61},{96.16,182.97},{25.98,75.56},{69.79,122.21},
{80.36,153.17},{46.84,109.51},{75.33,144.70},{27.97,83.34},
{79.32,149.37},{89.43,183.03},{63.62,123.51},{64.08,125.41},
{31.31,75.90},{47.85,107.86},{31.41,87.37},{90.41,157.01},
{19.09,66.83},{85.48,147.29},{ 9.35,45.51},{78.32,145.47},
{40.27,92.98},{54.85,113.99},{36.07,77.94},{76.15,157.52},
{49.07,102.81},{38.90,95.01},{70.96,140.87},{39.03,87.93},
{33.72,84.07},{66.39,128.59},{44.04,108.08},{33.14,82.42},
{ 2.73,37.89},{26.45,77.66},{37.03,91.89},{54.00,124.50},
{63.59,134.17},{16.03,62.07},{ 0.31,32.38},{25.76,72.06},
```

```
{98.95,193.05},{78.40,150.31},{40.62,98.71},{64.81,137.01},
{97.67,169.77},{25.90,85.88},{ 2.96,41.00},{22.47,69.35},
{92.01,169.06},{95.45,164.77},{48.19,105.23},{ 7.76,52.64},
{ 7.21,40.27},{61.00,116.59},{ 2.23,40.79},{ 5.47,52.55},
{71.36,131.37},{68.30,118.58},{65.41,109.22},{45.47,100.67},
{41.21,87.43},{43.77,89.17},{91.86,166.20},{21.43,48.32},
{10.75,36.04},{70.80,139.92},{87.62,173.81},{37.35,95.39},
{75.73,137.61},{89.33,141.27},{15.56,70.91},{37.29,96.21},
{22.13,62.15},{89.49,149.91},{50.07,97.91},{65.22,122.88},
{63.93,130.94},{94.46,168.37},{38.82,88.52},{74.84,154.31},
{54.08,106.91},{62.50,133.84},{77.25,128.70},{16.46,67.64},
{73.45,139.73},{70.40,139.76},{94.44,163.56},{97.97,175.20},
{33.23,89.79},{81.72,146.38},{ 4.70,36.62},{74.68,148.95},
{72.68,140.81},{ 8.87,39.55},{94.28,183.46},{27.16,68.99},
{43.16,92.09},{53.48,101.44},{51.57,124.00},{75.99,156.88},
{94.41,174.23},{72.17,144.36},{71.46,131.88},{42.40,84.41},
{40.17,92.91},{27.88,88.57},{49.06,97.55},{14.86,48.97},
{73.36,140.86},{57.78,108.11},{84.53,158.66},{38.27,100.15},
{12.74,55.85},{14.22,61.66},{65.98,122.01},{63.38,143.72},
{37.68,71.66},{76.78,122.46},{75.28,133.60},{69.72,143.09},
{ 5.52,52.41},{74.35,139.08},{59.99,125.76},{65.64,133.87},
