CAJA BLANCA

Resuelve los siguientes ejercicios por caja blanca, haz un pantallazo del lucidchart y pégalo aquí en el documento(coloca la **numeración** como las **bolitas** que vimos en clase, puedes usar llaves para englobar varias líneas en una sola bolita). Obtén el grafo, la complejidad ciclomática, los caminos y pon un pequeño ejemplo de que tendría que ocurrir para que el algoritmo pasase por ese camino.

1.- Algoritmo de ordenación de la burbuja
Para más información sobre este algoritmo tienes el enlace debajo.
Puedes probarlo si quieres en el netbeans.inventate un array.

```
public static void burbuja(int[] A) {
      int i, j, aux;
      for (i = 0; i < A.length - 1; i++) {
         for (j = 0; j < A.length - i - 1; j++) {
            if (A[j + 1] < A[j]) {
               aux = A[j + 1];
               A[j + 1] = A[j];
               A[j] = aux;
            }
         }
      }
}
```

Ejemplo de ejecución:

https://puntocomnoesunlenguaje.blogspot.com/2012/07/metodo-de-ordenacion-burbuja.html

2 Movimiento ajedrez- este método recibe un movimiento posible de ajedrez donde se mueven las figuras en las columnas a-h y las filas 1-8, el método no comprueba el movimiento individual característico de cada una o si hay fichas en medio amigas o enemigas, solo comprueba que el rango es adecuado y el parámetro de entrada es del tamaño pedido, el parametro orden seria por ejemplo b2g7, j9a5,k0g5, b6a8, etc…
Si el movimiento es correcto se comprobará en otro método.

```
//comprueba si la "orden" es correcta, es decir, tiene 4 caracteres, 2 y 2, y están en el rango del tablero
bool esMovimiento(string orden, Movimiento &mov)
{
   bool ret;
   char v0,v1,v2,v3;
   if(orden.length()==4)
    {
         v0=orden[0];
         v1=orden[1];
         v2=orden[2];
         v3=orden[3];


            Coordenada orig;
            Coordenada dest;

            orig=convertirCoordenada(v0,v1);
                  dest=convertirCoordenada(v2,v3);
            mov.origen.x=orig.x;
            mov.origen.y=orig.y;
            mov.destino.x=dest.x;
            mov.destino.y=dest.y;

            if(orig.x!=-1 && dest.x!=-1)//si no sale fuera del tablero
             {
                  ret= true;
             }
            else // se va del tablero
             {
                        ret=false;
             }
    }
   else
        ret= false;

   return ret;
}
```

3 movimiento de la torre, (como parámetro del tipo partida poned “Partid1”) simulad un movimiento con alguna figura de por medio y las que necesites para recorrer los caminos, ¿alguna mejora en el código?

```
bool movtorre(int x1, int x2, int y1, int y2,Partida &partida)
{
   bool esValido=false;

        int i;
        esValido=true;
        if(x1==x2)//fichas x el camino(columna)
         {
                for(i=menor(y1,y2)+1;i<mayor(y1,y2);i++)
                 {
                        if(partida.tablero[x1][i].tipo!=VACIA)
                           esValido=false;
                 }
         }
        if(y1==y2)//fichas x el camino(fila)
         {
                for(i=menor(x1,x2)+1;i<mayor(x1,x2);i++)
                 {
                        if(partida.tablero[i][y1].tipo!=VACIA)
                        esValido=false;
                 }
         }

return esValido;
}
```

4 void calculadora () Este ejemplo esta en C++, pero la conversión es sencilla

```
“cout<<””;    ” es lo mismo que “System.out.print(“”);”
“cin>>”; es lo mismo que ‘ Scanner sc=new Scanner(System.in);
sc.next();’
// EL método MENU LO PONGO PARA GUIAROS SIMPLEMENTE
char menu (){
      char operacion;
      cout <<"¿Que operacion deseas realizar?\n";
      cout <<"Pulsa + para sumar\n";
      cout <<"Pulsa - para restar\n";
      cout <<"Pulsa * para multiplicar\n";
      cout <<"Pulsa / para dividir\n";
      cout <<"Pulsa z cuando quieras apagar la calculadora\n";
      cin>> operacion;
      return operacion;
}

//hay que hacerlo de este
void calculadora (){
      char operacion;
      operacion=menu ();
      while (operacion!='z') {
            switch (operacion) {
                  case '+':
                        suma ();
                        break;
                  case '-':
                        resta ();
                        break;
                  case '*':
                        multi ();
                        break;
                  case '/':
                        div ();
                        break;
            }
            menu ();
      }
}
```

