

Ejercicio 1

Dado el siguiente pseudocódigo:

PROGRAMA CAJA-BLANCA

DECLARACIONES

Entero: I, J, COCIENTE, RESTO, PASO

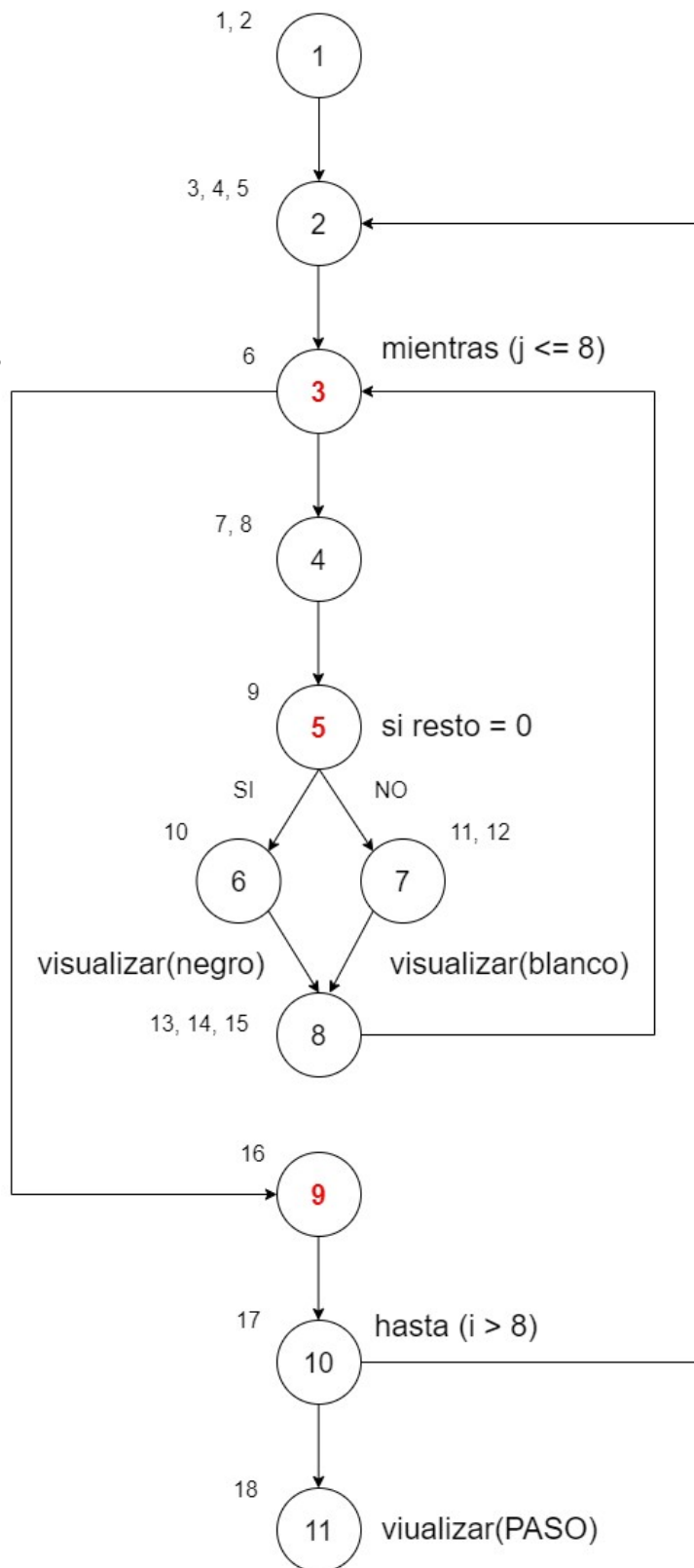
INICIO

1. I = 0
2. PASO = 0
3. **Repetir**
4. J = 1
5. PASO = PASO + 1
6. **Mientras (J <= 8) hacer**
7. COCIENTE = (I + J) / 2
8. RESTO = (I + J) - COCIENTE * 2
9. Si RESTO = 0 entonces
10. Visualizar (negro)
11. Sino
12. Visualizar (blanco)
13. Fin si
14. J = J + 1
15. **Fin-Mientras**
16. I = I + 1
17. **Hasta (I > 8)**
18. Visualizar (PASO)

Fin-programa

Realizar lo siguiente

1. Dibuja el grafo de flujo, indica el sentido de las flechas en las aristas, indica también el Sí y el No en los nodos predicado, dentro de cada nodo indica el número de línea o líneas del pseudocódigo que están incluidas.