{60.07,107.21},{34.70,85.96},{ 4.61,24.00},{71.23,140.57},
{57.74,121.04},{ 4.23,21.04},{ 5.29,54.13},{25.21,81.66},
{56.30,124.65},{57.39,116.99},{85.46,154.23},{85.10,170.56},
{85.89,146.44},{11.25,63.44},{ 8.29,40.05},{79.22,135.00},
{59.96,123.38},{97.19,163.58},{59.92,116.15},{ 7.36,53.16},
{85.64,143.10},{10.47,41.00},{70.35,125.13},{ 6.02,50.75},
{73.95,139.07},{81.88,144.64},{42.93,104.10},{34.71,76.10},
{ 5.69,52.52},{48.59,84.24},{48.10,94.76},{30.71,88.16},
{68.56,136.98},{22.53,77.31},{16.95,63.73},{27.59,64.07},
{65.14,118.01},{75.78,146.28},{33.86,91.29},{98.93,171.89},
{83.61,167.09},{93.15,164.61},{35.08,95.68},{83.79,149.46},
{52.03,119.04},{23.64,85.42},{18.22,57.37},{74.06,138.99},
{50.25,84.46},{21.37,66.09},{61.15,137.81},{99.61,162.24},
{96.95,173.40},{43.05,95.67},{89.28,163.68},{79.99,158.47},
{59.69,130.55},{ 0.52,38.07},{49.77,111.52},{89.25,164.41},
{25.92,57.36},{81.92,152.67},{61.39,125.63},{29.09,57.77},
{23.31,81.37},{56.87,120.86},{74.59,142.88},{11.80,42.65},
{33.64,101.80},{99.49,184.80},{ 1.85,33.70},{88.99,152.68},
{90.98,186.45},{59.23,136.17},{47.73,120.71},{85.64,158.30},
{30.41,72.27},{20.02,67.23},{68.23,148.39},{15.82,45.42},
{80.29,149.38},{59.94,123.61},{92.82,159.34},{59.14,128.74},
{25.35,77.45},{71.64,145.51},{19.97,70.58},{89.04,171.14},
{53.43,99.51},{67.84,129.83},{54.76,107.18},{15.45,44.48},
{41.74,97.86},{30.68,89.22},{68.71,129.61},{76.58,125.21},
{64.96,129.77},{47.73,102.10},{92.55,169.95},{81.52,161.02},
{51.35,105.53},{56.53,123.58},{51.51,95.62},{59.01,124.59},
{16.29,63.29},{52.83,120.16},{97.68,184.76},{46.83,106.76},
{ 0.02,51.18},{26.76,80.54},{ 2.94,32.88},{87.10,157.17},
{36.90,71.20},{62.84,108.89},{16.87,66.87},{ 5.03,40.84},
{43.45,108.16},{38.04,75.39},{76.77,150.88},{74.60,139.64},
{40.12,91.15},{49.06,100.84},{94.95,184.68},{74.31,134.38},
```

