

**Nombre:**

**Padrón:**

## Fundamentos de Programación

**Exámen integrador - 11/7/2024**

**Condición de aprobación:** Se deben tener al menos 2 ejercicios Bien y adicionalmente otro ejercicio con Regular o más.

### Ejercicio 1

---

#### Parte práctica

Peter Parker, a.k.a. Spiderman, quiere obtener información sobre la especie de araña que lo picó. Sólo llegó a ver sus colores y sabe que no es venenosa porque sobrevivió a su picadura.

Una araña está representada por el struct `araña_t` :

```
typedef struct color {
    char nombre[MAX_NOMBRE_COLOR];
    bool es_brillante;
} color_t;

typedef struct araña {
    char nombre[MAX_NOMBRE];
    color_t colores[MAX_COLORES];
    int cant_colores;
    bool es_venenosa;
} araña_t;
```

**Se pide:**

Dado un vector de arañas y su tope, crear una función **recursiva** que devuelva la cantidad de arañas que contengan los colores "Rojo" y "Azul" y no sean venenosas.

#### Parte teórica

1. ¿Cuándo conviene implementar un algoritmo repetitivo con una estructura de control iterativa y cuándo con recursividad?
2. ¿Qué tipo de dato es el campo **colores** del struct `araña`?

## Ejercicio 2

---

### Parte práctica

Annie y Hallie deciden realizar una broma a Meredith en el mismo hotel donde hicieron que se reencuentren sus padres (Elizabeth y Nicholas).

La broma resultará exitosa si:

- Annie y Hallie están cerca, es decir se encuentran en el mismo piso.
- No son rodeadas por sus padres, es decir su padre (Nicholas) y su madre (Elizabeth) no se encuentran en los pisos contiguos del piso en que están ellas, uno en cada uno.
- El padre (Nicholas) no ve la broma durante su ejecución, es decir Meredith no se encuentra en la misma habitación que Nicholas.

Si el hotel puede ser representado como una matriz de habitaciones donde una fila representa un piso, y una habitación puede ser representada con la siguiente estructura:

```
typedef struct habitacion{
    int numero;
    char ocupantes[MAX_OCUPANTES]; // inicial del nombre
    int cantidad_ocupantes;
} habitacion_t;
```

### Se pide

Dada una matriz de habitaciones que representa el hotel en el instante que se ejecuta la broma, crear una función que devuelva **true** si la broma resulta exitosa y **false** en caso contrario.

### Ejemplos

Con esta disposición de personas:

```
| - | - | - | - | M | - | |
|---|---|---|---|---|---|---|
| - | - | A | - | - | - | - |
| - | - | H | - | - | - | - |
| N | - | - | - | - | - | - |
| - | - | - | E | - | - | - |
```

Devolverá **false**.

Con esta disposición de personas:

```
| - | - | - - | - | - | - | |
| - | - | - - | - | - | - |  
| - | - | - - | E | - | - | - |  
| - | - | H A | - | - | M | - |  
| - | - | - - | - | - | N | - |  
| - | - | - - | - | - | - | - |
```

Devolverá **false**.

Con esta disposición de personas:

```
| - | - | - | - | - | - - | - |  
| - | - | - | - | - | - - | - |  
| - | - | - | - | - | - - | - |  
| - | - | H | A | - | - - | - |  
| - | - | - | - | - | N M | - |  
| - | - | E | - | - | - - | - |
```

Devolverá **false**.

Con esta disposición de personas:

```
| N | - | - | - | - | M | - | |
|---|---|---|---|---|---|---|---|
| - | - | A | - | - | H | - |  
| - | - | - | - | - | - | - |  
| - | - | - | - | - | - | - |  
| - | - | - | - | - | E | - | - |
```

Devolverá **true**.

## Parte teórica

1. ¿Qué son las pre condiciones de una función? ¿Qué son las post condiciones de una función? Dar ejemplos.
2. Explicar si la siguiente afirmación es Verdadera o Falsa y por qué: "Dado un vector y su tope, al acceder al elemento `vector[tope]` estoy accediendo a basura".

## Ejercicio 3

---

### Parte práctica

Linguini debe realizar un pedido de materia prima para el Restaurante de Gusteau, pero

antes debe revisar el stock actual.

El restaurante quiere tener en stock cantidad 100 de cada producto y no tienen en cuenta los productos que le queden 2 o menos días para que caduquen.

### Se pide

Dado un archivo de **stock** con el siguiente formato:

```
DIAS_PARA_CADUCAR;PRODUCTO;CANTIDAD
```

Se debe crear un programa en el lenguaje de programación **Python** que cree un archivo **pedido** con los productos y cantidades necesarias para que Linguini pueda encargarse del pedido.

Formato del archivo **pedido**:

```
PRODUCTO;CANTIDAD
```

### Aclaración

- Para el archivo **stock** se puede asumir que no hay productos repetidos.

### Ejemplo

Teniendo el archivo

**stock.csv**

```
4;YOGURT;67
1;BATATA;40
60;LEVITE NARANJA;100
10;PAPA;3
```

Se obtiene el siguiente archivo:

**pedido.csv**

```
YOGURT;33
BATATA;100
PAPA;97
```

### Parte teórica

Si el archivo de stock estuviese ordenado por nombre de producto y se recibiera un archivo de materia prima con los productos que pidió Linguini también ordenado por nombre de producto, explique detalladamente con sus palabras cómo actualizaría el stock recorriendo ambos archivos una única vez.