5 Pares Y sumas

```
main(){
	cout <<"Introduce un valor para mostrar sus valores pares y la suma:\n";
	cin>> num;

	paresYSumas();
}


void paresYSumas(int num){

	int cadena[num];

	cout <<"Dime "<<num<<" numeros\n";

	for (int i=0; i<num; i++){

		cin>> cadena[i];

	}

	int suma=0;

	cout <<"Los numeros pares son: \n";

	for (int i=0; i<num; i++){

		if(cadena[i]%2==0){

			cout <<cadena[i]<<"\n";

			suma=suma+cadena[i];

		}

	}

	cout <<"La suma es: "<<suma<<"\n";



}
```

6 SUMA DE SUBCONJUNTOS

De los dos siguientes no hace falta que me pongais casos de prueba

suma de subconjuntos

```
List<Integer> suma0(List<Integer> set) {
   BigInteger max = BigInteger.ONE.shiftLeft(set.size());
   BigInteger mask = BigInteger.ONE;
   while (mask.compareTo(max) < 0) {
      List<Integer> subset = new ArrayList<>();
      int s = 0;
      for (int i = 0; i < set.size(); i++)
         if (mask.testBit(i)) {
            s += set.get(i);
            subset.add(set.get(i));
         }
      if (s == 0)
         return subset;
      mask = mask.add(BigInteger.ONE);
   }
   return Collections.emptyList();
}
```

7: programa que lee por teclado un número entero positivo. Si el valor introducido no es un número entero positivo se muestra un mensaje y se vuelve a pedir.

```
import java.util.Scanner;
public class Ejemplo2Scanner {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N;
        do {
            System.out.print("Introduce un número entero positivo: ");
            while (!sc.hasNextInt()) {
                System.out.println("Valor no válido");
                sc.next();
                System.out.print("Introduce un número entero positivo: ");
            }
            N = sc.nextInt();
            if(N <= 0){
              System.out.println("El número debe ser positivo");
            }
        } while (N <= 0);
        System.out.println("Número introducido: " + N);
    }
}
```

CAJA NEGRA

Realizad los ejercicios como los casos resueltos en clase, para facilitar la comprensión y seguimiento poned el número de prueba al que haceis referencia(clase correcta/incorrecta que probaréis en cada caso) en el campo “clases cubiertas”
Ejemplo:

Asume		Condición	Clases correctas	Clases erróneas
	A	N° de parámetros	{ n = 1 } **1**	{ n < 1 } **2.1** { n > 1 } **2.2**
	B	Tipo de los parámetros	{ x ∈ N } **3**	{ x ∉ N } **4**
	C	Intervalo	{ x > 9 , x < 1000000000 } **5**	{ x < 10 } **6.1** { x > 999999999 } **6.2**

	Entradas	Salidas	Clases Cubiertas
Clases correctas	(456248)	842654	1 , 3 , 5
Clases erróneas	( )	Error	2.1
	(2148 , 215879 , 345872)	Error	2.2
	(254.3689)	Error	4
	(‘z’)	Error	4
	(8)	Error	6.1
	(1000000001)	Error	6.2

EJERCICIO 1.
Una empresa distribuidora de artículos de limpieza paga a los vendedores, además de su sueldo básico, un monto variable en función de su desempeño mensual.
Los artículos se agrupan en dos líneas de productos: A y B. Por la venta de los primeros corresponde abonarles una comisión del 6% sobre las ventas, y, por los últimos, un 3%.
Estas comisiones se verán reducidas por las inasistencias de cada vendedor. En un 10% si faltara 1 día, en un 20%, si faltara 2 y en un 30%si faltara 3 o más días durante ese mes.
Además, la comisión se reducirá si no cumplen con el promedio diario de visitas planificadas. Si hicieran entre 10 y 20% menos de visitas, se verá reducida en un 15%, y si hicieran más del 20%de visitas menos, se reducirá en un 25%.
Se pide: representar mediante tablas de decisión el proceso que permita calcular la comisión a pagar a cada vendedor.

