



Estructuras de Datos y Funciones

Ejercicios

Introducción a la Programación Orientada a Objetos

Escriba los archivos de C++ para ejecutar los programas que resuelvan cada uno de los requerimientos solicitados. Escriba la solución a los puntos en un archivo `.cpp` diferente y escriba en un archivo de texto la explicación de la forma en que resolvió cada uno de los puntos del ejercicio (use el número de la línea de código para referenciar instrucciones en el documento cuando lo necesite). Comprima los archivos `.cpp` y el documento de texto en un archivo `.zip` con el nombre **EjercicioClase4-<codigo estudiante>.zip**. En el campus virtual estará un enlace habilitado para que suba el archivo hasta el **domingo 11 de marzo a las 23:55**.

1. Se requiere un programa que retorne el mayor o el menor valor de un arreglo. La función puede recibir un arreglo de flotantes o de enteros a ordenar (use sobrecarga) y el tamaño del arreglo. También recibe un booleano para indicar si se desea buscar el mayor (`true`) o el menor (`false`); por defecto la función buscará el mayor.
2. Samus Aran cuenta con el power suit, el cual tiene un analizador de daño basado en los golpes que recibe. El power suit cuenta con un programa que recibe un arreglo de enumeraciones, donde cada enumeración representa un golpe recibido. Cada golpe tiene un nivel de daño y le resta puntos de energía como se indica a continuación:
 - MORDEDURA = 50
 - PUÑETAZO = 100
 - ESPADAZO = 150
 - RAYO = 200
 - EXPLOSION = 300

Defina una función que reciba el nivel de energía inicial, la cantidad de golpes recibidos y un arreglo de enumeraciones con las secuencia de golpes que recibió el power suit, y retorne el nivel de energía restante después de recibir los golpes

3. Se requiere un programa para calcular las notas de los estudiantes de un curso basado en las siguientes consideraciones:

- Se tiene una estructura llamada **Estudiante** que tiene los siguientes miembros: nombre del estudiante, código del estudiante, nota del quiz1, nota del quiz2, nota del parcial, nota del proyecto y nota final.
- Se tiene un arreglo de estructuras **Estudiante** donde cada índice del arreglo corresponde a un estudiante. Use como ejemplo el siguiente código:

```
// Definir estructura

struct Ejemplo{
    int     miembro1;
    float   miembro2;
    int     miembroN;
};

// Arreglo de estructuras

Ejemplo arreglo [10];

// Acceso (Lectura y escritura) de componentes del arreglo

arreglo [2].miembro1 = 100;
cout << arreglo [2].miembro1;
```

Para cargar los valores del arreglo se puede usar un ciclo o se pueden cargar directamente en el archivo .cpp

- Calcule la nota final de cada estudiante teniendo que la evaluación del curso es la siguiente:
 - quizes: 25 %
 - parcial: 35 %
 - proyecto: 40 %
 - Finalmente muestre en consola la lista de todos los estudiantes con todas las notas incluyendo la nota final
4. Se requiere un programa que valide el ingreso de claves de un video juego. El usuario debe presionar cinco teclas en consola y el programa valida si la secuencia de teclas ingresada corresponde a una clave del juego o no, y debe mostrar a que clave pertenece. Para resolver este ejercicio se requiere el uso de arreglos, enumeraciones y funciones. Para resolver este problema siga las siguientes instrucciones:

- Declare una enumeración con los valores UP = 72, DOWN = 80, LEFT = 75, RIGHT = 77 (Estos valores corresponde al código de las teclas de dirección para leer del teclado)
- Para leer de teclado se usará la función `getch()` de la librería `<conio.h>`, y cada valor que lea, corresponderá a un valor entero que indica el código del carácter que leyó del teclado

```
#include <conio.h>
int character = getch();
```

- Declare un arreglo de enteros donde se van a almacenar los valores de las cinco teclas presionadas por el usuario. Muestre un mensaje para pedir que presione las cinco teclas y realice un ciclo para almacenar los valores
- Defina una función que reciba un arreglo de enteros y el puntero de una arreglo de string, y no tenga valor de retorno. Esta función es usada para transformar el arreglo de enteros en un arreglo de string usando un `switch` donde se ingrese un entero, y si los valores corresponden a los valores de la enumeración, el string valdrá “up”, “down”, “left”, “right” respectivamente, sino corresponden haga el default para que transforme el entero a carácter y el carácter a string en mayúscula como se indica a continuación:

```
int letra;
char letraChar = letra;
string letraString(1, letra);
toupper(letraString);
```

- Cree una función que reciba como argumento el arreglo de string que fue transformado y retorne un string que indica el identificador de la clave ingresada. Si no corresponde a ninguna clave el string será “clave incorrecta”. Las claves se encuentran almacenadas en un arreglo de dos dimensiones llamado `claves` y el identificador de cada clave estará en el primer índice de cada arreglo. Los valores de claves a evaluar y la función de evaluación se presentan a continuación:

```
string claves[6][6] =
{{"vida", "up", "down", "left", "right", "A"},
{"armas", "left", "right", "A", "X", "Y"},
{"municion", "down", "up", "right", "left", "B"},
{"revivir", "up", "down", "X", "Y", "B"},
{"mil Vidas", "up", "down", "left", "right", "R"},
{"carros", "A", "B", "X", "Y", "R"} };

string verificarClave(string* caracteresString){
```

```

string clave = "Clave no existe";
for(int i = 0; i < 6 ; i++){
    bool coinciden = true;
    for(int j = 0; j < 5 ; j++){
        bool eval = /// Defina a que debe coincidir eval
        if(eval){
            coinciden = false;
            break;
        }
    }

    if(coinciden){
        clave = /// Defina valor que debe tomar clave
        break;
    }
}
return clave;
}

```

- Finalmente muestre en consola el identificador de la clave