2. Calcula la complejidad ciclomática de las tres formas conocidas, indicando cómo se calcula el valor.

- $\text{Aristas} - \text{Nodos} + 2 \rightarrow 13 - 11 + 2 = 4$
- $\text{Nodos Predicados} + 1 \rightarrow 3 + 1 = 4$
- $\text{Regiones} \rightarrow 4$

3. Evalúa el riesgo.

El módulo tiene tres caminos independientes, por lo tanto:

- Bajo riesgo de errores
- Código moderadamente fácil de entender, probar y mantener
- Ideal para pruebas unitarias estructuradas

4. Determina los caminos independientes.

- $1 \rightarrow 2 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11$
- $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 8 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11$
- $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 7 \rightarrow 8 \rightarrow 3 \rightarrow 9 \rightarrow 10 \rightarrow 11$

5. Calcular los casos de prueba que resultan para cada camino.

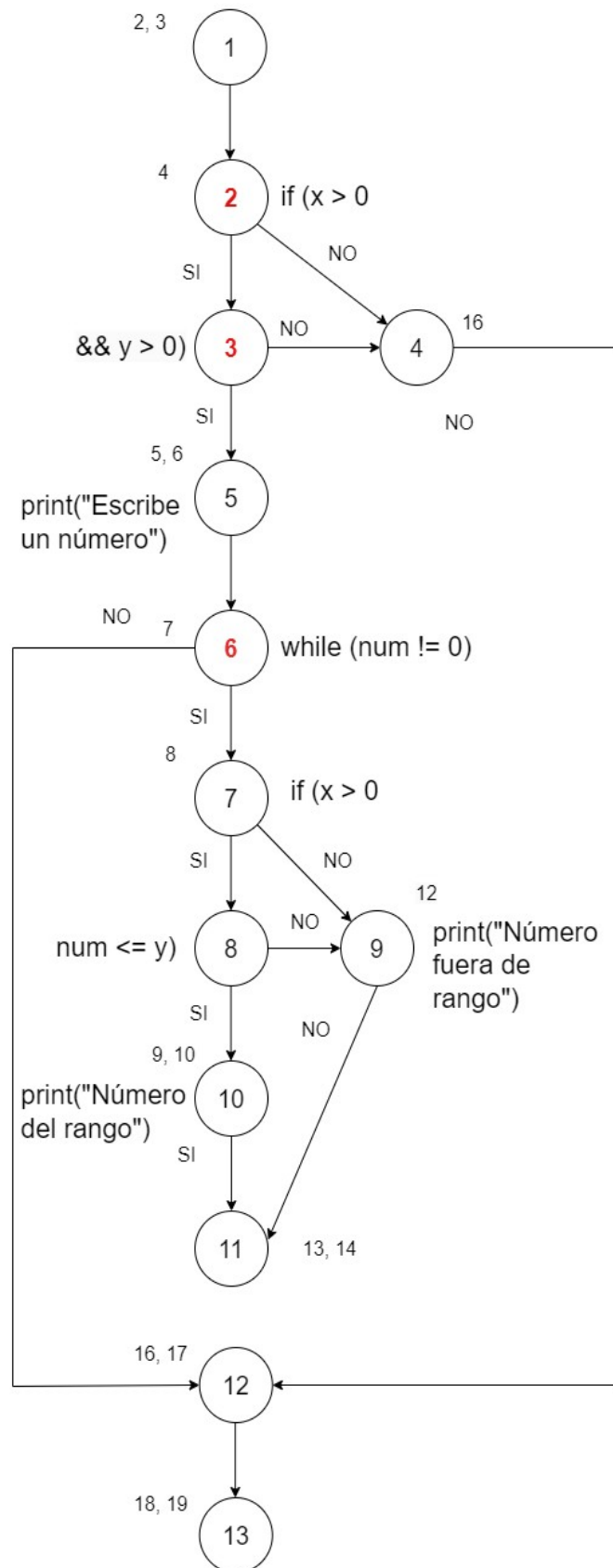
Caso de prueba	Entrada	Condición a evaluar	Camino cubierto	Resultado esperado
CP 1	2, 4, 3	$(a > 1)$	1-2-7-8	$x = x + 1$
CP 2	1, 6, 3	$(a \leq 1) \ \&\& \ (b > 5)$	1-2-3-7-8	$x = x + 1$
CP 3	1, 4, 1	$(a > 1) \ \&\& \ !(b > 5) \ (a \leq 1) \ \&\& \ (b \leq 5) \ \&\& \ (c < 2)$	1-2-3-5-7-8	$x = x + 1$
CP 4	1, 4, 3	$(a > 1) \ \&\& \ (b > 5) \ \&\& \ !(c < 2) \ (a \leq 1) \ \&\& \ (b \leq 5) \ \&\& \ (c \geq 2)$	1-2-3-5-8	$x = x - 1$