EJERCICIO 2.
La empresa QUIMICA SA comercializa sus productos químicos a dos clases de clientes: productores industriales y distribuidores.
Es política de la empresa otorgarles bonificaciones de acuerdo al monto de ventas de cada factura.
Si el cliente es productor industrial y el pedido es igual o superior a $ 1.000 pero inferior a $ 5.000, tendrá una bonificación del 5%. Si el cliente es distribuidor y el pedido es igual o superior a $ 5.000, pero inferior a $ 20.000, gozará de un descuento del 8%. Si el cliente es industrial y el pedido es inferior a $ 1.000, no se efectuará ninguna bonificación, al igual que si el cliente es distribuidor y el pedido es inferior a $ 5.000. Si el cliente es distribuidor y el pedido es mayor o igual a $ 20.000, se lo bonificará con un 15%. Si el cliente es industrial y el pedido es igual o mayor a $ 5.000 tendrá una bonificación del 10%.
Se pide: reflejar estas reglas de decisión en una tabla de decisión.

EJERCICIO 3.
La empresa WTC SA vende libros, entregándolos en el domicilio del cliente y cargándoles el costo del despacho en la factura respectiva.
Los despachos pueden ser aéreos o terrestres, dependiendo de la urgencia del cliente y la disponibilidad de medios de transporte.
Por vía aérea la tarifa es de $ 2 por kg., hasta 10 kgs. de peso, y de $ 1,50 por cada kilo excedente. Si el envío es por tierra la tarifa es de $ 1,50 por kg. hasta 10 kgs. de peso, y de $ 1 por cada kilo excedente. El cargo mínimo por vía aérea es de $ 4,00, y de $ 3,00 si es por tierra. En los envíos terrestres a más de 200 kms. El cargo por kg. y el mínimo se incrementan en un 20%.
Se pide: confeccionar la tabla de decisión que permitan calcular el cargo por despacho.

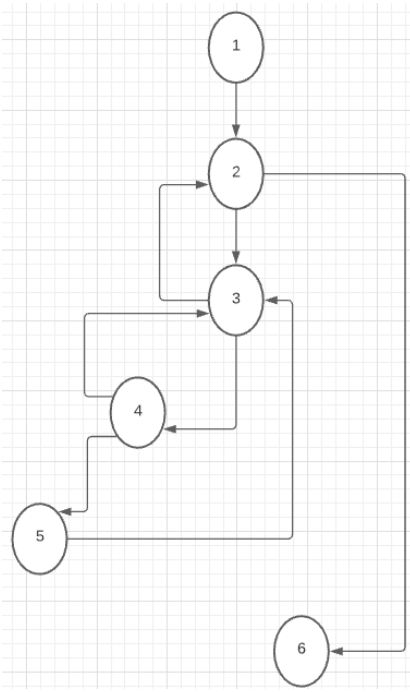
EJERCICIO 4.
El Sector Programación de Stocks realiza diariamente el siguiente proceso para la reposición de los materiales que se almacenan en sus depósitos.
Clasifica los materiales en A, B y C de acuerdo al valor de las existencias al final del año anterior. Los clasificados como A son los de mayor valor y los C, los de menor valor. También se los clasifica de acuerdo a su incidencia en el proceso productivo en materiales “críticos” y “no críticos”.
Los artículos identificados como A se piden cuando el stock actual es menor o igual al punto de pedido. Para ello se calcula el Lote Económico y se solicita esa cantidad al Sector Compras.
De los materiales B se pide, el último día de cada mes, la cantidad que falta

para llegar al Stock Máximo, más lo que se prevé consumir durante el período de reaprovisionamiento. Previamente debe calcularse el Stock Máximo, multiplicando el consumo diario por 30.
Los materiales C se piden cada fin de bimestre calendario. La cantidad a solicitar es la que falta para llegar al Stock Máximo, más lo que se estima consumir durante el período de reaprovisionamiento. Antes se calcula el Stock Máximo (consumo diario por 60).
Si los materiales son críticos (ya sean A, B o C) se los trata como si fueran A, debiendo colocarse además en la Solicitud de Compra un sello con la leyenda "Material Crítico".
Se pide: confeccionar una tabla de decisión que permita determinar la cantidad de materiales a pedir.

