6CS005 Learning Journal - Semester 1 2019/20

Bishrut Neupane 1928726

Table of Contents

1 POSIX Threads

1.1 Password Cracking

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

No.	Nanoseconds	Seconds
1.	508144274294	508.144274294
2.	504983681529	504.983681529
3.	507994538268	507.994538268
4.	508551703597	508.551703597
5.	506553751119	506.553751119
6.	507681018088	507.681018088
7.	505525211300	505.525211300
8.	505730813811	505.730813811
9.	506607479040	506.607479040
10.	505055320878	505.055320878
Meantime in Nanoseconds	506682779191.4	
Meantime in Seconds		

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

If we increase from two initials to three initial the running time will be increases because we have added another loop which takes more time to run the loop.

```
1.  #include <stdio.h>
2.  #include <string.h>
3.  #include <stdlib.h>
4.  #include <crypt.h>
5.  #include <time.h>
6.  #include <math.h>
7.
8.
9.
10. int n_passwords = 4;
11.
12. char *encrypted_passwords[] = {
13.   "$6$KB$t7URKN07IpCV3VKvOB827yLWDjxPsdVJJI9V5yqRbuEmepRDqjGAlapr2G4s6gKTCi82Hc1/NW/JR7v53WyXj/",
14.   "$6$KB$z1NK/wrNzMrVL2aZ4wbqldqS6GA3bgIehxART1ChD05bD5UZl3sXrN7eNsXr1Xbf71UegPUhPicQNwBT6fgyM1",
15.   "$6$KB$DOBoXJMpjGBLxcJePibVbPUiC2GdZgKinYenlPygdimi.TQ5dr.ZTgD01xLHXT6BT7beWXmFL6b/R8FxO/f4z.",
```

```
  "$6$KB$QWkb1xmXZGyP8XQTuLWpatEVtuWr6zw9o7CFDNO6bhGogAEXy9By1Ks6hzk.peTt3cwC.0NBkixzm.ial8nUZ0"
};


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


void crack(char *salt_and_encrypted){
  int x, d, f, b;
  char salt[7];
  char plain[7];
  char *enc;
  int count = 0;

  substr(salt, salt_and_encrypted, 0, 6);

  for(x='A'; x<='Z'; x++){
    for(d='A'; d<='Z'; d++){
      for(f='A'; f<='Z'; f++){
    for(b=0; b<=99; b++){
        sprintf(plain, "%c%c%c%02d", x, d, f, b);
        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);
  int i;

  for(i=0;i<n_passwords;i<i++) {
    crack(encrypted_passwords[i]);
  }

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

  return 0;
}
```

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

```
#189879  CVA78 $6$KB$t7URKN07IpCV3VKvOB827yLWDjxPsdVJJI9V5yqRbuEmepRDqjGAlapr2G4s6gKTCi82Hc1/NW/JR7v53WyXj/

#382384  FRB83 $6$KB$z1NK/wrNzMrVL2aZ4wbqldqS6GA3bgIehxART1ChD05bD5UZl3sXrN7eNsXr1Xbf71UegPUhPicQNwBT6fgyM1

#452405  GSA04 $6$KB$DOBoXJMpjGBLxcJePibVbPUiC2GdZgKinYenlPygdimi.TQ5dr.ZTgD01xLHXT6BT7beWXmFL6b/R8FxO/f4z.

#1287246 TBC45 $6$KB$QWkb1xmXZGyP8XQTuLWpatEVtuWr6zw9o7CFDNO6bhGogAEXy9By1Ks6hzk.peTt3cwC.0NBkixzm.ial8nUZ0


Time elapsed was 13134120445358ns or 13134.120445358s
```

Paste your source code for your multithread password cracker here

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


int n_passwords = 4;

char *encrypted_passwords[] = {

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

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

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

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

void substr(char *dest, char *src, int start, int length){
  memcpy(dest, src + start, length);
  *(dest + length) = '\0';
}
void pass()
{
  int i;
pthread_t thread_1, thread_2;

    void *kernel_function_1();
    void *kernel_function_2();
for(i=0;i<n_passwords;i<i++) {


    pthread_create(&thread_1, NULL,kernel_function_1, encrypted_passwords[i]);
    pthread_create(&thread_2, NULL,kernel_function_2, encrypted_passwords[i]);

    pthread_join(thread_1, NULL);
    pthread_join(thread_2, NULL);
 }
}
```

```
43.
44. void *kernel_function_1(void *salt_and_encrypted){
45.   int b, i, s;
46.   char salt[7];
47.   char plain[7];
48.   char *enc;
49.   int count = 0;
50.
51.   substr(salt, salt_and_encrypted, 0, 6);
52.
53.   for(b='A'; b<='M'; b++){
54.     for(i='A'; i<='Z'; i++){
55.       for(s=0; s<=99; s++){
56.         sprintf(plain, "%c%c%02d", b,i,s);
57.         enc = (char *) crypt(plain, salt);
58.         count++;
59.         if(strcmp(salt_and_encrypted, enc) == 0){
60.           printf("#%-8d%s %s\n", count, plain, enc);
61.         }
62.       }
63.     }
64.   }
65.   printf("%d solutions explored\n", count);
66. }
67.
68.
69. void *kernel_function_2(void *salt_and_encrypted){
70.   int b, i, s;     // Loop counters
71.   char salt[7];    // String used in hahttps://www.youtube.com/watch?v=L8yJjIGleMwshing the password. Need space
72.   char plain[7];   // The combination of letters currently being checked
73.   char *enc;       // Pointer to the encrypted password
74.   int count = 0;   // The number of combinations explored so far
75.
76.   substr(salt, salt_and_encrypted, 0, 6);
77.
78.   for(b='N'; b<='Z'; b++){
79.     for(i='A'; i<='Z'; i++){
80.       for(s=0; s<=99; s++){
81.         sprintf(plain, "%c%c%02d", b,i,s);
82.         enc = (char *) crypt(plain, salt);
83.         count++;
84.         if(strcmp(salt_and_encrypted, enc) == 0){
85.           printf("#%-8d%s %s\n", count, plain, enc);
86.         }
87.       }
88.     }
89.   }
```

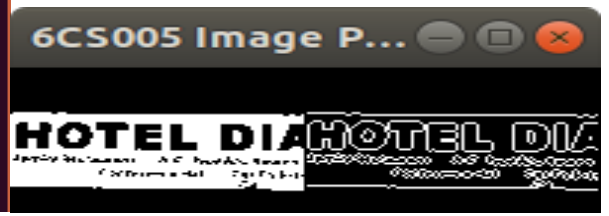
```
90.   printf("%d solutions explored\n", count);
91. }
92.
93. //Calculating time
94.
95. int time_difference(struct timespec *start, struct timespec *finish, long long int *difference)
96.  {
97.       long long int ds =  finish->tv_sec - start->tv_sec;
98.       long long int dn =  finish->tv_nsec - start->tv_nsec;
99.
100.             if(dn < 0 ) {
101.               ds--;
102.               dn += 1000000000;
103.         }
104.             *difference = ds * 1000000000 + dn;
105.             return !(*difference > 0);
106.       }
107.       int main(int argc, char *argv[])
108.       {
109.
110.           struct timespec start, finish;
111.           long long int time_elapsed;
112.
113.           clock_gettime(CLOCK_MONOTONIC, &start);
114.
115.
116.
117.                 pass();
118.
119.           clock_gettime(CLOCK_MONOTONIC, &finish);
120.             time_difference(&start, &finish, &time_elapsed);
121.             printf("Time elapsed was %lldns or %0.9lfs\n", time_elapsed,
122.                                                 (time_elapsed/1.0e9));
123.         return 0;
124.       }
```

Write a paragraph that compares the original results with those of your multithread password cracker.

1.2 Image Processing

Insert the image displayed by your program