```
{54.58,129.65},{86.95,160.05},{55.69,120.67},{22.24,56.35},
{39.13,90.38},{60.57,131.42},{18.30,53.91},{38.32,95.23},
{55.98,122.71},{13.06,52.26},{38.17,90.53},{95.59,168.64},
{18.62,61.91},{91.96,176.78},{21.70,63.04},{81.25,152.90},
{42.63,81.69},{70.76,137.17},{11.34,51.74},{13.05,39.96},
{48.79,110.92},{86.88,157.80},{27.50,76.94},{88.77,149.45},
{64.06,118.59},{85.27,147.23},{81.98,143.52},{84.58,156.89},
{98.89,174.03},{34.99,82.30},{92.73,172.82},{ 0.75,35.91},
{ 8.61,52.25},{19.10,53.49},{41.42,106.22},{47.89,106.29},
{83.85,148.92},{23.87,57.76},{88.54,165.40},{94.60,173.80},
{19.56,68.73},{54.48,120.93},{67.85,119.02},{24.27,74.92},
{78.18,159.68},{51.03,109.92},{33.42,98.61},{77.69,136.60},
{40.97,84.93},{ 4.75,39.85},{25.01,87.53},{11.52,68.65},
{34.40,78.01},{13.79,65.82},{ 6.27,56.21},{44.00,103.71},
{83.26,143.34},{49.88,101.41},{73.05,140.87},{52.12,104.77},
{66.38,121.93},{87.42,166.29},{ 3.08,43.40},{97.34,169.74},
{31.67,81.37},{66.59,140.38},{45.21,102.87},{41.90,88.79},
{76.28,138.48},{ 1.57,39.23},{42.25,103.21},{43.90,116.91},
{ 3.74,32.26},{ 5.85,46.20},{12.12,51.00},{96.69,164.76},
{10.84,38.65},{63.15,121.09},{62.55,115.31},{42.63,87.47},
{28.87,77.89},{40.33,103.79},{54.09,121.41},{18.21,65.54},
{ 5.68,69.26},{85.82,136.13},{17.38,62.76},{81.55,154.58},
{77.85,163.45},{84.33,163.20},{20.70,57.52},{84.73,173.72},
{86.17,167.83},{10.73,46.04},{71.83,151.69},{29.03,85.40},
{66.88,141.73},{98.13,177.38},{30.52,76.72},{75.67,134.21},
{34.35,77.06},{89.94,181.44},{95.53,176.57},{25.20,76.67},
{40.27,99.43},{33.58,84.87},{63.60,128.21},{90.24,173.53},
{70.81,145.77},{65.90,122.10},{ 5.06,37.13},{72.71,132.60},
{14.98,74.10},{25.67,80.69},{32.04,92.28},{37.97,87.84},
{47.47,93.50},{90.41,154.15},{ 7.68,45.93},{65.44,123.28},
{33.96,76.14},{79.75,154.15},{72.74,161.05},{69.83,135.64},
{29.13,74.88},{81.07,152.23},{ 9.39,57.45},{54.28,115.35},
{58.15,119.42},{47.87,95.57},{29.20,71.41},{23.84,56.54},
{69.13,132.86},{ 9.92,57.01},{84.33,146.61},{69.75,134.84},
{63.33,127.11},{50.30,101.96},{ 7.43,22.18},{21.08,58.29},
{63.42,98.06},{ 6.04,51.75},{88.80,134.59},{10.57,48.82},
{14.86,59.58},{38.78,89.65},{ 8.60,63.72},{ 3.05,49.55},
{79.64,140.40},{61.74,114.00},{47.51,110.09},{37.02,83.39},
{22.32,78.38},{10.91,53.39},{97.40,188.44},{97.81,192.36},
{46.23,100.00},{84.68,160.74},{56.35,103.47},{87.88,148.54},
{11.47,48.34},{74.44,155.79},{24.04,77.24},{85.92,156.32},
{95.47,166.23},{50.51,99.85},{79.61,150.41},{56.65,126.88},
{10.18,46.83},{94.11,165.73},{16.84,68.82},{ 1.32,53.18},
{35.24,82.85},{76.50,133.97},{49.70,115.78},{84.58,154.89},
{ 2.58,54.96},{28.05,56.88},{39.27,84.67},{ 1.26,26.79},
{37.45,81.33},{26.01,83.07},{72.33,139.74},{ 8.92,61.33},
{95.21,187.73},{12.96,57.68},{47.92,99.43},{20.09,47.75},
{74.33,144.70},{36.15,79.94},{83.59,153.50},{ 5.37,46.07},
{98.97,198.88},{85.91,179.15},{95.38,169.13},{85.64,160.16},
{56.06,100.60},{96.89,176.63},{83.93,150.17},{48.80,111.06},
{22.27,79.22},{95.58,173.58},{56.49,124.20},{92.03,146.39},
{66.49,131.04},{39.22,90.96},{17.98,48.44},{ 1.00,33.27},
```

