INFORME PARCIAL INFORMÁTICA 2

Participantes: David Alejandro Morón Acacio y Luisa María Bohórquez Ardila

Primeros aportes (14/09/2023)

1. ***Análisis del problema y consideraciones para la alternativa de solución propuesta***
2. ***Esquema donde describa las tareas que usted definió en el desarrollo de los algoritmos.***
3. ***Algoritmos implementados***
4. ***Problemas de desarrollo que afrontó***
5. ***Evolución de la solución y consideraciones a tener en cuenta en la implementación.***

Consideraciones:

1. Usar arreglos, punteros y memoria dinámica.

Preguntas formuladas en el proceso:

1. ¿Qué ventajas tiene la memoria dinámica ante el stack?
2. ¿Para qué su usa el Git-ignore?
3. ¿Cuál es el formato de entrada de los datos para los LED’S?

Acuerdos:

1. Trabajar de 08:00 -09:00 am y de 08:00 pm-12:00 am en el archivo todos los días
2. Subir el segundo commit a las 11:50 pm todos los días
3. Realizaremos el vídeo el viernes 22 de septiembre.

Enlace al proyecto en tinkercard: <https://www.tinkercad.com/things/81HGCE5FQIT?sharecode=mrxWwzQoLmXpCdjmZE6az5yd21304jRjNfOSm302SNQ>

Cosas por hacer:

1. Creación de la matriz de LEDS
2. Creación de la función verificación que encienda y apague los 64 LEDS cada cierto tiempo
3. Creación de la función imagen que recibe ordenes e imprime una figura indicada
4. Imprimir los patrones
5. Menú para el usuario

Desarrollo de las especificaciones

1.

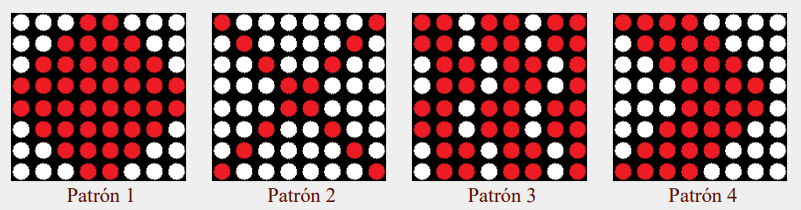
**Código:**

|  |
| --- |
| char high='+', low='-';  char matriz[8][8];  for(int i=0; i<8; i++){  for(int j=0;j<8;j++){  matriz[i][j]= high;  }  }  //Imprimir la matriz  for(int i=0; i<8; i++){  for(int j=0;j<8;j++){  char posicion= matriz[i][j];  char espacio= ' ';  cout<< espacio<< posicion;  }  cout<<endl;  } |

2.

3.

4.



**Patrón 1:** Haremos un triángulo teniendo en cuenta las filas y luego replicaremos ese código de manera decreciente.

**Código:**

|  |
| --- |
| char high='+', low='-';  char matriz[8][8];  int cotainferior=3, cotasuperior=5;  //primera parte de la figura  for(int i=0; i<5; i++){  for(int j=0;j<8;j++){  if(j>=cotainferior and j<cotasuperior){  for(int control2=j; control2<cotasuperior;control2++){  matriz[i][j]=high;}  }else{  matriz[i][j]=low;  }  }  if(i<4){  cotainferior-=1;  cotasuperior+=1;  }    }  //segunda parte de la figura  cotainferior=1; cotasuperior=7;  for(int i=5; i<8; i++){  for(int j=0;j<8;j++){  if(j>=cotainferior and j<cotasuperior){  for(int control2=j; control2<cotasuperior;control2++){  matriz[i][j]=high;}  }else{  matriz[i][j]=low;  }  }  cotainferior+=1;  cotasuperior-=1;  }    //Imprimir la matriz  for(int i=0; i<8; i++){  for(int j=0;j<8;j++){  char posicion= matriz[i][j];  char espacio= ' ';  cout<< espacio<< posicion;  }  cout<<endl;  } |

**Patrón 2:** Analizamos la figura por los extremos, a medida que va iterando la función vamos desplazando la posición de los leds encendidos unos a la izquierda(restando) y otros a la derecha(sumando).

**Código:**

|  |
| --- |
| char high='+', low='-';  char matriz[8][8];  int extremoIzquierdo=0, extremoDerecho=7;  //Figura  for(int i=0; i<8; i++){  for(int j=0;j<8;j++){  if(j==extremoIzquierdo){  matriz[i][j]=high;  }else if(j==extremoDerecho){  matriz[i][j]=high;  }else{  matriz[i][j]=low;  }  }  extremoIzquierdo+=1;  extremoDerecho-=1;  }  //Imprimir la matriz  for(int i=0; i<8; i++){  for(int j=0;j<8;j++){  char posicion= matriz[i][j];  char espacio= ' ';  cout<< espacio<< posicion;  }  cout<<endl;  } |

**Patrón 3:** Haremos que cada cuadrado de 2x2 de la matriz de LEDs se encienda secuencialmente, dejando entre cada cuadrado una columna de 2x1 sin encender.