```
bishrut@herald-OptiPlex-3050:~/Downloads/bishrut hpc/2a1$ cc -o ip_coursework_06
9 ip_coursework_069.c -lglut -lGL -lm
bishrut@herald-OptiPlex-3050:~/Downloads/bishrut hpc/2a1$ chmod a+x mr.py
bishrut@herald-OptiPlex-3050:~/Downloads/bishrut hpc/2a1$ ./mr.py ./ip_coursewor
k_069|grep Time>imgpro2a.csv
bishrut@herald-OptiPlex-3050:~/Downloads/bishrut hpc/2a1$
```




```
Insert the code for your multithreaded edge detector here
```

```c
#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>
#define width 100
#define height 72
typedef struct argument_t{
    int start;
    int stride;
}Bis_thread;

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

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

  for(i=args->start;i<n_pixels;i+=args->stride) {
    int x, y;
    int b, d, f, h;
    int r;

    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) {
        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();
}
void *detector(void *args){

detect_edges(image,results,args);

}
static void key_pressed(unsigned char key, int x, int y) {
  switch(key){
    case 27:
```

```
75.       tidy_and_exit();
76.       break;
77.     default:
78.       printf("\nPress escape to exit\n");
79.       break;
80.   }
81. }
82. int time_difference(struct timespec *start, struct timespec *finish,
83.                       long long int *difference) {
84.   long long int ds =  finish->tv_sec - start->tv_sec;
85.   long long int dn =  finish->tv_nsec - start->tv_nsec;
86.
87.   if(dn < 0 ) {
88.     ds--;
89.     dn += 1000000000;
90.   }
91.   *difference = ds * 1000000000 + dn;
92.   return !(*difference > 0);
93. }
94.
95.
96. int main(int argc, char **argv) {
97.
98.   struct timespec start, finish;
99.   long long int time_elapsed;
100.
101.     clock_gettime(CLOCK_MONOTONIC, &start);
102.
103.   printf("image dimensions %dx%d\n", width, height);
104.     pthread_t t1, t2, t3, t4;
105.
106.     Bis_thread t1_arguments;
107.     t1_arguments.start = 0;
108.     t1_arguments.stride = 4;
109.
110.     Bis_thread t2_arguments;
111.     t2_arguments.start = 1;
112.     t2_arguments.stride = 4;
113.
114.     Bis_thread t3_arguments;
115.     t3_arguments.start = 2;
116.     t3_arguments.stride = 4;
117.
118.     Bis_thread t4_arguments;
119.     t4_arguments.start = 3;
120.     t4_arguments.stride = 4;
121.
```

```
122.
123.     pthread_create(&t1, NULL, detector, &t1_arguments);
124.     pthread_create(&t2, NULL, detector, &t2_arguments);
125.     pthread_create(&t3, NULL, detector, &t3_arguments);
126.     pthread_create(&t4, NULL, detector, &t4_arguments);
127.
128.     pthread_join(t1, NULL);
129.     pthread_join(t2, NULL);
130.     pthread_join(t3, NULL);
131.     pthread_join(t4, NULL);
132.
133.     clock_gettime(CLOCK_MONOTONIC, &finish);
134.     time_difference(&start, &finish, &time_elapsed);
135.     printf("Time elapsed was %lldns or %0.9lfs\n", time_elapsed,
136.            (time_elapsed/1.0e9));
137.
138.     signal(SIGINT, sigint_callback);
139.
140.     printf("image dimensions %dx%d\n", width, height);
141.
142.
143.     glutInit(&argc, argv);
144.     glutInitWindowSize(width * 2,height);
145.     glutInitDisplayMode(GLUT_SINGLE | GLUT_LUMINANCE);
146.
147.     glutCreateWindow("6CS005 Image Progessing Courework");
148.     glutDisplayFunc(display);
149.     glutKeyboardFunc(key_pressed);
150.     glClearColor(0.0, 1.0, 0.0, 1.0);
151.
152.     glutMainLoop();
153.
154.     tidy_and_exit();
155.
156.   }
157.
158.   unsigned char image[] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
159.     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
160.     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
161.     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
162.     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
163.     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
164.     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
165.     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
166.     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
167.     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
168.     0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
```

```
169.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
170.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
171.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
172.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
173.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
174.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
175.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
176.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
177.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
178.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
179.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
180.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
181.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
182.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
183.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
184.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
185.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
186.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
187.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
188.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
189.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
190.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
191.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
192.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
193.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
194.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
195.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
196.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
197.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
198.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
199.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
200.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
201.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
202.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
203.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
204.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
205.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
206.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
207.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
208.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
209.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
210.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
211.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
212.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
213.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
214.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
215.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
```

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

```
263.      255,255,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
264.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,255,
265.      255,255,0,255,255,0,255,0,0,255,255,255,255,255,255,255,255,255,255,
266.      255,255,255,255,255,255,255,255,255,0,255,255,255,0,255,255,255,255,255,
267.      255,255,255,255,255,0,0,255,255,255,255,255,255,255,0,0,255,255,255,
268.      255,255,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
269.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
270.      255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
271.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
272.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
273.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
274.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
275.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
276.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
277.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
278.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
279.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
280.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
281.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
282.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
283.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
284.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
285.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
286.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
287.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,255,0,0,255,
288.      0,255,255,0,255,0,0,255,255,0,0,255,255,255,0,255,0,0,255,
289.      0,0,0,255,0,255,255,255,0,255,255,255,255,255,255,255,0,255,255,
290.      255,0,255,255,255,255,255,0,255,255,255,255,0,255,0,255,0,0,255,
291.      255,0,0,0,0,0,0,0,255,0,0,0,255,0,255,0,255,255,255,
292.      255,255,255,255,255,0,255,0,255,255,0,0,255,255,255,255,255,0,255,
293.      0,0,0,255,0,0,0,255,0,0,0,255,255,0,255,255,255,255,0,
294.      0,255,0,0,0,255,0,0,255,0,0,255,0,0,255,255,255,255,0,
295.      0,255,0,0,0,255,0,255,0,0,0,255,0,0,255,255,0,0,0,
296.      0,255,0,255,255,255,255,0,0,255,255,0,0,255,0,0,255,0,0,
297.      255,0,255,255,255,255,255,255,255,255,255,0,0,255,255,0,255,255,255,
298.      255,255,255,0,0,255,255,255,255,255,255,0,255,255,0,255,255,0,255,
299.      255,255,255,255,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,
300.      255,255,255,255,255,0,255,0,0,255,0,0,0,255,255,255,0,255,0,
301.      255,255,255,0,255,0,255,255,255,255,0,255,255,255,255,255,255,255,255,
302.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,255,255,
303.      255,255,0,0,255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,
304.      0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
305.      255,255,255,255,255,255,255,255,255,255,255,0,255,255,255,255,255,255,255,255,
306.      0,255,255,255,255,255,255,0,255,255,255,255,255,255,255,255,255,255,255,
307.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
308.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
309.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
```

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