EJERCICIO 5.
El Concejo Deliberante de una ciudad ha aprobado una moratoria para el pago de las tasas adeudadas por sus contribuyentes desde 1996, la cual no excederá de 48 cuotas mensuales y prevé un régimen de condonación de intereses.
Los intereses de la deuda serán del 0,6 % mensual directo y se condonarán en un 100%si el pago es al contado, en un 30% si se paga hasta en 30 cuotas.
No habrá descuentos para planes superiores a 30 cuotas, excepto que el contribuyente sea jubilado o pensionado, en cuyo caso tendrá un descuento del 15%.
Para el cálculo de las cuotas se aplicará sobre la deuda total (tasa más recargo financiero) un interés del 0,8% mensual directo si opta por pagar en 2 a 12 cuotas; del 1% mensual, si opta por 13 a 24 cuotas; del 1,2%, si opta por 25 a 36 cuotas y del 1,4%, para 37 a 48 cuotas. Si el contribuyente es jubilado o pensionado, se le cobrará el 50% de la tasa de interés. Si la cancelación es al contado sólo se le cobrará la deuda total.
Se pide: confeccionar la tabla de decisión que permita obtener el valor de la cuota a cobrar al contribuyente que se adhiera a la moratoria.

EJERCICIOS CAJA BLANCA;

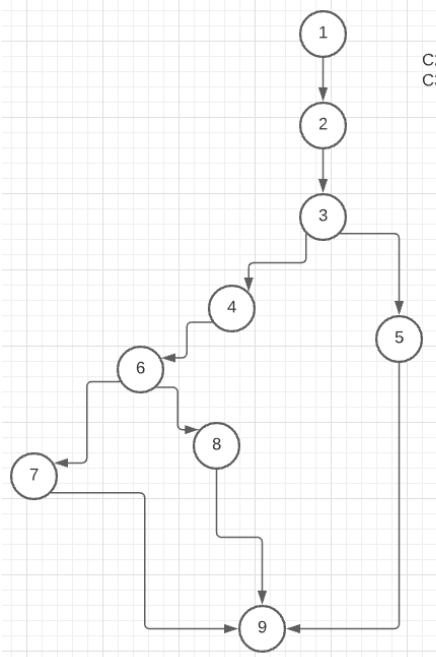
1.



CC=8-6+2=4
C1 --> 1-2-6
C2 --> 1-2-3-2
C3 --> 1-2-3-4-3
C4 --> 1-2-3-4-5-3

```
1  public static void burbuja(int[] A) {
       int i, j, aux;
2      for (i = 0; i < A.length - 1; i++) {
3          for (j = 0; j < A.length - i - 1; j++) {
4              if (A[j + 1] < A[j]) {
5                  aux = A[j + 1];
                   A[j + 1] = A[j];
                   A[j] = aux;
               }
           }
       }
6  }
```

2.