Ejercicio 2

Realiza el grafo de flujo, calcula la complejidad ciclomática, define el conjunto básico de caminos, elabora los casos de prueba para cada camino y evalúa el riesgo para la siguiente función Java:

```
01 static int Contador2 (int x, int y) {
02     Scanner entrada = new Scanner(System.in);
03     int num, c = 0;
04     if ( x > 0 && y > 0) {
05         System.out.println("Escribe un numero: ");
06         num = entrada.nextInt();
07         while (num !=0) {
08             if ( num >= x && num <= y) {
09                 System.out.println("\tNúmero en el rango");
10                 c++;
11             } else
12                 System.out.println("\tNúmero fuera de rango");
13             System.out.println("Escribe un número");
14             num = entrada.nextInt();
15         } // fin while
16     } else
17         c = -1;
18     entrada.close();
19     return c;
20 }
```

1. Dibujo del grafo de flujo.



2. Cálculo de la complejidad ciclomática.

- $\text{Aristas} - \text{Nodos} + 2 \rightarrow 15 - 12 + 2 = 5$
- $\text{Nodos Predicados} + 1 \rightarrow 4 + 1 = 5$
- $\text{Regiones} \rightarrow 5$

3. Definición del conjunto básico de caminos.

- 1-2-3-5-6-7-9-10-12-13-14
- 1-2-3-4-13-14
- 1-2-3-5-6-8-9-11-12-13-14
- 1-2-4-13-14

4. Elaboración de los casos de prueba para cada camino.

Caso de prueba	Entrada	Condición a evaluar	Camino cubierto	Resultado esperado
CP 1	1, 10	5, 0	1-2-7-8	$x = 1$
CP 2	1, 10	11, 5, 0	1-2-3-7-8	$x = 1$
CP 3	1, 10	3, 5, 7, 0	1-2-3-5-7-8	$x = 3$
CP 4	0, 5	(no importa)	1-2-3-5-8	$x = -1$

5. Evaluación del riesgo.

El módulo tiene cuatro caminos independientes, por lo tanto:

- Bajo riesgo de errores
- Código moderadamente fácil de entender, probar y mantener
- Riesgo moderado: Depende de entrada del usuario, por lo que hay posibilidad de errores si se ingresan valores no numéricos (aunque no está cubierto con manejo de excepciones).