```
357.      0,0,255,255,255,0,0,0,255,255,255,255,0,0,0,0,255,0,0,
358.      0,255,255,0,0,0,0,0,0,0,0,0,0,255,255,255,0,0,
359.      0,255,255,255,255,255,0,0,0,0,255,255,255,255,0,0,0,0,255,
360.      255,255,255,255,255,0,0,0,0,0,0,0,0,255,255,255,0,0,
361.      0,0,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,
362.      255,255,255,0,0,0,0,255,255,255,0,0,0,255,255,255,255,255,0,
363.      0,0,0,0,0,0,255,255,0,0,0,0,0,0,0,0,0,0,0,
364.      255,255,255,0,0,0,0,255,255,255,255,0,0,0,0,255,255,255,255,
365.      0,0,0,0,255,255,255,255,255,255,0,0,0,0,0,0,0,0,0,
366.      255,255,255,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,
367.      255,0,0,0,0,255,255,255,0,0,0,0,255,255,255,0,0,0,255,
368.      255,255,255,255,0,0,0,0,0,0,0,255,255,0,0,0,0,0,0,
369.      0,0,0,0,0,0,255,255,0,0,0,0,255,255,255,0,0,0,0,
370.      255,255,255,255,255,0,0,0,0,255,255,255,255,255,255,0,0,0,0,
371.      255,255,255,255,255,255,255,255,0,0,0,0,255,255,255,255,255,255,255,
372.      255,255,255,255,255,255,0,0,0,0,255,255,0,0,0,0,0,255,255,
373.      255,0,0,0,255,255,255,255,255,0,0,0,0,0,0,255,255,0,
374.      0,0,0,255,255,255,255,0,0,0,0,255,255,0,0,0,0,0,255,
375.      0,0,0,0,0,255,255,0,0,0,0,0,0,0,0,0,0,255,
376.      255,0,0,0,0,0,0,0,0,0,255,255,255,0,0,0,0,255,255,
377.      255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,0,0,0,0,0,
378.      0,0,255,255,255,255,0,0,0,255,255,255,255,255,255,0,0,0,0,
379.      0,255,255,255,0,0,0,0,255,255,255,255,0,0,0,0,255,255,255,
380.      0,0,0,0,0,0,0,0,0,0,255,255,0,0,0,0,0,0,0,
381.      0,0,0,0,255,255,0,0,0,0,0,0,0,0,0,255,255,255,0,
382.      0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,0,
383.      0,0,0,0,0,0,0,255,255,255,255,0,0,0,255,255,255,255,255,
384.      255,0,0,0,0,0,0,255,255,255,0,0,0,0,255,255,255,255,0,0,
385.      0,0,255,255,255,255,0,0,0,0,0,0,0,0,255,255,255,0,0,
386.      0,0,0,0,0,0,0,0,0,0,255,255,0,0,0,0,0,0,0,0,
387.      0,255,255,255,0,0,0,0,0,255,255,255,255,255,255,255,255,255,255,255,255,
388.      255,255,0,0,0,0,0,0,0,0,0,255,255,255,255,255,255,255,0,0,0,
389.      255,255,255,255,255,255,255,0,0,0,0,0,255,255,255,255,0,0,0,255,
390.      255,255,255,0,0,0,255,255,255,255,255,255,255,0,0,0,0,0,255,255,
391.      255,255,255,0,0,0,0,0,0,0,0,0,0,0,255,255,255,255,255,255,255,
392.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
393.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
394.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
395.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
396.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
397.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
398.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
399.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
400.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
401.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
402.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
403.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
```

```
404.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
405.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
406.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
407.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
408.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
409.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
410.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
411.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
412.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
413.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
414.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,
415.      255,255,255,255,255,255,255,255,255,255,255,0,0,255,255,0,0,255,255,0,
416.      255,255,255,255,255,255,255,255,255,255,255,255,255,255,0,0,255,255,255,0,
417.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
418.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
419.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
420.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
421.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
422.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
423.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
424.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
425.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
426.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
427.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
428.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
429.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
430.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
431.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
432.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
433.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
434.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
435.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
436.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
437.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
438.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
439.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
440.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
441.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
442.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
443.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
444.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
445.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
446.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
447.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
448.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
449.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
450.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
```

