

# EJERCICIO 1

El invierno esta llegando a Lorien...

Y los elfos son bastante friolentos, por lo que cuando se acercan los fríos invernales, arrancan los preparativos para realizar la compra de bolsas de agua caliente.

La bolsa que tiene cada elfo, puede estar en estado **N** (como nueva), **B** (bueno), **M** (maso) o **D** (deprimente).

El super donde las compran tiene 3 medidas **S**, **M** y **L**. - Si el número de calzado es **menor a 38**, el elfo usa una bolsa talla **S**. - Si el número de calzado es **mayor o igual a 38 y menor a 41**, el elfo usa una bolsa talla **M**. - Si no, usará una bolsa talla **L**.

La semana anterior, se realizó un censo, del cual se obtuvo un vector de casas de la comunidad élfica, con la siguiente información sobre los elfos.

```
typedef struct elfo {
    char nombre[MAX_NOMBRE];
    int edad;
    int altura;
    int numero_calzado;
    char estado_bolsa;
    char color_bolsa;
} elfo_t;

typedef struct casa {
    char direccion[MAX_DIRECCION];
    elfo_t elfos[MAX_ELFOS];
    int cant_elfos;
} casa_t;
```

## Variante 1

Como la situación elfoeconómica no es buena, solo reemplazarán aquellas que estén en estado deprimente (**D**).

1. Determinar cuántas bolsas de agua caliente se comprarán, cumpliendo con la siguiente firma y sabiendo que el super tiene una cantidad limitada de bolsas por cada talla.

```
int bolsas_a_comprar(casa_t casas[MAX_CASAS], int tope_casas, int max_bolsas_s, int max_bolsas_m, int max_bolsas_l);
```

## Variante 2

Como la situación elfoeconómica es muy buena, reemplazarán todas las bolsas menos las que estén en estado como nuevas (**N**).

1. Determinar a cuántos elfos no se les comprará bolsa, cumpliendo con la siguiente firma y sabiendo que el super tiene una cantidad limitada de bolsas por cada talla.

```
int elfos_sin_bolsa_nueva(casa_t casas[MAX_CASAS], int tope_casas, int max_bolsas_s, int max_bolsas_m, int max_bolsas_l);
```

## Variante 3

Como la situación elfoeconómica es muy buena, reemplazarán todas las bolsas menos las que esten en estado como nuevas (**N**).

Cabe aclarar que cada posición del vector de casas se corresponde con la misma posición en el vector **bolsas\_por\_casa**.

Es decir que si en la casa de la posición 0, se reemplazarán 3 bolsas, **bolsas\_por\_casa** en la posición 0 deberá contener un 3 al finalizar la función.

1. Cargar en el vector recibido como parámetros, cuántas bolsas de agua caliente se necesitan en cada casa, cumpliendo con la siguiente firma.

```
void actualizar_bolsas_por_casa(casa_t casas[MAX_CASAS], int tope_casas, int bolsas_por_casa[MAX_CASAS]);
```

## Variante 4

El último año las bolsas rojas (**R**) vinieron falladas y los elfos no pudieron usarlas, es por eso que se cambiarán las que estén en estado deprimente (**D**) y las rojas (**R**) que estén en estado como nuevas (**N**).

Cabe aclarar que cada posición del vector de casas se corresponde con la misma posición en el vector **bolsas\_por\_casa**.

Es decir que si en la casa de la posición 0, se reemplazarán 3 bolsas, **bolsas\_por\_casa** en la posición 0 deberá contener un 3 al finalizar la función.

1. Cargar en el vector recibido como parámetros, cuántas bolsas de agua caliente se necesitan en cada casa, cumpliendo con la siguiente firma.

```
void actualizar_bolsas_por_casa(casa_t casas[MAX_CASAS], int tope_casas, int bolsas_por_casa[MAX_CASAS]);
```

## Variante 5

Antes de realizar el pedido, se necesita saber si realmente pueden hacer la compra, ya que el envío cuesta caro y **no pueden comprar más de 100 bolsas** *#crisis*.