```
{80.50,169.11},{82.88,173.50},{57.92,111.58},{68.90,130.96},
{38.65,87.82},{20.49,73.50},{58.10,106.58},{38.99,88.07},
{41.65,91.96},{82.00,140.65},{33.81,91.57},{23.62,78.38},
{75.76,151.93},{40.42,116.08},{30.84,84.99},{76.73,148.69},
{ 7.81,62.57},{66.31,134.40},{70.76,143.33},{16.91,60.19},
{ 8.16,44.28},{72.52,138.18},{28.13,66.46},{70.97,126.90},
{54.46,112.39},{61.94,110.06},{45.47,92.99},{90.78,156.57},
{66.08,109.59},{12.35,56.66},{17.23,52.58},{71.21,158.27},
{61.22,122.83},{50.62,117.43},{26.52,54.24},{57.46,126.25},
{28.74,90.15},{10.49,48.72},{62.11,138.78},{79.20,140.57},
{73.06,141.61},{19.01,52.29},{15.35,51.63},{98.03,174.79},
{19.43,73.68},{57.18,95.12},{28.71,84.43},{56.17,114.52},
{12.66,50.03},{35.83,78.66},{73.95,137.25},{96.68,169.26},
{13.12,70.59},{72.84,142.80},{80.36,150.96},{29.96,67.92},
{41.51,98.54},{51.54,102.60},{81.96,131.69},{74.14,138.07},
{10.89,57.54},{68.18,127.68},{ 2.46,49.79},{51.99,119.32},
{11.65,55.78},{75.74,132.46},{80.32,146.13},{96.60,165.24},
{66.52,147.54},{18.28,66.38},{66.88,138.55},{66.37,141.96},
{94.67,179.38},{56.35,108.03},{74.57,130.64},{59.22,113.97},
{52.15,104.28},{46.60,107.63},{60.24,126.30},{75.58,129.28},
{47.49,100.09},{96.21,162.75},{20.45,68.77},{39.10,94.83},
{29.27,76.85},{ 8.94,36.23},{97.54,167.51},{13.40,64.23},
{26.77,72.87},{85.95,157.69},{11.91,43.38},{65.80,107.86},
{41.47,89.57},{77.07,131.29},{ 2.25,41.23},{21.29,83.85},
{97.15,164.38},{33.93,89.39},{90.07,158.68},{ 1.90,32.41},
{59.48,126.95},{18.54,62.78},{36.23,97.52},{86.61,160.14},
{28.96,57.15},{15.98,67.67},{58.71,127.68},{45.96,105.75},
{90.77,155.07},{72.45,143.70},{40.83,85.07},{40.26,94.81},
{ 7.74,52.45},{42.50,102.66},{ 6.20,30.04},{40.97,97.68},
{57.38,131.11},{81.44,168.91},{ 2.14,42.62},{47.76,106.77},
{82.71,158.27},{97.74,181.81},{57.87,122.41},{99.50,182.94},
{77.02,151.94},{39.36,86.25},{ 2.21,56.76},{55.28,132.20},
{74.87,132.06},{42.57,109.85},{10.54,46.27},{68.74,123.83},
{99.98,188.43},{35.14,89.08},{79.73,145.31},{95.77,160.03},
{81.03,147.97},{26.37,71.29},{97.06,185.07},{90.27,164.86},
{84.43,162.64},{77.12,138.46},{78.68,141.56},{60.29,129.14},
{11.58,45.54},{47.52,115.01},{50.02,113.78},{95.50,184.29},
{43.90,87.31},{50.31,92.19},{57.40,110.42},{41.27,95.48},
{52.73,122.70},{97.37,181.72},{87.18,156.97},{ 0.27,27.24},
{32.53,78.79},{94.39,163.45},{29.47,74.81},{20.24,73.08},
{ 4.56,59.03},{38.65,90.42},{63.17,145.59},{12.77,49.40},
{14.36,72.37},{71.92,127.46},{35.96,72.29},{54.82,105.88},
{30.31,90.56},{70.58,121.99},{46.53,104.57},{83.62,164.66},
{10.50,43.60},{27.90,68.17},{46.54,100.24},{48.49,115.46},
{53.57,109.34},{19.56,55.62},{81.20,152.35},{55.46,96.62},
{94.19,158.45},{56.27,124.61},{80.51,146.14},{67.37,141.02},
{31.71,89.49},{93.04,161.28},{42.11,91.72},{33.56,93.79},
{ 4.44,60.08},{ 8.21,47.86},{90.51,177.18},{27.26,73.60},
{29.63,85.02},{52.43,112.75},{73.07,130.27},{ 1.79,20.71},
{28.08,81.49},{51.24,117.62},{47.53,109.75},{79.66,172.71},
{15.43,51.60},{39.10,78.55},{48.00,114.36},{ 2.19,39.04},
{16.43,49.62},{47.88,102.29},{24.64,68.34},{57.88,127.64},
```