```
451.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
452.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
453.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
454.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
455.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
456.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
457.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
458.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
459.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
460.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
461.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
462.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
463.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
464.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
465.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
466.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
467.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
468.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
469.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
470.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
471.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
472.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
473.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
474.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
475.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
476.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
477.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
478.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
479.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
480.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
481.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
482.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
483.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
484.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
485.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
486.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
487.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
488.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
489.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
490.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
491.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
492.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
493.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
494.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
495.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
496.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
497.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
```

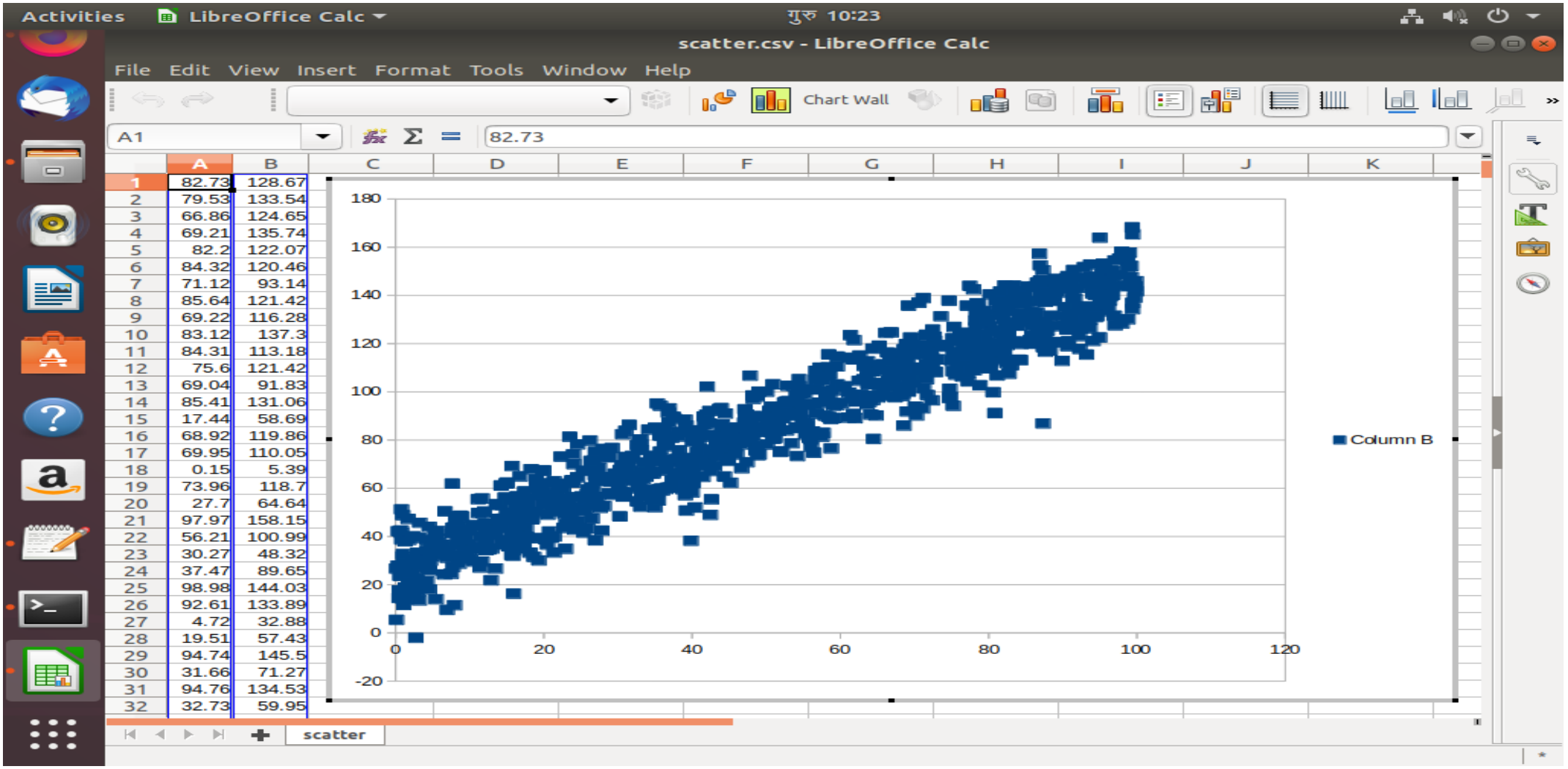
```
498.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
499.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
500.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
501.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
502.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
503.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
504.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
505.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
506.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
507.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
508.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
509.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
510.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
511.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
512.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
513.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
514.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
515.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
516.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
517.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
518.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
519.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
520.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
521.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
522.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
523.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
524.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
525.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
526.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
527.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
528.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
529.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
530.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
531.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
532.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
533.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
534.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
535.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
536.      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
537.    };
```

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

Insert an explanation of the results presented in the above table.

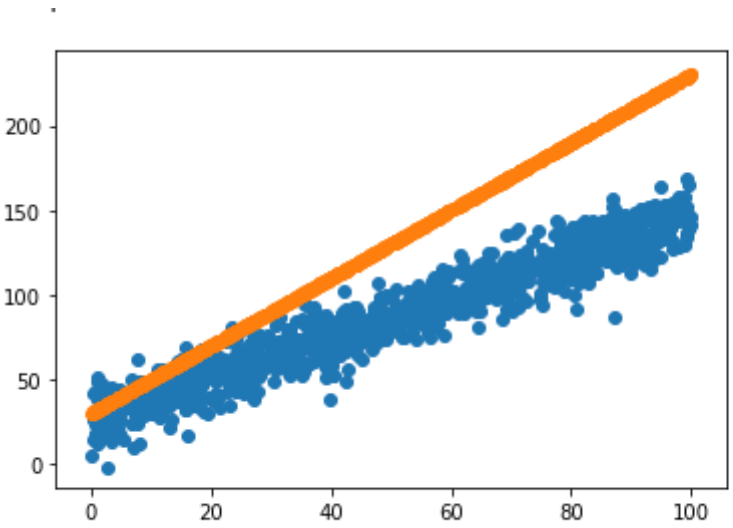
1.3 Linear Regression

Insert a scatter plot of your data.

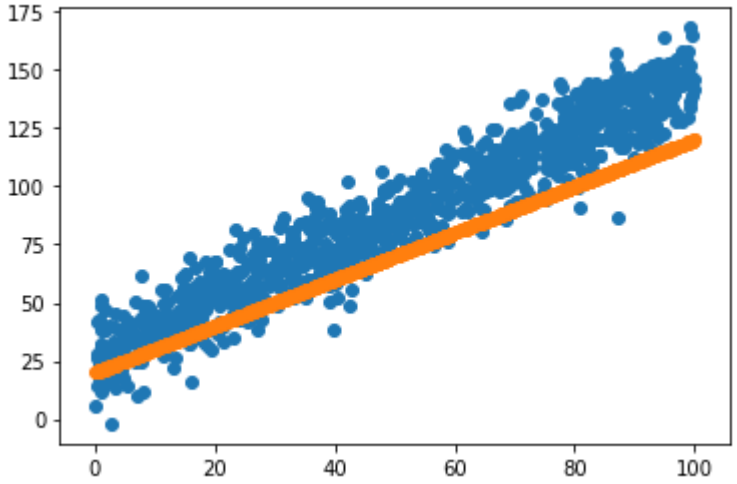


Have 3 guesses at the optimum values for m and c and present them in a graph that overlays your data.

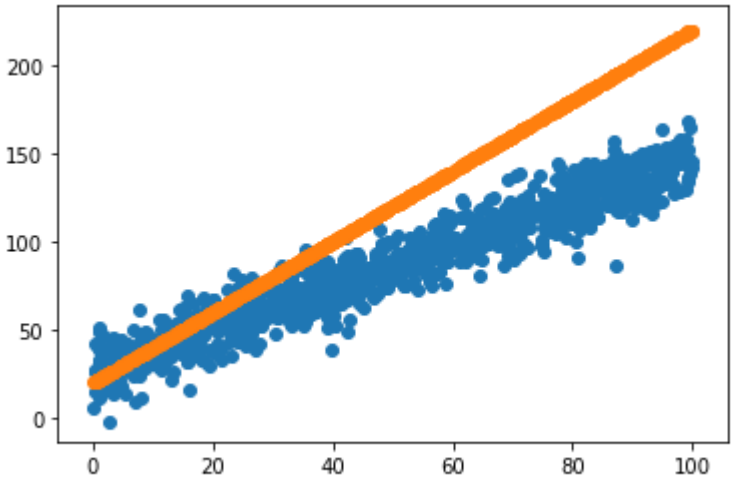
Guess1:



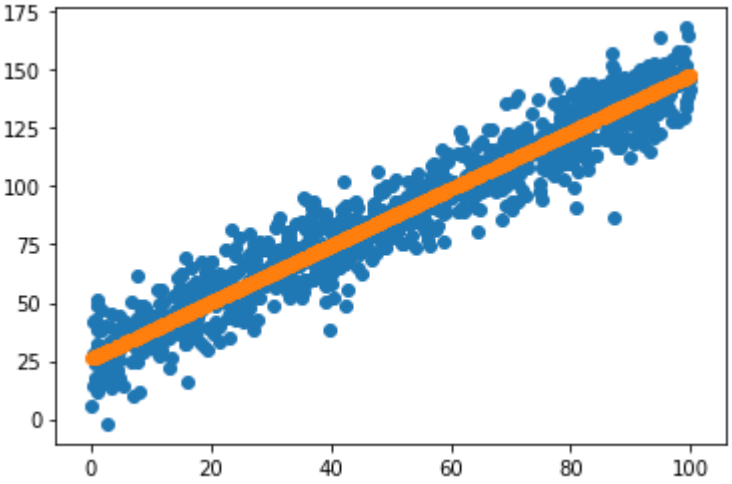
Guess2:



Guess3:



Insert a graph that presents your data with the solution overlaid.



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

Paste your source code for your multithread linear regression program here.