Solo cambiarán las bolsas en estado deprimente (**D**) y las bolsas en estado maso (**M**).

La compra podrá hacerse solo si la cantidad de bolsas a comprar es **mayor a 20** y **menor o igual a 100**.

1. Derminar si los elfos pueden hacer la compra o no, cumpliendo con la siguiente firma.

```
bool pueden_comprar(casa_t casas[MAX_CASAS], int tope_casas);
```

## EJERCICIO 2

---

Las elecciones en el ministerio de magos son inminentes, Gandalf está dando sus primeros pasos en la política y preparó unas palabras fuertes pero ciertas, frías pero conmovedoras, confusas pero certeras...

Luego de su paso por el estrado y viendo las caras perplejas de los asistentes quiere saber cuan convincente fue.

Los magos asistentes al evento están sentados y su distribución puede verse como una matriz, donde la fila 0 corresponde a los que están sentados mas cerca del escenario.

Cada asiento está representado con la siguiente estructura:

```
typedef struct mago {
    bool hay_alguien;
    char nombre[MAX_NOMBRE];
    char vota_a; // Gandalf - Saruman - Indeciso
    int nivel; // del 1 al 5
} mago_t;
```

### Variante 1

1. Crear una función determine en cuántas filas consiguió los votos de todos.

### Variante 2

1. Crear una función que determine si convenció a mas del 50% de los magos de nivel 4 o superior.

### Variante 3

1. Crear una función determine en cuál fila tiene más votos.

### Variante 4

1. Crear una función determine si valió la pena esforzarse en semejante discurso.  
*Para él, valió la pena, si hay menos de 5 filas completamente vacías.*

### Variante 5

1. Crear una función que determine que porcentaje de magos de nivel 1 y 2 consiguió.

## EJERCICIO 3

---

### Variante 1

1. Explicar algoritmo de ordenamiento por **Inserción**.  
**Atención!** Está prohibido, en esta explicación, el uso de cualquiera de las palabras reservadas de C. Hemos dicho!
2. Ejemplificar paso a paso ordenando la siguiente palabra: **FRODO**.

### Variante 2

1. Explicar algoritmo de ordenamiento por **Selección**.  
**Atención!** Está prohibido, en esta explicación, el uso de cualquiera de las palabras reservadas de C. Hemos dicho!
2. Ejemplificar paso a paso ordenando la siguiente palabra: **BILBO**.

### Variante 3

1. Explicar algoritmo de ordenamiento por **Burbujeo**.  
**Atención!** Está prohibido, en esta explicación, el uso de cualquiera de las palabras reservadas de C. Hemos dicho!
2. Ejemplificar paso a paso ordenando la siguiente palabra: **PIPIN**.

## EJERCICIO 4

---

### Variante 1

Legolas se cansó de cortar la barba a Gimli y armó el siguiente algoritmo que lo hace por él:

```
1. void cortar_la_barba_de_gloin(int largo_actual_de_la_barba_de_gimli) {
2.   char gimli_esta_conforme = 'N';
3.   while(gimli_esta_conforme != 'S') {
4.     largo_actuale_la_barba_de_gimli--;
5.     print("Estás conforme? (S para terminar)");
6.     scanf(" %c", &gimli_esta_conforme);
7.   }
8. }
```

1. ¿Qué buenas prácticas no se están cumpliendo en el siguiente algoritmo?
2. ¿Cómo las corregiría?

### Variante 2

Legolas se cansó de cortar la barba a Gimli y armó el siguiente algoritmo que lo hace por él:

```
1. void cortar_barba(int largo_actual) {
2.   char no_esta_conforme = 'S';
3.   while(no_esta_conforme == 'S') {
4.     largo_actual--;
5.     print("Te corto mas?");
6.     scanf(" %c", &no_esta_conforme);
7.   }
8. }
```

1. ¿Qué buenas prácticas no se están cumpliendo en el siguiente algoritmo?
2. ¿Cómo las corregiría?