```
  {34.97,95.54},{ 4.03,42.88},{10.94,60.98},{36.51,105.88},
  {74.16,142.76},{25.29,69.47},{59.63,112.76},{ 2.42,39.41},
  {61.17,115.63},{ 0.02,57.34},{82.24,158.82},{84.95,156.06},
  {38.16,65.51},{29.66,89.43},{84.26,144.66},{37.30,90.54},
  {47.88,108.10},{99.07,171.56},{ 5.35,46.07},{91.88,159.90},
  {66.92,118.74},{51.63,104.98},{86.92,149.65},{91.78,168.41}
};

/*****************************************************************************/
```

e. Insert a table that shows running times for the original and multi-thread versions.

Calculating running times for the original program and for multi-thread version.		
	Time for original program in second.	Time for multi-thread program in second.
1st	0.153255700	0.729684644
2nd	0.154264022	0.730370932
3rd	0.153620850	0.729603310
4th	0.155420145	0.739406606
5th	0.152466764	0.720501911
6th	0.150451341	0.730797864
7th	0.155607034	0.731083206
8th	0.151813714	0.712123609
9th	0.153453562	0.744400154
10th	0.153750012	0.707047138
Total time:	1.534103144	7.275019374
Mean Time:	0.1534103144	0.7275019374

From above table the mean time to run an original program is 0.1534103144 sec and for multi-thread program mean time is 0.7275019374 sec. Which shows that the multi-thread program took more time to execute compared to original program because in multi-thread program each of the evaluations i.e. eights values for m and c, performs on a different thread. So, it took more time than the original one.

2 CUDA
2.1 Password Cracking

CUDA based Password Cracking program.

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

/*****************************************************************************
**
  Compile with:
    nvcc -o CrackAZ99-With-Data-cuda CrackAZ99-With-Data-cuda.cu

  To run program:
    ./CrackAZ99-With-Data-cuda

  Dr Kevan Buckley, University of Wolverhampton, 2018
*****************************************************************************
******/
/*****************************************************************************
  This function returns 1 if the attempt at cracking the password is
  identical to the plain text password string stored in the program.
  Otherwise,it returns 0.
*****************************************************************************/
__device__ int is_a_match(char *attempt) {
  char plain_password[] = "ML4249";

  char *a = attempt;
  char *p = plain_password;

  while(*a == *p) {
    if(*a == '\0') {
      return 1;
    }
    a++;
    p++;
  }
  return 0;
}

/*****************************************************************************
  The kernel function run in 675 threads uses
```

```
  nested loops to generate all possible passwords and test whether they match
  the hidden password.
*****************************************************************************/



__global__ void crack() {
  char alpha[26]=
{'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'};

  char num[10] = {'0','1','2','3','4','5','6','7','8','9'};

  char result[7];
  result[6] = '\0';
  int e, f, g, h;
  for(e=0;e<=9;e++) {
    for(f=0; f<=9; f++) {
      for(g=0; g<=9; g++) {
        for(h=0; h<=9; h++) {
        result[0] = alpha[blockIdx.x];
        result[1] = alpha[threadIdx.x];
        result[2] = num[e];
        result[3] = num[f];
        result[4] = num[g];
        result[5] = num[h];
        if(is_a_match(result)) {
        printf("password found: %s\n", result);
        } else {
            //printf("tried: %s\n", result);
        }
    }
  }
}
}
}

 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;
  clock_gettime(CLOCK_MONOTONIC, &start);

     crack <<<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;
}

/******************************************************************************/
```

a. Insert a table that shows running times for the original, multi-thread and CUDA versions.

Calculating running times for the original program, multi-thread version and CUDA version.			
	Time for original program.	Time for multi-thread program.	Time for CUDA program.
1st	852.086086160 Sec	424.101145135 Sec	0.085126579 Sec
2nd	852.851117573 Sec	428.928177644 Sec	0.086923126 Sec
3rd	866.621660790 Sec	428.028800344 Sec	0.089922869 Sec
4th	841.839096854 Sec	426.398617337 Sec	0.087610839 Sec
5th	870.891680057 Sec	432.247245291 Sec	0.085183689 Sec
6th	825.242112556 Sec	430.344344965 Sec	0.084066495 Sec
7th	864.673123279 Sec	430.50855403 Sec	0.084478937 Sec
8th	877.859196113 Sec	431.871656769 Sec	0.088146979 Sec
9th	871.069347481 Sec	433.916258052 Sec	0.090095827 Sec
10th	854.746969314 Sec	475.311364531 Sec	0.091534554 Sec
Total time:	8577.88039018 Sec	4341.656164098 Sec	0.873089894 Sec
Mean Time in sec:	857.788039018 Sec	434.1656164098 Sec	0.0873089894 Sec
Mean Time in Min :	14.296467317 Min	7.23609360683 Min	0.014551498234 Min

b. Write a short analysis of the results
Ans:

From above table the mean time to run an original program is 14.296467317 Min and for multi-thread program mean time is 7.23609360683 Min and for CUDA program mean time is 0.0873089894 Sec or 0.014551498234 Min. CUDA version crack password faster than other two program because in CUDA password cracking program runs parallel over 675 threads. So, it is fastest among other two program.