Ejercicio 3

Dado el siguiente fragmento de programa en java:

```
Import java.io.*;

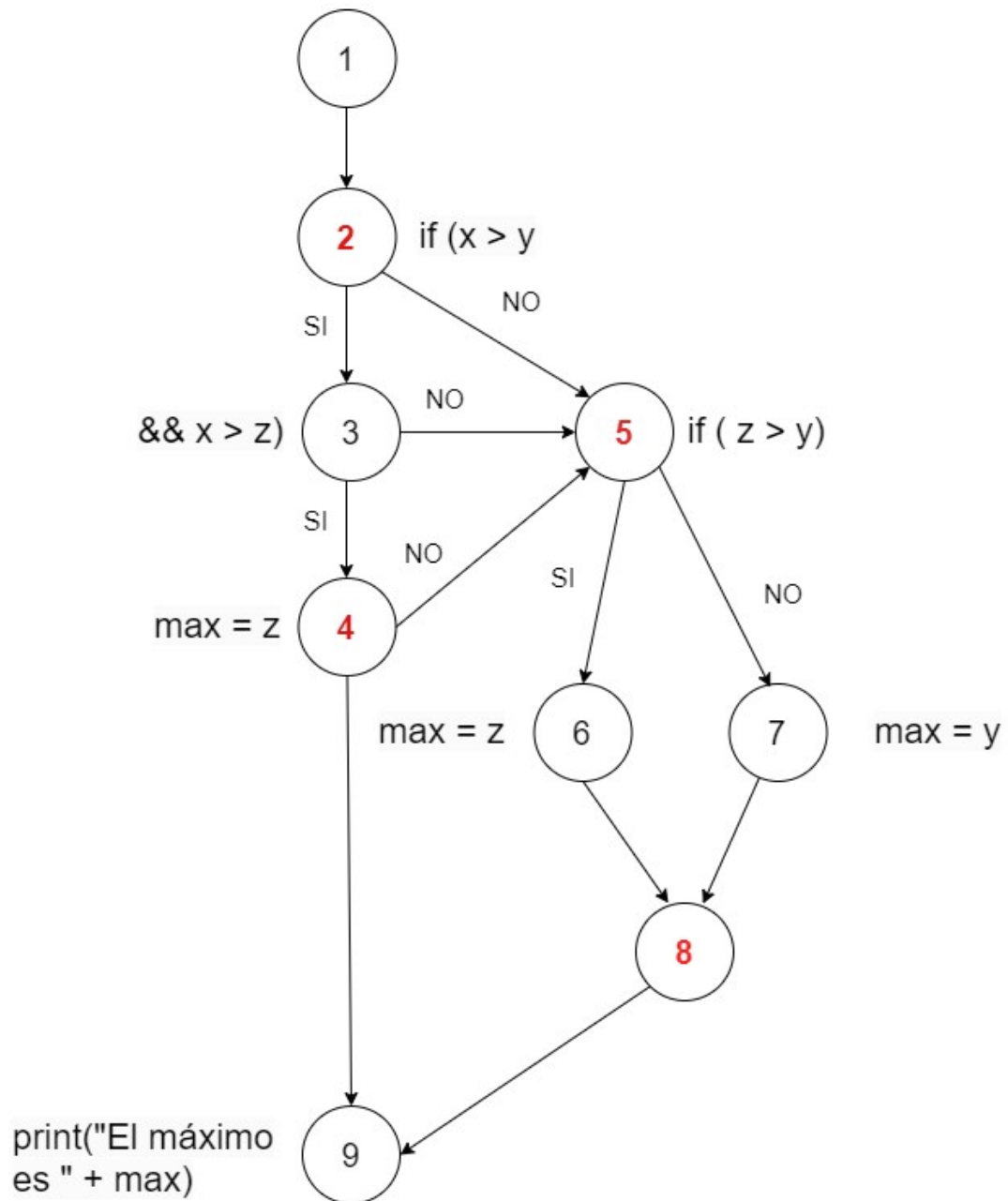
Public class Maximo
{
    public static void main (String args[]) throws IOException
    {
        BufferedReader  entrada  =  new  BufferedReader  (new
        InputStreamReader(System.in));

        Int x,y,z,max;

        System.out.println("Introduce x,y,z: ");
        x = Integer.parseInt (entrada.readLine());
        y = Integer.parseInt (entrada.readLine());
        z = Integer.parseInt (entrada.readLine());

        if (x>y && x>z)
            max = x;
        else
            if (z>y)
                max = z;
            else
                max = y;
        System.out.println ("El máximo es "+ max);
    }
}
```

Dibuja el diagrama de flujo.