### Variante 3

Legolas se cansó de cortar la barba a Gimli y armó el siguiente algoritmo que lo hace por él:

```
1. int cortar_pelo(int largo_actual) {
2.   char esta_conforme = 'N';
3.   while(true) {
4.     largo_actual--;
5.     print("Estás conforme? (S para terminar)");
6.     scanf(" %c", &esta_conforme);
7.     if (esta_conforme == 'S'){
8.       return largo_actual;
9.     }
10.  }
11. }
```

1. ¿Qué buenas prácticas no se están cumpliendo en el siguiente algoritmo?
2. ¿Cómo las corregiría?

### Variante 4

Legolas se cansó de cortar la barba a Gimli y armó el siguiente algoritmo que lo hace por él:

```

1.      const char ENE = 'N';
2.      const char ESE = 'S';
3.
4.      void cortar_barba(int largo_actual) {
5.          char gimliEstaConforme = ENE;
6.          while(gimliEstaConforme != ESE) {
7.              largo_actual--;
8.              print("Estás conforme? (%c para terminar)", ESE);
9.              scanf(" %c", &gimliEstaConforme);
10.         }
11.     }

```

1. ¿Qué buenas prácticas no se están cumpliendo en el siguiente algoritmo?
2. ¿Cómo las corregiría?

## Variante 5

Legolas se cansó de cortar la barba a Gimli y armó el siguiente algoritmo que lo hace por él:

```

1. void cortar_barba(int largo_actual)
2. {
3.     int esta_conforme = 1;
4.     while(esta_conforme != 0)
5.     {
6.         int un_centimetro = 1;
7.         int dos_centimetros = 2;
8.         largo_actual = un_centimetro + un_centimetro - un_centimetro + largo_actual;
9.         print("Estás conforme? (0 para terminar)");
10.        scanf("%c", &esta_conforme);
11.    }}

```

1. ¿Qué buenas prácticas no se están cumpliendo en el siguiente algoritmo?
2. ¿Cómo las corregiría?

## EJERCICIO 5

---

Los elfos no son un pueblo de muchos amigos, pero cuando alguien es merecedor, le hacen regalos extraordinarios.

Se armó un vector con los amigos de los elfos (actuales o que alguna vez lo fueron) y los regalos que les hicieron, de cada amigo se tiene la siguiente información:

```
typedef struct elfamigo {
    char nombre[MAX_NOMBRE];
    char nombre_mejor_regalo[MAS_NOMBRE_REGALO];
    bool amigo_actualmente;
    int cantidad_regalos;
} elfamigo_t;
```

### Variante 1

1. Cree una rutina recursiva, que muestre los nombres de los que son amigos actualmente. 2. ¿Qué parámetros recibe y por qué? ¿Dónde se ven los componentes principales de la rutina recursiva?

### Variante 2

1. Cree una rutina recursiva, que muestre los nombres de aquellos en los que el nombre empieza con C.
2. ¿Qué parámetros recibe y por qué? ¿Dónde se ven los componentes principales de la rutina recursiva?

### Variante 3

1. Cree una rutina recursiva, que muestre los nombres de aquellos en los que el mejor regalo tiene más de 10 letras.
2. ¿Qué parámetros recibe y por qué? ¿Dónde se ven los componentes principales de la rutina recursiva?

### Variante 4

1. Cree una rutina recursiva, que muestre los nombres de aquellos que recibieron más de 5 regalos.
2. ¿Qué parámetros recibe y por qué? ¿Dónde se ven los componentes principales de la rutina recursiva?

### Variante 5

1. Cree una rutina recursiva, que muestre los nombres de aquellos que recibieron más de 5 regalos.
2. ¿Qué parámetros recibe y por qué? ¿Dónde se ven los componentes principales de la rutina recursiva?