```c
#include <stdio.h>
#include <math.h>
#include <time.h>
#include <pthread.h>
int i;
  double bm = 1.3;
  double bc = 10;
  double be;
  double dm[8];
  double dc[8];
  double e[8];
  double step = 0.01;
  double best_error = 999999999;
  int best_error_i;
  int minimum_found = 0;
  double om[] = {0,1,1, 1, 0,-1,-1,-1};
  double oc[] = {1,1,0,-1,-1,-1, 0, 1};

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

int n_data = 1000;
point_t data[];

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

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

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

```
40.
41.   mean = error_sum / n_data;
42.
43.   return sqrt(mean);
44. }
45. int time_difference(struct timespec *start, struct timespec *finish, long long int *difference)
46. {
47.   long long int ds =  finish->tv_sec - start->tv_sec;
48.   long long int dn =  finish->tv_nsec - start->tv_nsec;
49.
50.   if(dn < 0 ) {
51.     ds--;
52.     dn += 1000000000;
53.   }
54.   *difference = ds * 1000000000 + dn;
55.   return !(*difference > 0);
56. }
57.
58. void *linear_regression_thread(void *args){
59.
60.   int *a = args;
61.   int i = *a;
62.
63.   dm[i] = bm +(om[i] * step);
64.       dc[i] = bc + (oc[i] * step);
65.   e[i] = rms_error(dm[i], dc[i]);
66.   if(e[i] < best_error) {
67.   best_error = e[i];
68.   best_error_i = i;
69.   pthread_exit(NULL);
70.   }
71. }
72.
73. int main() {
74.    struct timespec start, finish;
75.   long long int time_elapsed;
76.   clock_gettime(CLOCK_MONOTONIC, &start);
77.
78.   int i;
79.   pthread_t p_threads[8];
80.
81.   be = rms_error(bm, bc);
82.
83.   while(!minimum_found) {
84.     for(i=0;i<8;i++) {
85.       pthread_create(&p_threads[i], NULL, (void*)linear_regression_thread, &i);
86.       pthread_join(p_threads[i], NULL);
```

```
    }
    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[] = {
        {82.73,128.67},{79.53,133.54},{66.86,124.65},{69.21,135.74},
        {82.20,122.07},{84.32,120.46},{71.12,93.14},{85.64,121.42},
        {69.22,116.28},{83.12,137.30},{84.31,113.18},{75.60,121.42},
        {69.04,91.83},{85.41,131.06},{17.44,58.69},{68.92,119.86},
        {69.95,110.05},{ 0.15, 5.39},{73.96,118.70},{27.70,64.64},
        {97.97,158.15},{56.21,100.99},{30.27,48.32},{37.47,89.65},
        {98.98,144.03},{92.61,133.89},{ 4.72,32.88},{19.51,57.43},
        {94.74,145.50},{31.66,71.27},{94.76,134.53},{32.73,59.95},
        {32.64,54.53},{38.78,69.06},{91.47,150.49},{77.99,119.35},
        {33.38,65.87},{79.28,123.62},{39.69,72.53},{95.47,140.97},
        {82.64,137.69},{25.53,51.33},{68.58,85.98},{92.25,132.34},
        {74.79,101.30},{ 1.32,18.87},{53.85,95.13},{78.75,128.26},
        { 2.91,21.77},{90.68,128.55},{11.44,35.27},{30.72,56.54},
        {49.06,74.08},{49.09,83.45},{62.54,104.58},{38.83,72.26},
        {78.43,130.83},{69.49,122.49},{27.27,56.35},{80.06,131.95},
        { 5.73,39.00},{80.21,140.42},{ 8.47,36.12},{86.98,152.43},
        {64.26,108.56},{95.74,133.36},{15.06,48.67},{31.96,72.31},
        {95.27,141.34},{61.10,89.26},{27.51,68.47},{26.48,60.30},
        {92.61,128.38},{ 8.25,47.51},{90.69,118.91},{45.40,79.96},
        {23.59,53.12},{46.71,68.27},{21.15,50.29},{27.99,76.29},
        { 7.75,43.57},{13.70,43.56},{74.85,97.83},{50.93,103.11},
        {33.80,64.85},{80.99,125.37},{92.41,126.27},{92.61,134.36},
        {34.70,55.32},{35.07,55.04},{86.87,157.26},{41.99,90.46},
        {16.27,44.43},{36.31,83.84},{22.35,73.11},{89.11,127.19},
        {56.11,77.28},{51.90,75.07},{35.74,94.18},{10.66,29.60},
        {61.27,114.15},{77.55,117.04},{61.17,99.68},{15.54,55.33},
        {91.99,143.18},{12.91,21.82},{48.52,89.94},{54.88,90.86},
        {73.59,131.33},{ 5.49,13.95},{92.31,147.29},{48.50,89.49},
        {40.02,58.26},{48.22,81.96},{17.08,52.59},{34.27,66.17},
```

```
134.         {59.06,94.26},{92.71,134.53},{37.70,65.30},{77.11,111.38},
135.         {43.27,74.12},{79.71,123.45},{ 0.86,38.69},{ 3.00,17.76},
136.         {56.03,80.33},{17.66,43.27},{18.39,47.08},{31.08,83.84},
137.         {32.64,77.85},{51.68,84.57},{78.46,134.18},{ 9.57,40.28},
138.         {68.38,98.26},{30.29,67.59},{86.15,131.86},{16.82,64.91},
139.         { 3.35,20.88},{65.78,98.73},{50.70,90.92},{38.26,71.11},
140.         {85.52,132.23},{44.06,83.02},{44.09,86.42},{81.86,114.30},
141.         {33.98,69.09},{93.80,147.73},{59.58,103.07},{98.75,154.73},
142.         {88.98,120.59},{78.08,109.00},{82.77,133.94},{76.49,106.31},
143.         {55.38,85.71},{46.56,79.57},{83.92,141.58},{81.38,133.52},
144.         { 4.88,35.01},{ 4.57,17.99},{57.96,90.07},{33.42,63.80},
145.         { 9.95,34.53},{47.14,92.75},{63.17,105.19},{95.01,163.93},
146.         {30.36,57.81},{ 2.46,23.97},{69.75,115.88},{64.85,111.01},
147.         {25.18,56.58},{69.84,104.78},{40.43,51.98},{75.61,107.05},
148.         {36.75,69.37},{50.08,100.02},{64.97,103.68},{41.72,86.64},
149.         { 1.70,47.26},{99.93,141.75},{24.57,64.51},{75.23,116.35},
150.         { 1.95,18.53},{78.84,102.70},{67.38,97.71},{55.35,82.37},
151.         {58.10,100.09},{53.10,96.07},{41.24,83.81},{68.86,111.98},
152.         {87.36,86.88},{54.06,98.42},{64.12,90.56},{11.77,49.66},
