**PREGUNTA 1 (15 + 15 = 30 Puntos) Programación**

1. *encontrado = false;*

*for(i = 0; i < k-1 && !encontrado; i++)*

*for(j = i+1; j < k && !encontrado; j++)*

*if(a[i] == a[j]) encontrado = true;*

1. *suma = 0;*

*for(i = 0; i < k; i++)*

*for(j = 0; k < k; j++)*

*suma = suma + a[i][j];*

**PREGUNTA 2 (10 + 10 = 20 Puntos) Complejidad Temporal**

* 2 datos iguales

Comparaciones: (K-1)+(K-2)+(K-3)+ … +2+1 = K x (K-1) /2 🡪 O(n2)

* Suma

Si la matriz tiene K2 datos, hay K2 sumas, si tiene 2 x K2 datos hay 2 x K2 sumas 🡪 O(n)

**PREGUNTA 3 (25 Puntos) Recursión**

*int sumaRec1(int a, int b)*

*{ if(b == 0) return a;*

*else { if(b < 0) return sumaRec1(a, b + 1) - 1;*

*else return sumaRec1(a, b - 1) + 1;*

*}*

*}*

**PREGUNTA 4 (5 + 5 + 10 + 5 = 25 Puntos) Punteros**

# Sí, mientras x es un “objeto”.

# No siempre, x puede no ser un puntero.

# Explique línea por línea:

# int dato; declaración de variable dato del tipo int

# int \*pDato; declaración de variable pDato del tipo puntero a int

# int \*pNumero; declaración de variable pNumero del tipo puntero a int

# dato = 10; a la variable dato se asigna 10

# pDato = &dato; a la variable pDato se asigna la dirección de la variable dato

# pNumero = pDato; a pNumero se asigna la dirección que almacena pDato

\*pDato += 5; el dato almacenado en la dirección que tiene pDato se

incrementa por 5

\*pNumero += 8; el dato almacenado en la dirección que tiene pNumero se

incrementa por 8

std::cout << "dato = " << dato << std::endl; se escribe dato = 23

# Se escribe:

a: 7, b: 9

a: 9, b: 7