

## Instrucciones

- Lea con detenimiento cada una de las actividades a realizar durante la experiencia.
- Cree un archivo con extensión **.cpp** con lo desarrollado. El nombre del archivo debe tener el siguiente formato: **TEL102\_P2\_Nombre\_Apellido.cpp**, sin tildes. (Ej. TEL102\_P2\_Nicolas\_Galvez.cpp)
- Enviar el archivo a través de la página de aula del ramo, sección “Práctico 2” hasta las 23:59:59 del día de hoy 28/09/2020 hora local continental de Chile (UTC-3).
- Trate de utilizar herramientas conocidas o aprendidas en clases. **No copie literalmente de recursos online.**
- Sea riguroso con las instrucciones de desarrollo.
- Comente adecuadamente el programa, describiendo lo que hace.
- ¡Éxito!

## 1 League of Myths

1. La empresa **Riot++**, es la creadora de uno de los videojuegos más famosos de la actualidad: League of Myths. En este juego existen varios personajes clasificados en 5 categorías: Top-Liner (T), Middle-Liner(M), Bottom-Liner (B), Support (S) y Jungle (J). Para cada uno de éstas categorías, existe una recomendación para que, al momento de jugar en línea, otro jugador pueda armar una combinación útil para el juego. Así, las relaciones pueden ser:
  - S se relaciona con T o B, y viceversa.
  - J se relaciona con M, y viceversa.

**Riot++** ha solicitado sus servicios como programador. Para eso le entrega la plantilla **p2.cpp** como base para su programa que se muestra a continuación.

*lom.cpp*

```
#include <iostream>
/* Practico 2
 * Las tildes han sido omitidas intencionalmente para evitar
 * problemas de compatibilidad
 */

struct help{
    char cat;
    help *connection =NULL;
};

/* Complete aqui su codigo para las funciones solicitadas */

int main()
{
    // Pruebe lo solicitado aca
    // Modifique los ejemplos para verificar otros casos.
    // Crear Secuencias de Categorías
    int l1=4; // Largo Categorías base
    int l2=10; // Largo Recomendaciones disponibles
    help *h1 = createSequence(l1); // Categorías bases
    help *h2 = createSequence(l2); // Recomendaciones disponible
    connectSequences(h1, l1, h2, l2); // Conexión de recomendaciones
    showHELP(h1, l1); // Mostrar recomendaciones

    return 0;
}
```

Note que la planilla contiene la estructura (**struct**) **help**, que incluye un caracter **cat** (categoría) y un puntero a un nodo de tipo **help**, vale decir, su recomendación.

1. Cree la función `help * createSequence(int length)` que recibe como entrada un entero. Esta crea un arreglo de tipo `help` de largo `length` asociado a una secuencia y retorna un puntero de tipo `help` que apunta a la primera posición del arreglo. La secuencia es solicitada al usuario por pantalla. Cada posición de la secuencia puede tomar solo valores T, M, B, S, J, como se muestra en el siguiente ejemplo.

*TEL102\_P2\_Nombre\_Apellido.cpp*

```
int main(){  
    ...  
    int l1=4; // Largo Categorías base  
    help *h1 = createSequence(l1);  
    ...  
}
```

*Salida (consola)*

```
...  
Creando Secuencia  
Ingrese cat: J  
Ingrese cat: T  
Ingrese cat: B  
Ingrese cat: S  
...
```

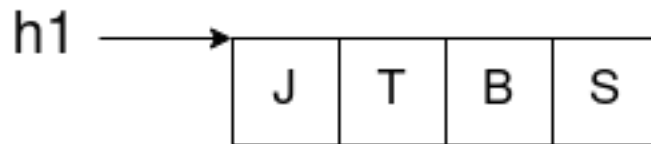


Figura 1: Esquema del arreglo generado por `createSequence()`

2. Cree la función `void connectSequences(help *h1, int l1, help *h2, int l2)` que recibe como entrada dos arreglos de tipo `help` como punteros (`h1` y `h2`) y sus respectivos largos (`l1` y `l2`), y no tiene retorno. Se toma como base la secuencia `h1` y se recorre de forma ordenada cada uno de sus elementos para encontrar una pareja compatible en `h2`, también recorrida de forma ordenada. Al encontrar una pareja, conectar los elementos respectivos en `h1` y `h2` (usando el puntero `connection` desde el elemento de `h1`). Recuerde las recomendaciones mencionadas inicialmente.