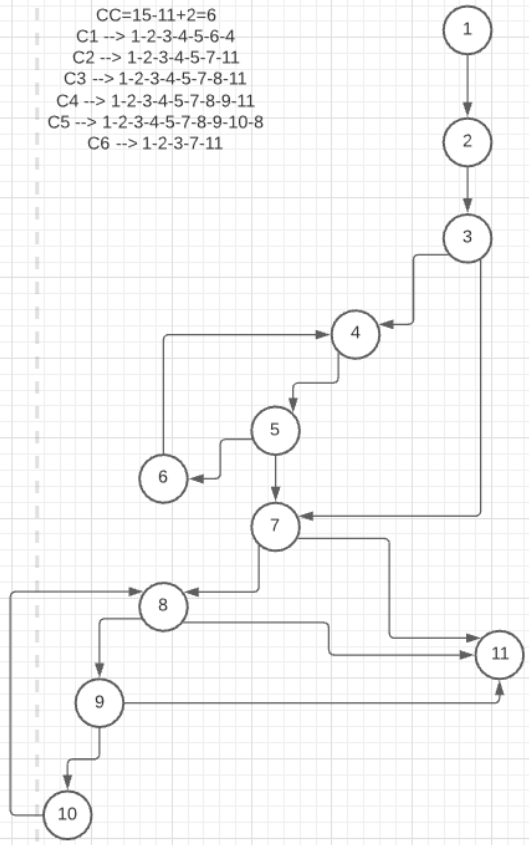


CC=10-9+2=3
C1 --> 1-2-3-5-9
C2 --> 1-2-3-4-6-8-9
C3 --> 1-2-3-4-6-7-9

```
1  bool esMovimiento(string orden, Movimiento &mov)
   {
2      bool ret;
       char v0,v1,v2,v3;
3      if(orden.length()==4)
       {
           v0=orden[0];
           v1=orden[1];
           v2=orden[2];
           v3=orden[3];

4          Coordenada orig;
           Coordenada dest;

           orig=convertirCoordenada(v0,v1);
                dest=convertirCoordenada(v2,v3);
           mov.origen.x=orig.x;
           mov.origen.y=orig.y;
           mov.destino.x=dest.x;
           mov.destino.y=dest.y;
6          if(orig.x!=-1 && dest.x!=-1)//si no sale fuera del tablero
           {
7              ret= true;
           }
           else // se va del tablero
8          {
                   ret=false;
           }
       }
       else
5          ret= false;

9      return ret;
   }
```

3.

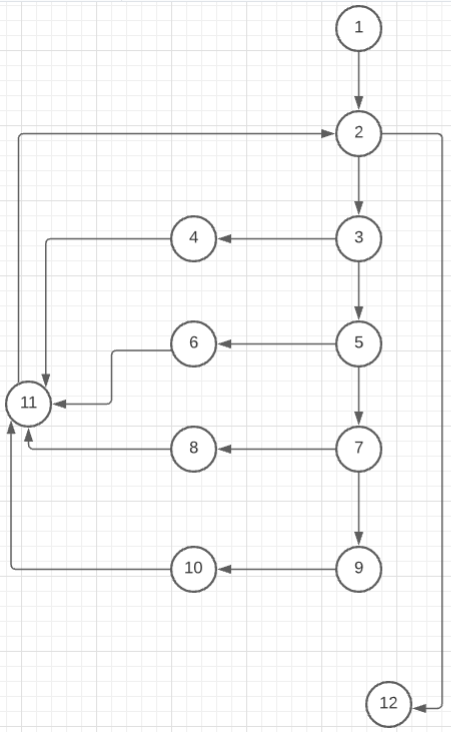
CC=15-11+2=6
C1 --> 1-2-3-4-5-6-4
C2 --> 1-2-3-4-5-7-11
C3 --> 1-2-3-4-5-7-8-11
C4 --> 1-2-3-4-5-7-8-9-11
C5 --> 1-2-3-4-5-7-8-9-10-8
C6 --> 1-2-3-7-11




```
1  {  bool movtorre(int x1, int x2, int y1, int y2,Partida &partida)
   {
2  {     bool esValido=false;
         int i;
         esValido=true;
3  {     if(x1==x2)//fichas x el camino(columna)
          {
4  {           for(i=menor(y1,y2)+1;i<mayor(y1,y2);i++)
            {
5  {              if(partida.tablero[x1][i].tipo!=VACIA)
                 esValido=false;
6  {           }
          }
7  {     if(y1==y2)//fichas x el camino(fila)
          {
8  {           for(i=menor(x1,x2)+1;i<mayor(x1,x2);i++)
            {
9  {              if(partida.tablero[i][y1].tipo!=VACIA)
               esValido=false;
10 {           }
          }

11 {  return esValido;
   }
```

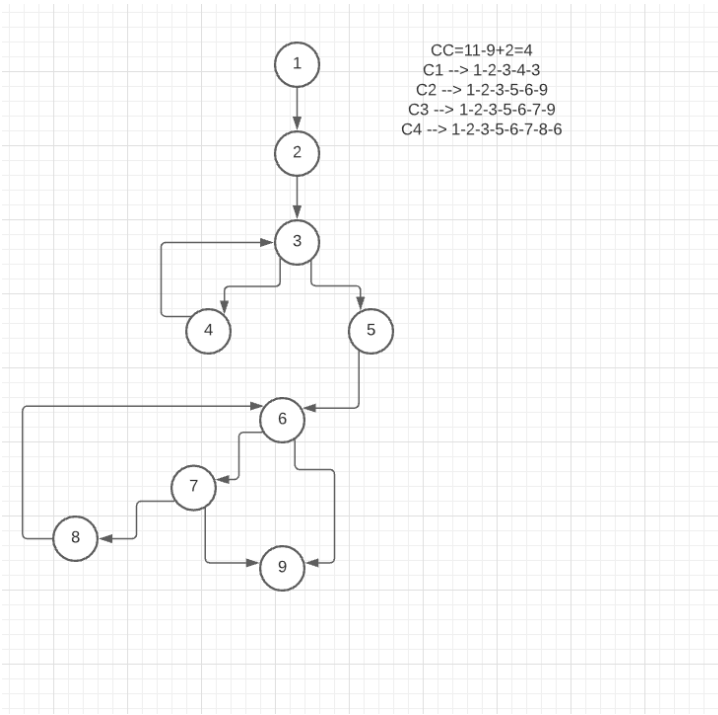
4.