2.2 Image Processing

CUDA based image processing program.

```
#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>
/
*****************************************************************************
******
  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:
    nvcc -o ip_coursework_014 'ip_coursework_014 .cu' -lglut -lGL -lm

  To run the program:

  ./ip_coursework_014

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

```
************************************************************************
******/
#define width 100
#define height 72

unsigned char results[width * height];

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

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

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

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

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

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,255,25
5,

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

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

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

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

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

255,255,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,
```

```
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,

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

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

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

255,255,255,255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,25
5,

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

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

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

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

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

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

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

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

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

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

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

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

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,

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,
  255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,
  255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,
  255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,

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

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

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

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

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

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

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,
  255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,
  255,255,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,
  255,255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,

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

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

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

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

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

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

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

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

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,

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

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

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

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

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

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

```
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,

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

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

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

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

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

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

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

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

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

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

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

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

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

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,
  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,
  255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,
  255,255,255,0,255,255,0,0,0,0,255,0,0,255,255,0,0,0,0,
  0,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,

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

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

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

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

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

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

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

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

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

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,
```

```
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,
```

```
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,

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

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

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

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

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

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,
  255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,255,255,

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

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

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

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

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

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,

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

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

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

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

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

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

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

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

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,
```

```
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,

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

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

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

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,
  255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,255,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
};
  __global__ void detect_edges(unsigned char *in, unsigned char *out) {
  unsigned int i = blockIdx.x ;



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

```
    if (x == 0 || y == 0 || x == width - 1 || y == height - 1) {
      out[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);
   unsigned char *d_results;
  unsigned char *d_image;

  struct timespec start, finish;
  long long int time_elapsed;

  clock_gettime(CLOCK_MONOTONIC, &start);

  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);

  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;
}

/*********************************************************************************/
```

b. Insert a table that shows running times for the original, multi-thread and CUDA versions.

Calculating running times for the original program, multi-thread version and CUDA version.			
	Time for original program in second.	Time for multi-thread program in second.	Time for CUDA program in second.
1st	0.000244189	0.000276838	0.075386513
2nd	0.000158769	0.000386129	0.074686924
3rd	0.000134126	0.000563675	0.076007492
4th	0.000124196	0.000390902	0.074578661
5th	0.000320527	0.000624942	0.072643177
6th	0.000182322	0.00058715	0.080916637
7th	0.000297749	0.000371939	0.078102321
8th	0.000330665	0.000629257	0.078525085
9th	0.000338829	0.000600857	0.078103622
10th	0.000286728	0.000544998	0.080501337
Total time:	0.0024181	0.004976687	0.769451769
Mean Time:	0.00024181	0.0004976687	0.0769451769

Write a short analysis of the results

2.3 Linear Regression

Insert a table that shows running times for the original, multi-thread and CUDA versions.

Calculating running times for the original program, multi-thread and CUDA version.			
	Time for original program in second.	Time for multi-thread program in second.	Time for CUDA program in second.
1st	0.153255700	0.729684644	
2nd	0.154264022	0.730370932	
3rd	0.153620850	0.729603310	
4th	0.155420145	0.739406606	
5th	0.152466764	0.720501911	
6th	0.150451341	0.730797864	
7th	0.155607034	0.731083206	
8th	0.151813714	0.712123609	
9th	0.153453562	0.744400154	
10th	0.153750012	0.707047138	
Total time:	1.534103144	7.275019374	
Mean Time:	0.1534103144	0.7275019374	

Write a short analysis of the results

3 MPI
3.1 Password Cracking
a. MPI based 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:
    mpicc -o CrackAZ99-With-Data-MPI CrackAZ99-With-Data-MPI.c
time_diff.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:

    mpiexec -n 3 ./CrackAZ99-With-Data-MPI

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

char *encrypted_passwords[] = {

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

"$6$KB$1vCJQRsKAizZvuTjEqLE./
zeAqZ7AdqzTdx5Tob.bQDRGlXOEVkNhuqMorBSvCp4fP0eoNm2wULs3DRABbrLQ/",

"$6$KB$VKxGV/w3/
XgUswkstexjD72Q5jKKSTyxSfIQlkwtApaQlnYvPfZA4O38d1fc4QvWxDc93T4HzfF/
ksMKgfzFM/",

"$6$KB$xJHmCu7jpWBZlq2AAXZUB5ZF8NcJgZkydHbCyfPwy3nE.djIlSmUxHDLAZdnwz8ys
5L8L9.61NMdYXsyN32r11"
};
```

```
/**
 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 first_block_crack(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 second_block_crack(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 time_elapsed;
  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++){
      first_block_crack(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++){
      second_block_crack(encrypted_passwords[i]);
   }
   }
    }
  }
  MPI_Finalize();

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

  printf("Elapsed Time: %0.9lfs or %0.9lfmin\n", (time_elapsed/1.0e9),
((time_elapsed/1.0e9)/60));


  return 0;
}

/*****************************************************************************/
```