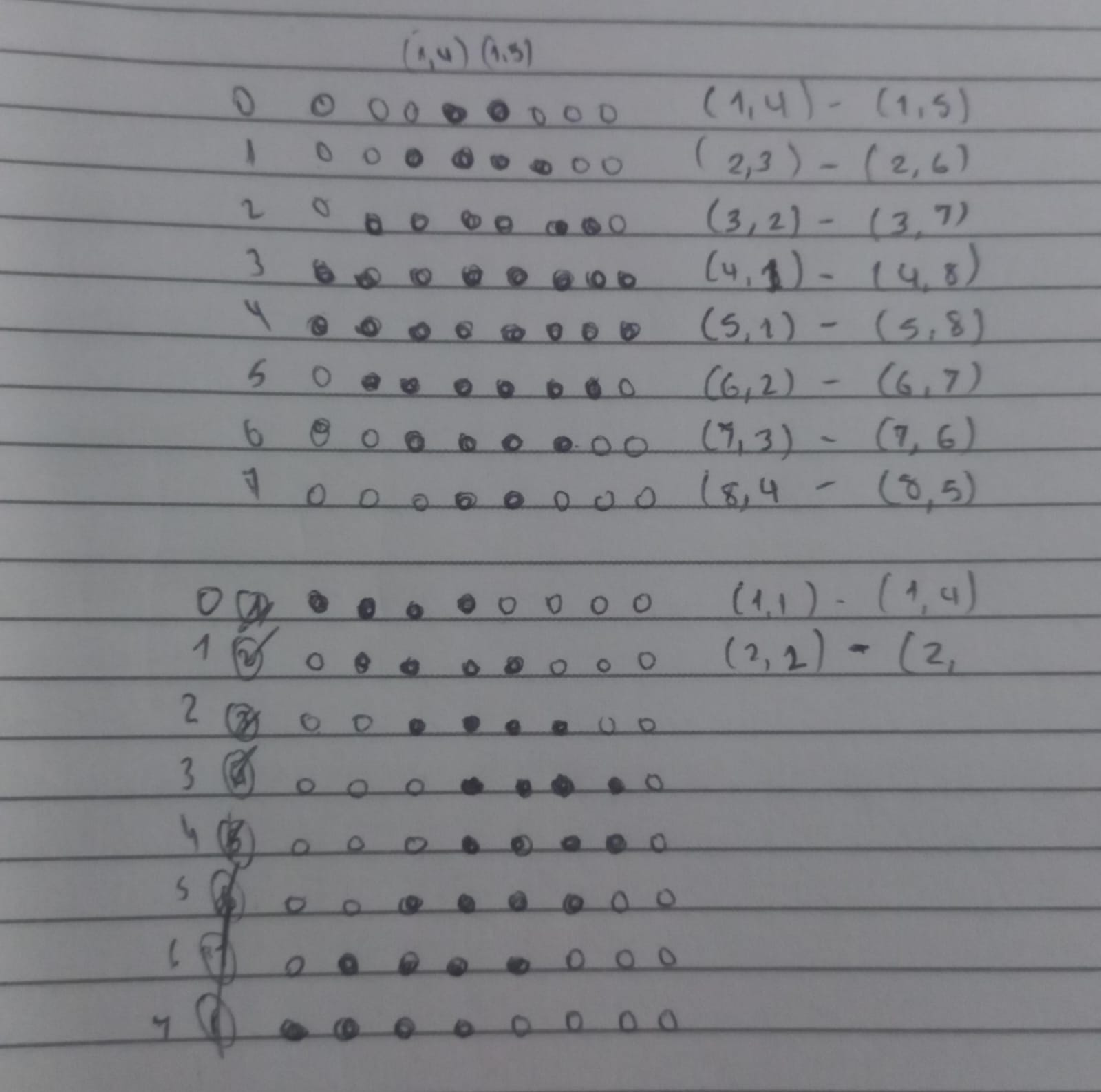
**Patrón 4:** Encendemos 4 leds en el primer extremo y lo vamos desplazando a la derecha hasta la cuarta iteración, repetimos el patrón tal cual en la quinta y de la sexta a la octava desplazamos los leds encendidos a la izquierda

**Código:**

|  |
| --- |
| char high='+', low='-';  char matriz[8][8];  int cotainferior=0, cotasuperior=4;  //primera parte de la figura  for(int i=0; i<5; i++){  for(int j=0;j<8;j++){  if(j>=cotainferior and j<cotasuperior){  for(int control2=j; control2<cotasuperior;control2++){  matriz[i][j]=high;}  }else{  matriz[i][j]=low;  }  }  if(i<3){  cotainferior+=1;  cotasuperior+=1;  }  }  //segunda parte de la figura  cotainferior=2; cotasuperior=6;  for(int i=5; i<8; i++){  for(int j=0;j<8;j++){  if(j>=cotainferior and j<cotasuperior){  for(int control2=j; control2<cotasuperior;control2++){  matriz[i][j]=high;}  }else{  matriz[i][j]=low;  }  }  cotainferior-=1;  cotasuperior-=1;  }  //Imprimir la matriz  for(int i=0; i<8; i++){  for(int j=0;j<8;j++){  char posicion= matriz[i][j];  char espacio= ' ';  cout<< espacio<< posicion;  }  cout<<endl;  } |

**Análisis en papel**

**Ejercicios 1 y 4**

****