Se pide a calcular la complejidad ciclomática de McCabe $V(G)$:

- $\text{Aristas} - \text{Nodos} + 2 \rightarrow 12 - 9 + 2 = 5$
- $\text{Nodos Predicados} + 1 \rightarrow 4 + 1 = 5$
- $\text{Regiones} \rightarrow 5$

Ejercicio 4

Dado el siguiente fragmento de programa en java:

If (a>1) and (b>5) and (c<2) **then**

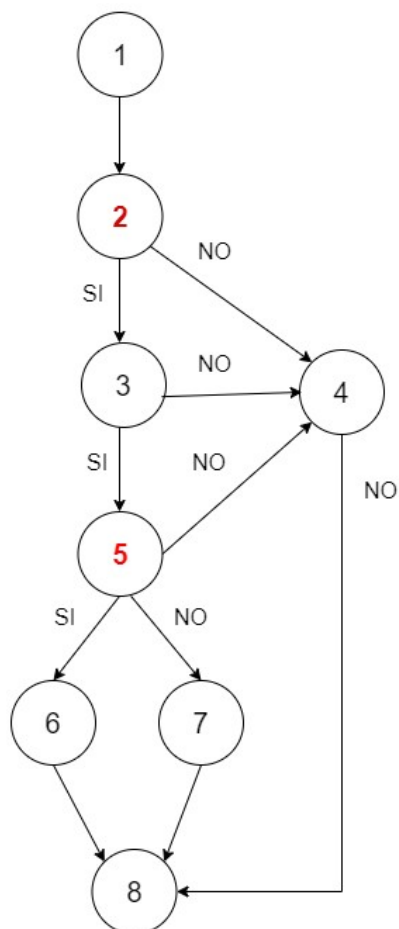
x=x+1;

else

x= x-1;

Se pide a:

1. Dibuja el grafo de flujo, indica el sentido de las flechas en las aristas, indica también el Sí y el No en los nodos predicado, dentro de cada nodo indica el número de línea o líneas del pseudocódigo que están incluidas.



2. Calcula la complejidad ciclomática de las tres formas conocidas, indicando cómo se calcula el valor.

- $\text{Aristas} - \text{Nodos} + 2 \rightarrow 10 - 8 + 2 = 4$
- $\text{Nodos Predicados} + 1 \rightarrow 3 + 1 = 4$
- $\text{Regiones} \rightarrow 4$

Tenemos 4 regiones internas por cada condición AND, y una región externa.

3. Evalúa el riesgo.

El módulo tiene tres caminos independientes, por lo tanto:

- Bajo riesgo de errores
- Código moderadamente fácil de entender, probar y mantener
- Ideal para pruebas unitarias estructuradas

4. Determina los caminos independientes.

- $1 \rightarrow 2(\text{SI}) \rightarrow 3(\text{SI}) \rightarrow 5(\text{SI}) \rightarrow 7 \rightarrow 8 \rightarrow \text{fin}$
- $1 \rightarrow 2(\text{NO}) \rightarrow 4 \rightarrow 8 \rightarrow \text{fin}$
- $1 \rightarrow 2(\text{SI}) \rightarrow 3(\text{SI}) \rightarrow 5(\text{NO}) \rightarrow 6 \rightarrow 8 \rightarrow \text{fin}$
- $1 \rightarrow 2(\text{SI}) \rightarrow 3(\text{SI}) \rightarrow 5(\text{NO}) \rightarrow 6 \rightarrow 8 \rightarrow \text{fin}$

5. Calcular los casos de prueba que resultan para cada camino.

Caso de prueba	Entrada	Condición a evaluar	Camino cubierto	Resultado esperado
CP 1	2, 6, 1	$(a > 1) \ \&\& \ (b > 5) \ \&\& \ (c < 2)$	1-2-3-5-7-8	$x = x + 1$
CP 2	1, 6, 1	$!(a > 1)$	1-2-4-8	$x = x - 1$
CP 3	2, 5, 1	$(a > 1) \ \&\& \ !(b > 5)$	1-2-3-6-8	$x = x - 1$
CP 4	2, 6, 2	$(a > 1) \ \&\& \ (b > 5) \ \&\& \ !(c < 2)$	1-2-3-5-6-8	$x = x - 1$