*TEL102\_P2\_Nombre\_Apellido.cpp*

```
int main(){  
    int l1=4; // Largo categorias base  
    int l2=10; // Largo Recomendaciones disponibles  
    help *h1 = createSequence(l1); // Categorias base  
    help *h2 = createSequence(l2); // Recomendaciones disponible  
    connectSequences(h1, l1, h2, l2); //Conexion de recomendaciones  
    ...  
}
```

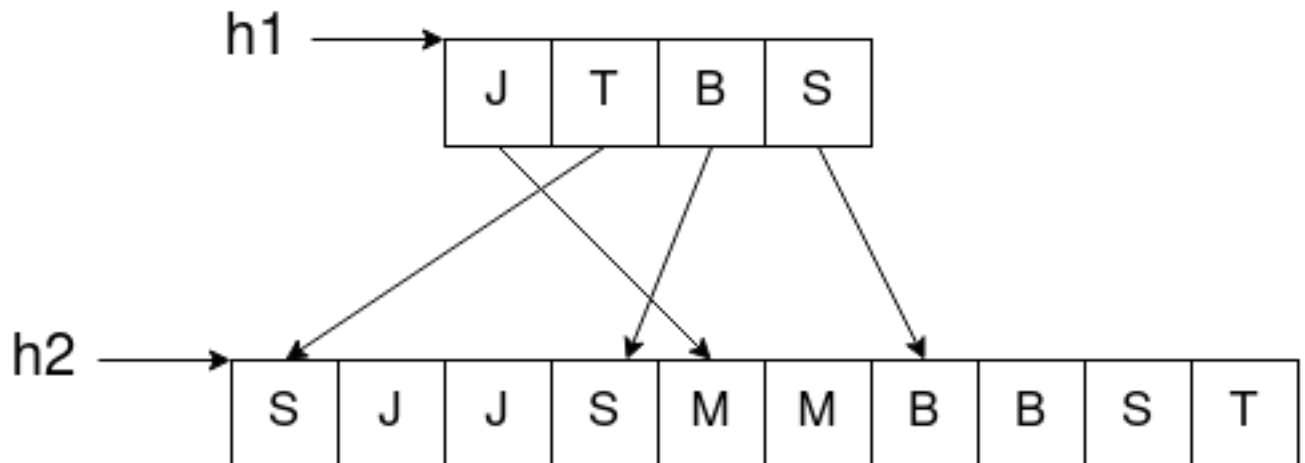


Figura 2: Esquema del resultado obtenido por `connectSequences()` para una secuencia `h2` dada.

3. Cree la función `showHELP(help *h, int length)` que recibe como entrada un puntero de tipo `help` a la secuencia base `h` y su largo `length`, y que muestra por pantalla las recomendaciones generadas a partir de `h`. Guíese por el ejemplo.

*TEL102\_P2\_Nombre\_Apellido.cpp*

```
int main(){
    int l1=4; // Largo categorias base
    int l2=10; // Largo Recomendaciones disponibles
    help *h1 = createSequence(l1); // Categorias base
    help *h2 = createSequence(l2); // Recomendaciones disponible
    connectSequences(h1, l1, h2, l2); //Conexion de recomendaciones
    showHELP(h1, l1); // Mostrar recomendaciones

    return 0;
}
```

*Salida (consola)*

```
Creando Secuencia :
Ingrese cat: J
Ingrese cat: T
Ingrese cat: B
Ingrese cat: S

Creando Secuencia :
Ingrese cat: S
Ingrese cat: J
Ingrese cat: J
Ingrese cat: S
Ingrese cat: M
Ingrese cat: M
Ingrese cat: B
Ingrese cat: B
Ingrese cat: S
Ingrese cat: T

Las recomendaciones son :
J-M
T-S
B-S
S-B
```