b. Insert a table that shows running times for the original, multi-thread version, CUDA version and MPI versions.

Calculating running times for the original program, multi-thread version, CUDA version and MPI versions.				
	Time for original program.	Time for multi-thread program.	Time for CUDA program.	Time for MPI program.
1st	852.086086160 Sec	424.101145135 Sec	0.085126579 Sec	448.548901729 Sec
2nd	852.851117573 Sec	428.928177644 Sec	0.086923126 Sec	450.614477814 Sec
3rd	866.621660790 Sec	428.028800344 Sec	0.089922869 Sec	457.101931998 Sec
4th	841.839096854 Sec	426.398617337 Sec	0.087610839 Sec	452.628022154 Sec
5th	870.891680057 Sec	432.247245291 Sec	0.085183689 Sec	459.816985690 Sec
6th	825.242112556 Sec	430.344344965 Sec	0.084066495 Sec	450.138745586 Sec
7th	864.673123279 Sec	430.50855403 Sec	0.084478937 Sec	453.292350263 Sec
8th	877.859196113 Sec	431.871656769 Sec	0.088146979 Sec	458.325443892 Sec
9th	871.069347481 Sec	433.916258052 Sec	0.090095827 Sec	455.466813751 Sec
10th	854.746969314 Sec	475.311364531 Sec	0.091534554 Sec	457.408307580 Sec
Total time:	8577.88039018 Sec	4341.656164098 Sec	0.873089894 Sec	4543.341980457 Sec
Mean Time in sec:	857.788039018 Sec	434.1656164098 Sec	0.0873089894 Sec	454.3341980457 Sec
Mean Time in Min :	14.296467317 Min	7.23609360683 Min	0.014551498234 Min	7.572236634095 Min

c. Write a short analysis of the results

3.2 Image Processing

a. MPI based Image Processing Program.

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

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

  To compile adapt the code below wo match your filenames:
    mpicc -o ip_coursework_014_mpi 'ip_coursework_014 _mpi.c' -lglut -
lGL -lm

 To run the program:
 mpiexec -n 4 ./ip_coursework_014_mpi

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

unsigned char image[], results[width * height];
int startIndex, endIndex;
int time_difference(struct timespec *start, struct timespec *finish,
```

```
                    long long int *difference);
void detect_edges(unsigned char *in, unsigned char *out);
static void key_pressed(unsigned char key,int x, int y);
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 !=4){
  printf("This program needs exactly 4 instances.\n");
  exit(-1);


