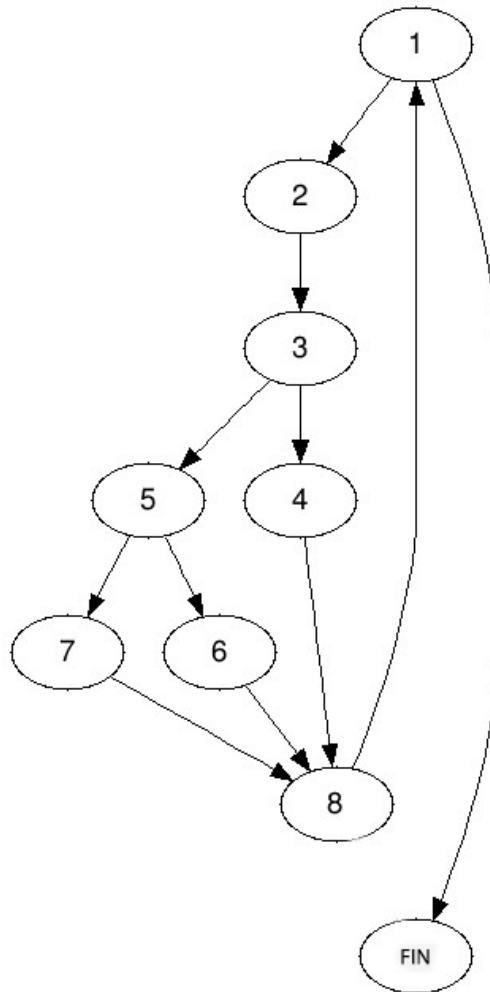


Ejercicio 1. Dado el siguiente pseudocódigo:

```
/* Algoritmo de búsqueda binaria. Datos de entrada, llave y Lista. Lmin es la longitud mínima  
y lmax la longitud del array */  
Lee llave  
x1=lmin  
x2=lmax  
resp=-1  
mientras (x1<x2-1) (1)  
    x=(x1+x2)/2 (2)  
    si (Lista[ix]==llave) entonces (3)  
        resp=ix (4)  
        x1=ix  
        x2=  
    de otro modo si (Lista[ix]<llave) entonces (5)  
        x1=ix (6)  
    de otro modo x2=ix (7)  
fin del mientras  
regresa resp (8)
```

1. Obtener el grafo de flujo relacionado al anterior pseudocódigo



2. Calcular la complejidad ciclomática

$$V(G) = 11 \text{ aristas} - 9 \text{ nodos} + 2 = 4$$

$$V(G) = n^{\circ} \text{ de regiones} + \text{exterior} = 3 + 1 = 4$$

Caminos independientes:

- Nodos: 1 - fin
- Nodos: 1 - 2 - 3 - 4 - 8 - 1 - fin
- Nodos: 1 - 2 - 3 - 5 - 6 - 8 - 1 - fin
- Nodos: 1 - 2 - 3 - 5 - 7 - 8 - 1 - fin

3. Rellenar la tabla

Camino	Dato de Prueba	Resultado esperado	Resultado real
1 - fin	x1 = 10 x2 = 5	resp = -1	resp = -1
1 - 2 - 3 - 4 - 8 - 1 - fin	x1 = 2 x2 = 10 Lista: array 10 valores del 0 al 9 llave = 6	resp = 6	resp = 6

Pruebas

1 - 2 - 3 - 5 - 6 - 8 - 1 - fin	x1 = 2 x2 = 10 Lista: array 10 valores del 0 al 9 llave = 9	resp = -1	resp = -1
1 - 2 - 3 - 5 - 7 - 8 - 1 - fin	x1 = 2 x2 = 10 Lista: array 10 valores del 0 al 9 llave = 1	resp = -1	resp = -1

Ejercicio 2. Realiza la prueba de la siguiente condición usando las tres técnicas conocidas:

```
If ( x<10 && y>20)
```

```
    Z= 7
```

```
Else
```

```
    Z = 8
```

Cobertura de decisiones:

IF verdadero: $x = 5, y = 25$

IF falso: $x = 5, y = 15$

Cobertura de condiciones:

Dos casos, en este caso los mismos que la cobertura de decisiones.

Caso 1: $x = 5, y = 25$

Caso 2: $x = 5, y = 15$

Cobertura de condiciones múltiple:

Caso 1 (V-V): $x = 5, y = 25$

Caso 1 (V-F): $x = 5, y = 15$

Caso 1 (F-V): $x = 15, y = 25$

Caso 1 (F-F): $x = 15, y = 15$

Ejercicio 3. Realiza la prueba de bucles del siguiente fragmento. La inicialización de *s* es un dato de entrada que se le pregunta al usuario:

```
do
{
    s = s+1
    Console.WriteLine("Introduce un número: ");
    numero = Int.Parse(Console.ReadLine());
    if( numero % 2 == 0)
        break;
}
while ( s < 10 )
```

a. Pasar por alto completamente el bucle.