153.         {99.43,134.33},{55.24,99.18},{56.44,74.73},{39.47,62.99},
154.         { 8.94,48.15},{92.91,130.45},{87.68,138.76},{80.37,116.69},
155.         {56.72,108.65},{ 0.76,24.26},{26.98,75.13},{ 0.39,42.16},
156.         {81.99,138.50},{88.32,117.16},{51.01,87.42},{21.38,55.45},
157.         {72.66,122.82},{18.04,53.56},{11.22,49.73},{36.75,60.26},
158.         {64.81,90.19},{72.72,121.14},{24.03,74.08},{41.38,81.38},
159.         {62.79,98.75},{63.66,109.17},{91.12,143.91},{ 7.41,34.06},
160.         {94.05,131.99},{53.12,90.28},{68.31,114.79},{25.33,67.23},
161.         {42.34,86.91},{94.61,131.38},{43.78,73.28},{50.18,78.10},
162.         {81.64,135.88},{11.27,44.45},{41.03,76.34},{21.25,57.54},
163.         {29.23,57.27},{35.74,75.16},{ 0.91,14.33},{30.08,59.05},
164.         {23.99,56.25},{90.79,120.98},{99.22,152.22},{94.21,143.09},
165.         {19.35,30.03},{82.04,113.25},{79.22,113.69},{83.40,144.06},
166.         {55.82,80.85},{42.49,48.94},{17.60,55.62},{35.65,81.91},
167.         {82.50,135.41},{81.15,114.46},{53.47,78.67},{44.30,73.73},
168.         {32.88,80.28},{99.26,147.55},{76.32,110.24},{78.97,110.27},
169.         {18.08,47.48},{87.01,140.40},{56.25,83.61},{42.62,55.40},
170.         {15.95,16.25},{47.85,106.69},{ 6.61,35.83},{66.38,116.30},
171.         {94.97,122.56},{42.29,73.37},{31.48,67.15},{69.67,105.40},
172.         {30.41,65.31},{ 2.98,19.40},{ 8.12,48.34},{80.41,127.03},
173.         {63.68,112.61},{24.60,78.23},{77.61,123.49},{39.87,38.20},
174.         {77.80,109.59},{58.53,107.63},{23.97,62.36},{ 7.77,27.38},
175.         { 0.80,41.55},{ 6.45,32.91},{45.32,82.24},{35.56,59.56},
176.         {65.05,97.68},{62.21,96.14},{86.61,121.99},{87.91,125.40},
177.         {48.08,88.87},{ 2.41,40.02},{69.55,119.31},{22.07,61.86},
178.         {61.87,121.40},{82.50,119.46},{26.97,38.40},{31.53,86.30},
179.         { 1.81,38.57},{72.57,108.34},{88.88,139.23},{63.90,95.79},
180.         {93.29,135.35},{86.26,143.55},{63.62,94.76},{20.24,38.84},
```

```
        {16.23,48.64},{72.87,108.22},{16.26,51.25},{37.86,66.06},
        {57.53,81.37},{61.66,97.20},{49.48,84.98},{95.20,142.45},
        {12.10,45.25},{47.79,84.80},{17.29,48.98},{47.11,87.23},
        {85.74,119.95},{89.94,142.94},{97.68,155.27},{78.73,123.81},
        {51.65,85.91},{52.82,96.05},{50.95,93.50},{16.14,37.21},
        {16.73,41.57},{57.25,95.50},{78.47,136.77},{42.35,75.64},
        {93.24,135.04},{12.56,38.20},{21.40,62.92},{70.60,136.98},
        {44.04,83.57},{ 6.43,36.61},{12.01,50.32},{79.61,119.78},
        {43.05,69.07},{14.42,53.01},{51.68,83.82},{25.59,55.77},
        { 9.14,31.58},{37.24,80.94},{15.73,69.21},{71.54,123.11},
        { 1.26,25.72},{ 4.25,38.46},{21.42,39.99},{44.12,79.01},
        {31.12,64.63},{85.27,143.62},{43.25,79.30},{77.27,104.30},
        {47.34,83.76},{90.57,125.82},{17.35,36.40},{82.01,130.41},
        {81.58,124.10},{68.62,117.62},{47.48,79.29},{ 4.30,26.77},
        { 6.94,32.22},{11.71,55.76},{22.62,54.74},{58.43,89.61},
        {69.10,111.51},{56.77,101.10},{67.10,102.75},{93.20,144.51},
        {83.61,128.56},{71.97,116.09},{75.19,122.16},{48.03,79.67},
        {97.95,143.80},{92.27,123.08},{23.88,63.39},{79.15,115.57},
        {24.42,51.27},{12.58,34.65},{46.58,78.16},{ 1.29,37.96},
        {17.09,45.61},{12.45,40.77},{82.75,107.46},{52.15,75.34},
        {39.51,68.51},{31.71,64.23},{39.36,72.00},{12.16,37.99},
        {83.13,127.76},{42.25,73.17},{45.32,77.14},{20.52,36.60},
        { 7.99,11.50},{23.34,55.47},{25.87,54.36},{78.73,112.49},
        {55.60,94.90},{31.98,73.40},{85.93,137.12},{58.56,97.64},
        {88.16,120.43},{78.65,136.60},{25.93,43.32},{84.83,136.32},
        {68.09,102.12},{68.36,111.80},{39.80,69.69},{ 0.38,27.89},
        { 4.49,27.85},{32.53,66.32},{54.23,97.63},{19.98,67.32},
        {90.62,143.43},{18.31,67.91},{95.66,146.41},{95.41,149.68},
        {71.64,111.15},{23.02,44.96},{97.06,154.54},{41.58,75.95},
        {79.80,130.01},{74.55,119.44},{72.19,113.27},{70.01,106.48},
        {75.24,94.18},{19.82,60.09},{96.31,137.91},{ 2.21,27.44},
        {40.52,70.36},{ 2.40,29.12},{35.24,57.25},{26.38,71.34},
        {26.02,59.48},{34.73,66.07},{45.15,78.23},{ 9.35,32.58},
        {19.37,57.18},{ 9.51,31.70},{15.03,49.81},{85.08,140.35},
        { 3.23,13.46},{58.26,108.47},{ 4.84,31.78},{49.49,83.50},
        {35.55,70.67},{26.51,55.44},{20.12,53.39},{72.73,119.37},
        {31.04,72.96},{30.66,58.35},{ 2.96,33.18},{18.68,31.50},
        {91.41,138.24},{44.67,81.81},{81.57,135.26},{ 0.17,26.66},
        {49.03,100.11},{54.47,102.27},{61.78,113.45},{22.67,59.51},
        {89.80,143.05},{33.05,78.20},{67.76,108.19},{ 7.64,41.18},
        {36.91,87.28},{95.44,147.27},{52.76,94.34},{ 3.52,29.51},
        {87.39,118.48},{41.48,64.71},{ 1.44,14.21},{95.04,136.99},
        {71.77,115.75},{23.39,47.58},{62.66,115.03},{15.98,34.38},
        {29.06,62.77},{ 2.94,28.25},{71.50,119.18},{65.24,119.14},
        {30.65,82.39},{16.36,38.82},{ 0.98,48.82},{33.19,56.41},
        {27.49,64.34},{53.69,102.47},{28.15,52.58},{40.21,66.07},
        {50.56,86.39},{74.71,97.44},{24.72,46.29},{48.05,80.63},
```

```