}


 if(rank == 0){
 startIndex = 0;
 endIndex=1799;
```

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

  clock_gettime(CLOCK_MONOTONIC, &start);

  detect_edges(image, results);

  //results, receive
  MPI_Recv(&results[1800],1800,
MPI_UNSIGNED_CHAR,1,0,MPI_COMM_WORLD,MPI_STATUS_IGNORE);
  MPI_Recv(&results[3600],1800,
MPI_UNSIGNED_CHAR,2,0,MPI_COMM_WORLD,MPI_STATUS_IGNORE);
  MPI_Recv(&results[5400],1800,
MPI_UNSIGNED_CHAR,3,0,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();

 }else if(rank==1) {
   startIndex = 1800;
   endIndex = 3599;
   detect_edges(image, results);
   MPI_Send(&results[1800], 1800, MPI_UNSIGNED_CHAR, 0, 0,
MPI_COMM_WORLD);

 }
 else if(rank==2) {
   startIndex = 3600;
   endIndex = 5399;
   detect_edges(image, results);
   MPI_Send(&results[3600], 1800, MPI_UNSIGNED_CHAR, 0, 0,
MPI_COMM_WORLD);

 }
 else if(rank==3) {
   startIndex = 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[] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,0,0,0,0,0,0,0,255,255,255,255,255,255,

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

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

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

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

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,255,25
5,

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

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

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

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,
```

```
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,

255,255,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,

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

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

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

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

255,255,255,255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,25
5,

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

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

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

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

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

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

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

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

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,
```

```
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,

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

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

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

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,

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,
  255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,
  255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,
  255,255,255,255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
  0,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,

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

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

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

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

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

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,
  255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,
  255,255,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,
  255,255,255,255,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,

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

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

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

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

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

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

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,

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

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

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

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

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

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

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

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,

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

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

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,
```

```
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,
  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,
  255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,
  255,255,255,0,255,255,0,0,0,0,255,0,0,255,255,0,0,0,0,
  0,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,

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

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

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

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

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

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

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

```
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,
```

```
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,

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

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

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

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

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

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

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

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

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,
  255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,255,255,

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

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

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

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

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

0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,25
5,

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

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

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

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

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

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

255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
255,
```

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

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

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

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

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

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

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

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

/******************************************************************************/
```

b. Insert a table that shows running times for the original program, multi-thread, CUDA and MPI version.

Calculating running times for the original program, multi-thread, CUDA and MPI version.				
	Time for original program in second.	Time for multi-thread program in second.	Time for CUDA program in second.	Time for MPI program in second.
1^{st}	0.000244189	0.000276838	0.075386513	0.000233578
2^{nd}	0.000158769	0.000386129	0.074686924	0.000235632
3^{rd}	0.000134126	0.000563675	0.076007492	0.000272333
4^{th}	0.000124196	0.000390902	0.074578661	0.000206501
5^{th}	0.000320527	0.000624942	0.072643177	0.000221924
6^{th}	0.000182322	0.00058715	0.080916637	0.000228754
7^{th}	0.000297749	0.000371939	0.078102321	0.000254298
8^{th}	0.000330665	0.000629257	0.078525085	0.000210393
9^{th}	0.000338829	0.000600857	0.078103622	0.000221053
10^{th}	0.000286728	0.000544998	0.080501337	0.000254021
Total time:	0.0024181	0.004976687	0.769451769	0.002338487
Mean Time:	0.00024181	0.0004976687	0.0769451769	0.0002338487

c. Write a short analysis of the results

3.3 Linear Regression

Insert a table that shows running times for the original program, multi-thread, CUDA and MPI version.

Calculating running times for the original program, multi-thread, CUDA and MPI version.				
	Time for original program in second.	Time for multi-thread program in second.	Time for CUDA program in second.	Time for MPI program in second.
1st	0.153255700	0.729684644		
2nd	0.154264022	0.730370932		
3rd	0.153620850	0.729603310		
4th	0.155420145	0.739406606		
5th	0.152466764	0.720501911		
6th	0.150451341	0.730797864		
7th	0.155607034	0.731083206		
8th	0.151813714	0.712123609		
9th	0.153453562	0.744400154		
10th	0.153750012	0.707047138		
Total time:	1.534103144	7.275019374		
Mean Time:	0.1534103144	0.7275019374		

Write a short analysis of the results

4 Verbose Repository Log

Paste your verbose format repository log here. With subversion this can be achieved by the following:
svn update
svn –v log > log.txt

gedit log.txt
Then select, copy and paste the text here