Debido a que es un bucle do-while, se realiza una iteración de forma obligatoria.

b. Pasar sólo una vez por el bucle.

Seleccionando un valor de *s* mayor que 10. Por ejemplo, *s* = 20;

c. Pasar dos veces por el bucle.

Asigno a *s* el valor 8 antes de entrar al bucle.

Segunda iteración: *s* = 9

Fin del bucle do-while

d. Hacer *m* pasos por el bucle, donde $m < n$

Con un valor de *m* = 5. Asigno a *s* el valor 5 antes de entrar al bucle.

Primera iteración: *s* = 6

Segunda iteración: *s* = 7

Tercera iteración: *s* = 8

Cuarta iteración: *s* = 9

Quinta iteración: *s* = 10

Fin del bucle do-while

e. Hacer $n-1$, n y $n+1$ pasos por el bucle.

Suponiendo que $n = 10$,

$n-1$ pasos (9): asigno a s el valor 1 antes de entrar al bucle.

n pasos: asigno a s el valor 0 antes de entrar al bucle.

$n+1$ pasos (11): asigno a s el valor -1 antes de entrar al bucle.

Ejercicio 4. Se va a realizar una entrada de datos de un empleado por pantalla gráfica, se definen 3 campos de entrada y una lista para elegir el oficio.

La aplicación acepta los datos de esta manera:

- Empleado: número de 3 dígitos que no empiece por 0
- Departamento: en blanco o número de dos dígitos
- Oficio: Analista, Diseñador, Programador o Elige oficio

Si la entrada es correcta el programa asigna un salario (que se muestra en pantalla) a cada empleado según estas normas:

- S1 si Oficio es Analista se asigna 2500
- S2 si el Oficio es Diseñador se asigna 1500
- S3 si el Oficio es Programador se asigna 2000

Si la entrada no es correcta el programa muestra un mensaje indicando la entrada incorrecta.

- ER1 si el Empleado no es correcto
- ER2 si el Departamento no es correcto
- ER3 si no se ha elegido Oficio

Representa las clases de equivalencia para cada condición de entrada mediante una tabla,

Pruebas

Condición de entrada	Tipo	Clases válidas		Clases no válidas	
		Entrada	Código	Entrada	Código
Empleado	Número	100 <=	V1	Empleado < 100	N1
		Empleado <= 999		Empleado > 999	N2
Departamento	Número	En blanco	V2	Empleado < 10	N3
		10 <= Departamento <= 99	V3	Empleado > 99	N4
Oficio	Lista	Oficio = "Analista"	V4	No se ha elegido oficio	N5
		Oficio = "Diseñador"	V5		
		Oficio = "Programador"	V6		

Ejercicio 5. A partir de la tabla (Definición de Clases de Equivalencia) obtenida en la actividad anterior, debes obtener los casos de prueba correspondientes. Para ello, deberás de representarlos en otra tabla donde cada fila representa un caso de prueba con:

- Los códigos de las clases de equivalencia que se aplican (los que asignaste en la actividad anterior, aunque también se podría haber asignado un número a cada clase)
- Los valores asignados a las condiciones de entrada
- El resultado esperado según el enunciado del problema

Para obtener los casos de prueba deberás de tener en cuenta estas dos reglas:

- Los casos de prueba deben de cubrir la mayor cantidad de clases de equivalencia válidas (podría escribirse solo uno que cubriera todas las clases de equivalencia válidas)
- Se escribe un solo caso de prueba por cada clase de equivalencia inválido. Si se prueban múltiples clases de equivalencia no válidas en el mismo caso de prueba, pueden darse alguna de estas situaciones:
 - que algunas de estas pruebas nunca se ejecuten porque la primera enmascara a las otras o
 - que alguna de estas pruebas termina la ejecución del caso de prueba

Los casos de prueba se van añadiendo a la tabla hasta que todas las clases de equivalencia válidas y no válidas hayan sido cubiertas.

Pruebas

Clases de equivalencia	CONDICIONES DE ENTRADA			Resultado esperado
	Empleado	Departamento	Oficio	
V1, V3, V4	100	10	Analista	Salario = 2500
V1, V3, V5	998	11	Diseñador	Salario = 1500
V1, V3, V6	101	99	Programador	Salario = 2000
V1, V2, V4	500	En blanco	Analista	Salario = 2500
V1, V2, V5	999	En blanco	Diseñador	Salario = 1500
V1, V2, V6	997	En blanco	Programador	Salario = 2000
N1, V3, V4	99	100	Analista	Empleado no es correcto
N2, V2, V5	1000	En blanco	Diseñador	Empleado no es correcto
V1, N3, V6	100	9	Programador	Departamento no es correcto
V1, N4, V4	101	100	Analista	Departamento no es correcto
V1, V3, N5	101	99	Elige oficio	No se ha elegido oficio