CC=15-12+2=5
C1 --> 1-2-12
C2 --> 1-2-3-4-11-2
C3 --> 1-2-3-5-6-11-2
C4 --> 1-2-3-5-7-8-11-2
C5 --> 1-2-3-5-7-9-10-11-2




```
   //hay que hacerlo de este
   void calculadora (){
1  {     char operacion;
         operacion=menu ();
2  {     while (operacion!='z') {
                switch (operacion) {
3  {              case '+':
4  {                    suma ();
                        break;
5  {              case '-':
6  {                    resta ();
                        break;
7  {              case '*':
8  {                    multi ();
                        break;
9  {              case '/':
10 {                    div ();
                        break;
               }
11 {           menu ();
          }
12 {  }
```

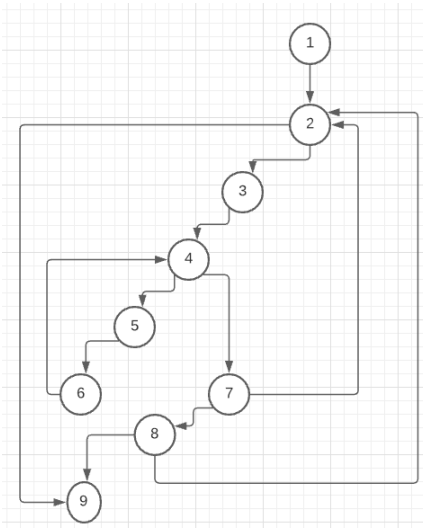
5.



CC=11-9+2=4
C1 --> 1-2-3-4-3
C2 --> 1-2-3-5-6-9
C3 --> 1-2-3-5-6-7-9
C4 --> 1-2-3-5-6-7-8-6

```
main(){
1   {   cout <<"Introduce un valor para mostrar sus valores pares y la suma:\n";
        cin>> num;
        paresYSumas();
    }

    void paresYSumas(int num){
2   {   int cadena[num];
        cout <<"Dime "<<num<<" numeros\n";
3   {   for (int i=0; i<num; i++){
4   {       cin>> cadena[i];
        }
5   {   int suma=0;
        cout <<"Los numeros pares son: \n";
6   {   for (int i=0; i<num; i++){
7   {       if(cadena[i]%2==0){
8   {           cout <<cadena[i]<<"\n";
                suma=suma+cadena[i];
            }
        }
9   {   cout <<"La suma es: "<<suma<<"\n";

    }
```

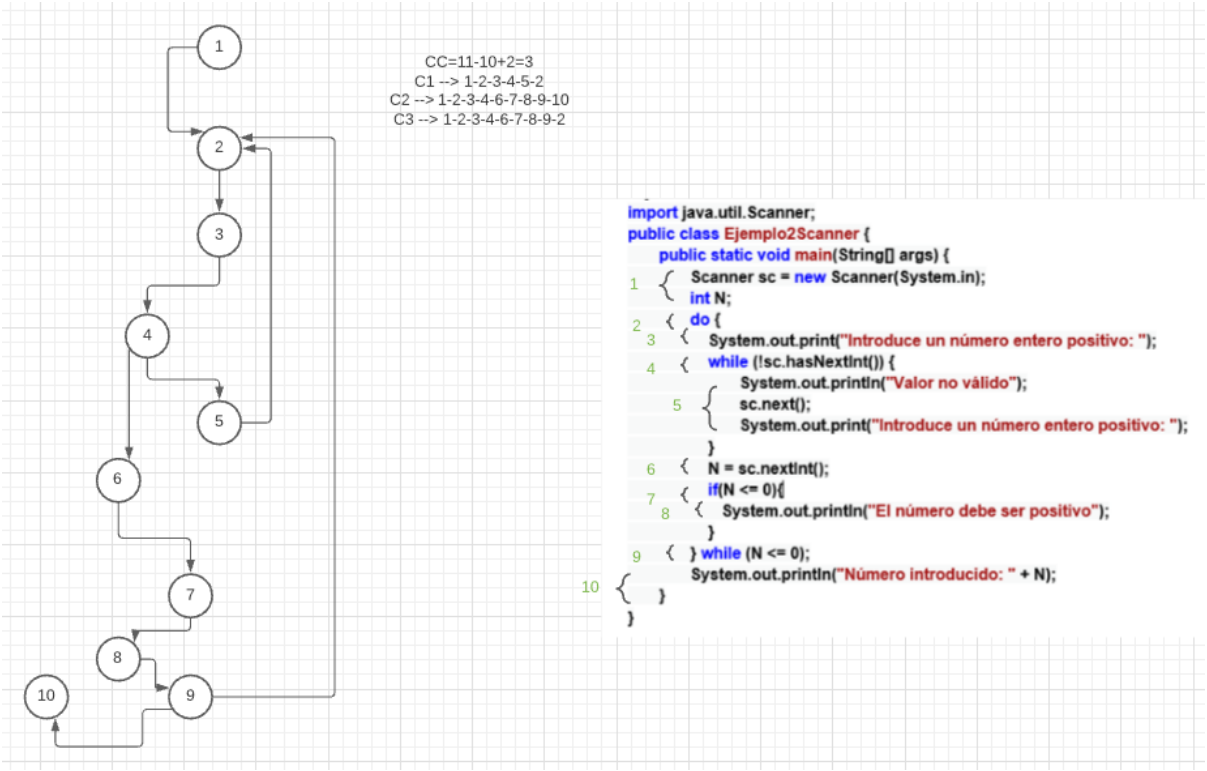
6.