228.         {34.99,52.13},{66.75,115.96},{17.62,49.17},{98.99,157.80},
229.         {37.96,72.18},{56.88,105.06},{48.27,97.04},{71.18,138.90},
230.         {46.35,82.02},{10.43,44.65},{24.14,42.85},{82.21,144.13},
231.         {96.85,148.15},{93.68,126.32},{33.02,61.55},{66.73,108.51},
232.         {83.89,136.35},{80.85,91.16},{79.21,128.88},{84.37,119.84},
233.         {38.41,71.48},{47.49,85.53},{ 1.54,24.44},{68.32,106.44},
234.         {22.82,54.16},{ 2.65,16.35},{19.91,53.53},{12.99,34.98},
235.         {30.87,57.17},{44.10,83.88},{15.84,31.99},{36.46,59.74},
236.         {26.25,79.73},{79.12,132.06},{86.26,132.45},{ 0.61,23.61},
237.         {33.94,59.37},{99.92,145.88},{26.20,53.99},{69.77,115.40},
238.         {69.07,107.00},{ 1.89,17.20},{38.25,81.40},{27.08,62.96},
239.         {23.09,53.98},{55.56,86.93},{ 6.68,50.41},{22.86,49.26},
240.         {17.25,50.25},{19.01,50.16},{35.07,85.09},{59.08,89.15},
241.         {87.02,128.83},{ 1.57,27.68},{97.76,148.25},{70.78,108.00},
242.         {38.01,65.83},{96.41,139.67},{ 2.86,22.44},{27.05,53.00},
243.         {90.99,134.97},{86.60,145.27},{54.66,99.42},{67.61,107.07},
244.         {85.16,137.50},{87.64,144.60},{14.69,36.65},{16.08,49.31},
245.         {14.45,44.07},{65.91,98.39},{50.74,90.72},{ 6.98,31.11},
246.         {52.76,83.96},{ 8.03,43.93},{17.58,52.58},{33.63,59.04},
247.         {87.65,137.34},{77.97,142.54},{30.56,69.47},{59.61,114.61},
248.         {14.05,53.07},{87.65,116.66},{33.19,75.96},{31.87,66.95},
249.         {25.89,57.59},{48.60,75.67},{80.25,109.89},{ 6.61,24.27},
250.         { 4.56,44.00},{40.17,62.33},{92.32,117.73},{75.07,112.71},
251.         {17.10,35.12},{39.06,66.60},{ 4.26,34.01},{52.95,102.49},
252.         {45.73,76.57},{ 4.72,29.94},{ 2.01,17.54},{39.08,88.44},
253.         {82.94,141.75},{44.51,90.97},{27.27,63.14},{60.16,95.38},
254.         {41.26,72.59},{66.50,104.49},{58.37,110.13},{62.11,96.01},
255.         {70.30,90.15},{18.47,47.61},{24.80,51.82},{79.02,133.40},
256.         {96.61,147.92},{18.14,33.27},{ 0.83,51.20},{99.67,143.65},
257.         {34.07,67.38},{57.28,110.02},{35.92,59.90},{66.15,124.45},
258.         {81.82,135.08},{ 2.97,28.49},{95.97,135.79},{51.17,80.95},
259.         {91.47,142.00},{94.09,121.08},{57.70,82.98},{67.96,100.92},
260.         {81.91,132.34},{11.55,39.74},{86.59,126.05},{ 5.36,41.72},
261.         {90.86,144.15},{81.02,137.56},{35.87,81.76},{63.73,105.92},
262.         {78.29,129.54},{96.72,150.04},{14.97,61.93},{45.76,77.17},
263.         {82.69,123.95},{85.82,132.89},{85.95,127.24},{15.04,36.92},
264.         {89.91,112.87},{30.86,58.13},{ 5.77,42.22},{75.24,108.41},
265.         { 8.43,32.09},{90.70,147.99},{80.16,112.57},{42.81,73.54},
266.         {82.47,123.41},{48.23,98.48},{77.48,143.96},{ 0.48,14.50},
267.         {29.75,63.12},{88.76,137.72},{33.59,70.61},{22.74,43.51},
268.         {82.15,116.11},{89.10,120.65},{26.56,68.17},{40.72,74.98},
269.         {68.46,99.23},{34.82,66.71},{36.56,67.33},{72.32,114.23},
270.         {29.65,65.99},{44.39,64.83},{82.08,116.35},{99.73,139.12},
271.         {79.04,118.48},{20.78,42.05},{72.39,96.47},{90.62,147.11},
272.         {35.99,59.11},{50.65,83.23},{59.04,100.47},{87.01,145.78},
273.         {43.71,76.56},{95.61,151.81},{50.25,88.96},{69.64,122.07},
274.         {40.07,79.38},{82.61,133.63},{20.84,39.75},{10.28,42.50},
```

```
275.         {47.43,70.82},{30.47,67.19},{69.16,100.10},{46.06,74.00},
276.         {93.78,152.76},{19.93,67.46},{79.61,130.88},{81.11,120.11},
277.         {76.16,123.94},{75.84,111.70},{50.97,85.30},{47.35,90.59},
278.         {93.21,115.44},{19.22,39.30},{11.67,29.58},{52.48,95.64},
279.         {38.76,59.62},{ 2.74,-2.03},{18.99,63.67},{82.38,128.08},
280.         {15.68,32.34},{39.19,83.38},{31.06,65.92},{28.91,73.05},
281.         {19.01,59.69},{76.62,117.74},{36.82,91.33},{86.28,121.19},
282.         {39.26,50.72},{41.45,70.26},{65.81,111.41},{77.09,117.88},
283.         {78.96,128.48},{16.41,56.61},{39.54,64.11},{72.45,110.54},
284.         {48.83,77.35},{27.61,51.82},{26.53,47.44},{83.06,111.09},
285.         {97.06,127.57},{89.01,146.82},{89.44,141.17},{69.18,100.25},
286.         { 1.11,11.60},{71.63,123.66},{92.93,151.73},{99.46,165.34},
287.         {36.49,71.56},{95.48,153.13},{65.33,102.37},{15.28,35.93},
288.         { 5.52,36.67},{ 0.78,42.47},{10.09,36.68},{ 5.75,37.39},
289.         {52.34,89.11},{14.55,47.37},{67.92,113.35},{36.66,77.34},
290.         {99.76,143.75},{26.67,58.72},{ 3.21,39.37},{87.70,124.12},
291.         {90.03,131.24},{51.54,91.39},{62.86,98.04},{52.75,90.87},
292.         {34.17,84.31},{62.00,89.08},{82.47,111.89},{61.38,123.48},
293.         {47.17,84.64},{20.91,53.51},{96.96,131.54},{46.06,85.14},
294.         {26.85,71.44},{91.67,138.51},{54.07,85.26},{51.63,89.63},
295.         {94.04,140.80},{67.75,107.07},{29.24,76.71},{38.29,75.78},
296.         {28.49,72.87},{60.51,102.28},{77.22,107.79},{99.25,145.86},
297.         {33.11,52.32},{72.47,125.80},{21.97,59.23},{14.25,61.11},
298.         {23.79,63.11},{77.78,109.13},{23.51,81.38},{66.92,110.89},
299.         {79.81,109.80},{56.72,94.63},{59.60,110.57},{57.68,104.54},
