

Instrucciones

- Lea con detenimiento y desarrolle **individualmente** 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_P1_Nombre_Apellido.cpp** (Ej. TEL102_P1_Nicolas_Galvez.py), sin incluir tildes.
- Enviar el archivo a través de la página de aula del ramo, sección “Práctico 7” hasta las 23:59:59 del día Domingo 09/08/2020 hora local continental de Chile (UTC-4).
- Trate de utilizar herramientas conocidas o aprendidas en clases. **No copie literalmente de recursos online.**
- Sea riguroso con las instrucciones de desarrollo.
- ¡Éxito!

1 Maestro Poke++

1. *PlusPlusFreak* la empresa de videojuegos más famosa de **PlusPlus City**, lo ha contratado para realizar el primer prototipo de su nuevo juego **Maestro Poke++**. En este juego, existen criaturas llamadas **Poke**, que poseen diversos atributos, entre ellas: nombre y peso. *PlusPlusFreak* le entrega el siguiente archivo base en el cual debe trabajar:

poke_main.cpp

```
#include<iostream>
#include<cstring>

// Estructura Poke

struct poke{
    char nombre[40];
    float peso;
};

// Acá se crean las funciones

// Segmento principal
int main(){

    // Programe aquí

    return 0;
}
```

En base a esta reducida información, *PlusPlusFreakle* solicita:

1. Easy Mode: Crear la función `poke fillPoke()`, que no recibe parámetros. La función debe retornar una variable del tipo `poke`, que represente a un nuevo **Poke** y sus atributos (nombre y peso). Los atributos deben ser solicitados por entrada estándar (`std::cin`). Un ejemplo de uso se muestra a continuación:

`poke_main.cpp`

```
int main(){
    \\ ...
    poke p;
    p = fillPoke();
    \\...
}
```

Salida (consola)

```
Ingrese nombre de Poke: Pika
Ingrese peso de Pika: 25
```

2. Medium Mode: Desarrolle la función `poke heavierPoke(poke team[6])`, que recibe como parámetro un arreglo de tipo `poke` de largo seis (6) llamado `team`. La función debe retornar el `poke` con mayor atributo peso dentro de `team`. Un ejemplo de uso se muestra a continuación:

`poke_main.cpp`

```
int main(){

    poke equipo[6], pesado;
    \\ ...

    pesado = heavierPoke(equipo);

    std::cout << "El Poke más pesado es " << pesado.nombre << " con ";
    std::cout << pesado.peso << " [kg]" << std::endl;
    \\...
}
```

Salida (consola)

```
El Poke más pesado es Snor con 250 [kg]
```

3. Hard Mode: Complete la función `int main()` del archivo `poke_main.cpp` de forma que se solicite información de seis (6) **Poke**. Estos deben ser almacenados en un arreglo de tipo `poke`, además el programa debe imprimir por salida standard (`std::cout`) el nombre y el peso del **Poke** más pesado. Un ejemplo de la compilación y funcionamiento del binario generado se muestra a continuación:

```
[user@pc practico1]$ g++ -o POKE main_poke.cpp
[user@pc practico1]$ ./POKE
Ingrese nombre de Poke; Pika
Ingrese peso de Pika: 25

Ingrese nombre de Poke; Char
Ingrese peso de Char: 30

Ingrese nombre de Poke; Vena
Ingrese peso de Vena: 120

Ingrese nombre de Poke; Snor
Ingrese peso de Snor: 250

Ingrese nombre de Poke; M2
Ingrese peso de M2: 90

Ingrese nombre de Poke; M
Ingrese peso de M: 27

El Poke más pesado es Snor con 250 [kg]
```