CC=12-9+2=5
C1 --> 1-2-9
C2 --> 1-2-3-4-5-6-4
C3 --> 1-2-3-4-7-2
C4 --> 1-2-3-4-7-8-2
C5 --> 1-2-3-4-7-8-9

```
  List<Integer> suma0(List<Integer> set) {
1 {  BigInteger max = BigInteger.ONE.shiftLeft(set.size());
     BigInteger mask = BigInteger.ONE;
2  { while (mask.compareTo(max) < 0) {
3  {  List<Integer> subset = new ArrayList<>();
     int s = 0;
4  {  for (int i = 0; i < set.size(); i++)
5  {    if (mask.testBit(i)) {
6  {      s += set.get(i);
         subset.add(set.get(i));
       }
7  {  if (s == 0)
         return subset;
8  {  mask = mask.add(BigInteger.ONE);
     }
9 {  return Collections.emptyList();
   }
```

7.


1
2
3
4
5
6
7
8
9
10
CC=11-10+2=3
C1 --> 1-2-3-4-5-2
C2 --> 1-2-3-4-6-7-8-9-10
C3 --> 1-2-3-4-6-7-8-9-2
import java.util.Scanner;
public class Ejemplo2Scanner {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
int N;
do {
System.out.print("Introduce un número entero positivo: ");
while (!sc.hasNextInt()) {
System.out.println("Valor no válido");
sc.next();
System.out.print("Introduce un número entero positivo: ");
}
N = sc.nextInt();
if(N <= 0){
System.out.println("El número debe ser positivo");
}
} while (N <= 0);
System.out.println("Número introducido: " + N);
}
}

EJERCICIOS CAJA NEGRA;

1.

Condiciones																								
linea Prod	A	A	A	B	B	B	A	A	A	B	B	B	A	A	A	B	B	B	A	A	A	B	B	B
faltas	1	2	3	1	2	3	1	2	3	1	2	3	0	0	0	0	0	0	1	2	3	1	2	3
promedio	20,00%	20,00%	20,00%	20,00%	20,00%	20,00%	10-20%	10-20%	10-20%	10-20%	10-20%	10-20%	-10,00%	10-20%	20,00%	-10,00%	10-20%	20,00%	-10,00%	-10,00%	-10,00%	-10,00%	-10,00%	-10,00%
Acciones																								
6%	X	X	X				X	X	X				X	X	X				X	X	X			
3%				X	X	X				X	X	X				X						X	X	X
-10%	X			X			X			X									X			X		
-20%		X			X			X			X									X			X	
-30%			X			X			X			X									X			X
-15%							X	X	X	X	X	X		X			X								
-25%	X	X	X	X	X	X									X			X						
	6%	6%	6%	3%	3%	3%	6%	6%	6%	3%	3%	3%	6%	6%	6%	3%	3%	3%	6%	6%	6%	3%	3%	3%
	-10%	-20%	-30%	-10%	-20%	-30%	-10%	-20%	-30%	-10%	-20%	-30%	0	0	0	0	0	0	-10%	-20%	-30%	-10%	-20%	-30%
	-25%	-25%	-25%	-25%	-25%	-25%	-15%	-15%	-15%	-15%	-15%	-15%	0	-15%	-25%	0	-15%	-25%	0	0	0	0	0	0
	3,90%	3,30%	2,70%	1,95%	1,65%	1,35%	4,50%	3,90%	3,30%	2,25%	1,95%	1,65%	6,00%	5,10%	4,50%	3,00%	2,55%	2,25%	5,40%	4,80%	4,20%	2,70%	2,40%	2,10%