300.         {27.83,42.43},{47.80,87.68},{58.79,76.51},{78.33,126.71},
301.         {85.14,128.99},{71.61,116.42},{58.09,96.85},{44.89,71.34},
302.         {33.12,80.19},{98.79,130.09},{44.57,82.03},{88.63,142.61},
303.         {61.96,98.55},{58.54,106.80},{19.17,61.00},{13.51,26.68},
304.         {76.68,124.52},{82.62,138.53},{78.13,122.09},{37.10,60.33},
305.         { 8.82,48.63},{71.64,105.27},{68.44,115.07},{ 7.66,61.91},
306.         {64.37,96.58},{54.90,88.28},{78.35,133.29},{79.84,129.58},
307.         { 3.09,28.37},{48.62,76.00},{38.26,63.99},{42.05,102.17},
308.         {48.89,73.66},{54.38,100.05},{11.16,55.43},{63.24,110.69},
309.         {68.17,114.15},{68.68,109.15},{53.43,90.23},{67.45,106.67},
310.         {10.60,34.41},{56.81,90.86},{11.42,27.08},{36.93,93.13},
311.         {41.64,89.77},{69.74,103.98},{23.07,55.12},{44.98,83.65},
312.         {35.75,72.65},{14.80,56.15},{72.19,115.53},{51.10,80.69},
313.         {96.54,140.10},{15.04,62.30},{21.17,56.15},{46.42,79.63},
314.         {22.35,52.01},{35.47,54.95},{ 4.27,21.33},{84.37,139.55},
315.         {43.95,93.24},{86.56,132.82},{44.35,83.36},{76.81,114.79},
316.         { 1.05,31.66},{32.76,76.15},{83.66,120.90},{12.14,42.52},
317.         {25.85,55.83},{82.12,140.05},{75.33,126.93},{32.92,75.90},
318.         { 7.52,24.51},{25.42,41.55},{42.57,67.15},{87.36,150.38},
319.         { 0.51,17.68},{45.70,84.75},{58.74,88.68},{28.62,74.38},
320.         {73.22,113.45},{78.64,114.25},{42.40,92.03},{84.22,132.25},
321.         {54.24,73.34},{ 2.71,30.27},{54.11,84.97},{66.74,112.66},
```

```
322.         {28.80,57.88},{87.02,146.20},{32.02,63.03},{59.57,94.41},
323.         {40.46,79.73},{23.74,49.78},{87.58,140.94},{84.15,113.32},
324.         {32.48,63.48},{ 4.59,25.85},{98.00,128.35},{12.23,37.43},
325.         {66.17,102.97},{50.73,93.82},{74.68,137.79},{43.72,92.85},
326.         {53.95,91.99},{54.47,105.25},{56.70,104.89},{16.59,46.52},
327.         {71.56,115.18},{80.62,99.79},{71.29,101.42},{16.81,56.15},
328.         {48.88,84.93},{ 8.41,40.02},{93.98,147.39},{39.20,86.04},
329.         {61.75,90.80},{ 1.06,32.69},{21.40,33.33},{ 8.60,28.69},
330.         {38.80,61.88},{14.41,38.37},{40.14,70.01},{69.45,105.44},
331.         {14.41,43.93},{51.20,93.11},{39.10,57.10},{21.04,39.51},
332.         {10.12,30.43},{70.13,93.88},{ 1.74,20.56},{12.23,34.33},
333.         {98.81,151.87},{50.48,92.07},{ 6.98, 9.52},{24.08,69.94},
334.         {15.72,40.89},{83.99,127.44},{47.21,90.46},{88.31,138.70},
335.         {91.05,132.13},{45.22,62.24},{87.76,128.67},{99.37,168.24},
336.         {94.38,140.24},{31.30,67.65},{40.85,84.03},{40.91,79.56},
337.         {77.14,135.74},{50.92,80.52},{17.81,49.14},{90.30,135.15},
338.         {28.44,64.60},{49.23,85.12},{81.63,141.58},{83.04,111.19},
339.         {28.39,63.30},{ 8.61,44.11},{25.36,50.79},{51.35,93.32},
340.         {64.49,80.42},{96.17,134.31},{96.10,144.32},{47.58,83.36},
341.         {94.38,131.03},{41.97,69.05},{37.86,62.21},{26.97,65.30},
342.         {37.57,88.95},{65.08,108.58},{17.68,39.80},{63.75,103.14},
343.         {91.86,132.07},{76.35,121.19},{22.98,34.87},{96.46,140.54},
344.         { 9.38,31.40},{42.97,82.09},{20.56,49.02},{13.73,41.31},
345.         {37.35,63.18},{69.54,105.57},{38.17,83.30},{47.04,80.34},
346.         {48.79,98.00},{39.34,61.59},{82.57,125.55},{40.77,82.18},
347.         {13.62,53.38},{35.33,95.17},{95.36,148.79},{20.25,62.00},
348.         {47.48,86.54},{30.22,61.07},{83.90,120.30},{85.81,123.25},
349.         {84.29,130.44},{52.84,95.43},{96.72,140.32},{ 3.29,45.68},
350.         {71.77,98.66},{ 8.52,42.40},{22.55,54.27},{15.08,47.10},
351.         {91.29,130.23},{16.48,40.04},{44.84,72.14},{34.44,73.42},
352.         {36.26,78.30},{58.45,115.51},{96.59,150.22},{63.80,98.30},
353.         {85.92,120.14},{93.68,129.88},{74.09,119.30},{99.44,136.93},
354.         {88.39,131.55},{64.40,117.89},{13.87,47.30},{81.17,106.77}
355.       };
```

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

Write a short analysis of the results.

2 CUDA

2.1 Password Cracking

```
Paste your source code for your CUDA based password cracker here
```

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

Write a short analysis of the results

2.2 Image Processing

```
Paste your source code for your CUDA based image processing.
```

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

Write a short analysis of the results

2.3 Linear Regression

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

```
Paste your source code for your CUDA based linear regression
```

Write a short analysis of the results

3 MPI

3.1 Password Cracking

```
Paste your source code for your MPI based password cracker here
```

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

Write a short analysis of the results

3.2 Image Processing

```
Paste your source code for your MPI based image processor
```

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

Write a short analysis of the results

3.3 Linear Regression

```
Paste your source code for your MPI based linear regression
```

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

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
```