Ejercicio 5

Dado el siguiente fragmento de programa en java cambiando las condiciones and por or:

```
If (a>1) and (b>5) and (c<2) then
```

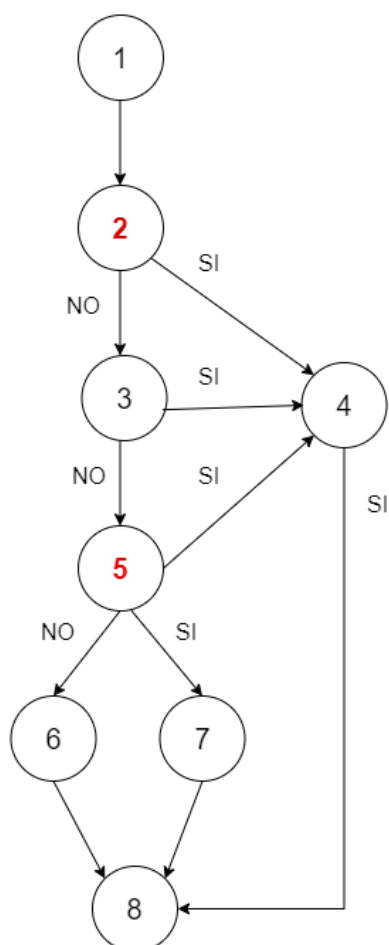
```
    x=x+1;
```

```
else
```

```
    x= x-1;
```

Se pide a:

1. Dibuja el grafo de flujo, indica el sentido de las flechas en las aristas, indica también el Sí y el No en los nodos predicado, dentro de cada nodo indica el número de línea o líneas del pseudocódigo que están incluidas.



2. Calcula la complejidad ciclomática de las tres formas conocidas, indicando cómo se calcula el valor.

- $\text{Aristas} - \text{Nodos} + 2 \rightarrow 10 - 8 + 2 = 4$
- $\text{Nodos Predicados} + 1 \rightarrow 3 + 1 = 4$
- $\text{Regiones} \rightarrow 4$

Tenemos 4 regiones internas por cada condición OR, y una región externa.

3. Evalúa el riesgo.

El módulo tiene tres caminos independientes, por lo tanto:

- Bajo riesgo de errores
- Código moderadamente fácil de entender, probar y mantener
- Ideal para pruebas unitarias estructuradas

4. Determina los caminos independientes.

- $1 \rightarrow 2(\text{SI}) \rightarrow 7 \rightarrow 8 \rightarrow \text{fin}$
- $1 \rightarrow 2(\text{NO}) \rightarrow 3(\text{SI}) \rightarrow 7 \rightarrow 8 \rightarrow \text{fin}$
- $1 \rightarrow 2(\text{NO}) \rightarrow 3(\text{NO}) \rightarrow 5(\text{SI}) \rightarrow 7 \rightarrow 8 \rightarrow \text{fin}$
- $1 \rightarrow 2(\text{NO}) \rightarrow 3(\text{NO}) \rightarrow 5(\text{NO}) \rightarrow 6 \rightarrow 8 \rightarrow \text{fin}$

5. Calcular los casos de prueba que resultan para cada camino.

Caso de prueba	Entrada	Condición a evaluar	Camino cubierto	Resultado esperado
CP 1	2, 4, 3	$(a > 1)$	1-2-7-8	$x = x + 1$
CP 2	1, 6, 3	$(a \leq 1) \ \&\& \ (b > 5)$	1-2-3-7-8	$x = x + 1$
CP 3	1, 4, 1	$(a > 1) \ \&\& \ !(b > 5) \ (a \leq 1) \ \&\& \ (b \leq 5) \ \&\& \ (c < 2)$	1-2-3-5-7-8	$x = x + 1$
CP 4	1, 4, 3	$(a > 1) \ \&\& \ (b > 5) \ \&\& \ !(c < 2) \ (a \leq 1) \ \&\& \ (b \leq 5) \ \&\& \ (c \geq 2)$	1-2-3-5-8	$x = x - 1$