2.

CONDICIONES								
tipo cliente	industrial	industrial	industrial	industrial	distribuidor	distribuidor	distribuidor	distribuidor
pedido	-1000	1000-5000	5000-20000	20000	-1000	1000-5000	5000-20000	20000
ACCIONES								
0%	X				X	X		
5%		X						
8%							X	
15%								X
10%			X	X				

3.

despachos	aereo	tierra										
peso	menos de 10 Kg	mas de 10 Kg										
Distancia	200 Km o menos	mas de 200Km			cant kilos							
					kilos excedentes							
CONDICIONES												
despachos	aereo	aereo	aereo	aereo	tierra	tierra	tierra	tierra	aereo	aereo	terrestre	terrestre
peso	menos de 10 Kg	menos de 10 Kg	mas de 10 Kg	mas de 10 Kg	menos de 10 Kg	menos de 10 Kg	mas de 10 Kg	mas de 10 Kg	-	-	-	-
Distancia	-	-	-	-	200 Km o menos	mas de 200Km	200 Km o menos	mas de 200Km	-	-	200 Km o menos	mas de 200Km
minimo	si	si	si	si	si	si	si	si	no	no	no	no
ACCIONES												
tarifa basica	2	2	2	2	1,5	1,5	1,5	1,5	2	2	1,5	1,5
tarifa adicional	-	-	1,5	1,5	-	-	1	1				
cargo	-	-	-	-	-	20%	-	20%	-	-	-	20%
Precio total	2*cant	2*cant	20+1,5*cant ex	20+1,5*cant ex	1,5*cant	1,8*cant	15+1*cant ex	18+1,2*cant ex	4	4	3	3,6

4.

		tipo material	a, b y C		material	critico		peticion	stockmedio	Lote economico
CONDICIONES						no critico			ult dia mes	llenarStock+ estimación
									fin de bimestre	llenar stock+estimación
tipo material	A	A	B	B	C	C				
Critico	SI	NO	SI	NO	SI	NO				
ACCIONES										
Calcular stock max B (30 dias)				X						
Calcular stock max C (60 dias)						X				
leyenda	si		si		si					
peticion	lote econnomico	lote econnomico	lote econnomico	ult dia mes	lote econnomico	ult dia bimens				
							Dejar solo uno de ellos			

5.

					intereses	0,6		condonacion	contado			contribuyente	jubilado		
						0,8			30 cuotas				no jubilado		
						1			mas de 30						
						1,2									
						1,4		tipo cuota	2-12						
									13-24						15
									25-36						50
									37-48						
									cancelacion	si					
										no					
		deuda													
	CONDICIONES														
intereses															
	tipo pago		contado	contado	30 cuotas	30 cuotas	mas de 30	mas de 30							
	contribuyente		jubilado	no jubilado	jubilado	no jubilado	jubilado	no jubilado							
	ACCIONES														
	interes		0,6%	0,6%	0,6%	0,6%	0,6%	0,6%							
	condonacion		100%	100%	30%	30%	0	0							
	15% desc		15%		15%		15%								
			sin cuotas	sin cuotas	45% sobre el 0,6	30% sobre el 0,6	15% sobre el 0,6	0,60%							
		cuotas													
	CONDICIONES														
	tipo de cuota		2-12	2-12	13-24	13-24	25-35	25-35	37-48	37-48					
	jubilado		si	no	si	no	si	no	si	no					
	ACCIONES														
	intereses		0,8	0,8	1	1	1,2	1,2	1,4	1,4					
	descuento		50%		50%		50%		50%						
			0,4	0,8	0,5	1	0,6	1,2	0,7	1,4					