

## Criterios de corrección del segundo parcial

### Ejercicio 1:

#### Partes a, b, c, d, e, f:

Esta parte vale 1 punto. La corrección es en general binaria (1 o 0 puntos).

#### Partes g, h:

Esta parte vale 3 puntos. La corrección es generalmente binaria (3 o 0 puntos), salvo que se detecte un error evidente en las cuentas (no así en el razonamiento).

### Ejercicio 2:

Cada variable vale 2 puntos.

Si se presenta el razonamiento y hay errores en las cuentas, se corrige como coherente c/error, pero el máximo que se puede sacar es 3 puntos.

### Ejercicio 3:

#### Parte a:

El ejercicio se resuelve con un while, ya que no se conoce a priori la cantidad de elementos iguales contiguos que van a haber al principio del vector.

Aplica los criterios generales de iteración:

Usa for para recorrer como un while y funciona correctamente: -5

Usa for para recorrer como un while y no funciona correctamente: -10

No incrementa iterador en el while: -5

Indexa mal un vector o matriz, pasándose de sus límites: -3

No corta la iteración cuando se cumple la condición y funciona correctamente: -5

No corta la iteración cuando se cumple la condición y funciona incorrectamente: -10

#### Parte b:

El ejercicio se resuelve con un while, ya que no se conoce a priori cuantos tramos de elementos iguales contiguos hay en el vector.

Aplica los criterios generales de iteración:

Usa for para recorrer como un while y funciona correctamente: -6

Usa for para recorrer como un while y no funciona correctamente: -12

No incrementa iterador en el while: -6

Indexa mal un vector o matriz, pasándose de sus límites: -3

No corta la iteración cuando se cumple la condición y funciona correctamente: -6

No corta la iteración cuando se cumple la condición y funciona incorrectamente: -12

#### Parte c:

El ejercicio se resuelve con un doble for. El primer for saca los elementos de a pares del vector de entrada, mientras que el segundo for se usa para generar las tiras de secuencias. El segundo for se puede hacer equivalentemente usando operaciones vectoriales.

Aplica los criterios generales de iteración:

Usa while para recorrer como un for: -5

Usa while para recorrer como un for, y no incrementa el contador: -12

Modifica el índice del for dentro del loop: -6

Usa el índice del for fuera del loop: -4

## **Ejercicio 4:**

### **Parte a:**

#### *Criterios específicos del problema*

No borra el elemento cuando corresponde: -6

No deja la estructura coherente (por ejemplo: borra en los datos pero no en filas y columnas): -6

Se olvida de mantener el primer elemento cuando no hay que eliminarlo: -6

#### *Criterios generales de recursión*

No usa if o elseif para separar los casos en la llamada recursiva: -12

No hay ningún paso base: -12

En la resolución recursiva utiliza variables no inicializadas, demostrando que no maneja bien el concepto de encapsulamiento ni el del alcance (scope) de las variables: -12

La llamada recursiva no asegura que la recursión termina: -12

Existe alguna llamada recursiva que no disminuye el largo del vector: -12

No se devuelve nada o algo distinto de lo esperado: -12

Recursión infinita por paso base incorrecto o inexistente: -6

Pierde el resultado de la llamada recursiva: -6

En algún caso la función no devuelve nada: -6

### **Parte b:**

#### *Criterios específicos del problema*

La recursión no tiene en cuenta el caso vacío o lo tiene en cuenta y no pone el mínimo y el máximo en 0: -3

#### *Criterios generales de recursión*

No usa if o elseif para separar los casos en la llamada recursiva: -12

No hay ningún paso base: -12

En la resolución recursiva utiliza variables no inicializadas, demostrando que no maneja bien el concepto de encapsulamiento ni el del alcance (scope) de las variables: -12

La llamada recursiva no asegura que la recursión termina: -12

Existe alguna llamada recursiva que no disminuye el largo del vector: -12

No se devuelve nada o algo distinto de lo esperado: -12

Recursión infinita por paso base incorrecto o inexistente: -6

Pierde el resultado de la llamada recursiva: -6  
En algún caso la función no devuelve nada: -6

### **Ejercicio 5:**

El ejercicio se resuelve con un doble for que va procesando cada posición de la matriz, e incrementado la cantidad de ocurrencias en función del valor.

#### *Criterios específicos del problema*

Hace un doble for para cada uno de los valores: -6  
Indexa mal un vector o matriz, pasándose de sus límites: -4

#### *Aplica los criterios generales de iteración:*

Usa while para recorrer como un for: -5  
Usa while para recorrer como un for, y no incrementa el contador: -12  
Modifica el índice del for dentro del loop: -6  
Usa el índice del for fuera del loop: -4

### **Ejercicio 6:**

#### *Criterios específicos del problema*

La recursión no tiene en cuenta el caso vacío o el caso de un elemento (es equivalente a la inexistencia de un paso base): -6  
Llama recursivamente a la función y  $v(1)$  es distinto de  $v(lv)$ , es decir que es falso (hace llamadas recursivas de más): -5

#### *Criterios generales de recursión*

No usa if o elseif para separar los casos en la llamada recursiva: -12  
No hay ningún paso base: -12  
En la resolución recursiva utiliza variables no inicializadas, demostrando que no maneja bien el concepto de encapsulamiento ni el del alcance (scope) de las variables: -12  
La llamada recursiva no asegura que la recursión termina: -12  
Existe alguna llamada recursiva que no disminuye el largo del vector: -12  
No se devuelve nada o algo distinto de lo esperado: -12  
Recursión infinita por paso base incorrecto o inexistente: -6  
Pierde el resultado de la llamada recursiva: -6  
En algún caso la función no devuelve nada: -6