







TALLER 1 PROBLEMA 3

Métodos Computacionales 2 - 2023-I

Para el ejercicio pruebe que corra en el Binder del curso.

Realice un programa en C++ que simule un combate uno contra uno. Usted debe crear una clase `Warrior`, la cual debe tener dos parámetros `health` (igual a 50 puntos) y `attack` (igual a 5 puntos) y un método `is_alive()`, el cual retorna `true` si `health > 0` o `false` en caso contrario. Adicionalmente debe crear la clase `Knight`, la cual tiene los mismos atributos y el mismo método que la clase `Warrior` pero tiene un `attack = 7`.

Finalmente debe crear la función `fight()`, la cual define el duelo entre dos luchadores. El duelo ocurre de acuerdo al siguiente principio: Cada turno, el primer guerrero golpea al segundo y el segundo pierde una cantidad de `health` igual al ataque del primer guerrero. Después de eso si aún sigue vivo, el segundo guerrero hace lo mismo al primer guerrero. La lucha termina cuando uno de los dos este muerto. Si el primer guerrero aún está vivo después de la pelea la función `fight()` debe retornar `true`, retorna `false` si el que queda vivo es el segundo.

Warrior		1		Knight		2		Knight									
♥ 50				♥ 50		♥ 50		♥ 50 - 5 = 45									
♣ 5				♣ 7		♣ 5		♣ 5									
																	
Warrior		3		Knight		4		Knight									
♥ 50 - 7 = 43				♥ 45		♥ 43		♥ 45 - 5 = 40									
♣ 5				♣ 7		♣ 5		♣ 7									
																	
Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Warrior	♥	50	50	43	43	36	36	29	29	22	22	15	15	8	8	1	-6
Knight	♥	50	45	45	40	40	35	35	30	30	25	25	20	20	15	15	10

♥ Health ♣ Attack

El programa debe incluir los siguientes archivos:

- `main.cpp`: este archivo debe incluir la función `main` tal cual como se encuentra abajo, y la función `bool fight()`.
- `warrior.h`: el header `warrior` debe tener la clase `Warrior` y declarar el método `is_alive()`, y en caso de ser necesario el constructor.

- `warrior.cpp`: este archivo debe implementar las funciones declaradas en el archivo `warrior.h`.
- `knight.h` igual que `warrior.h`, pero con la clase `Knight`
- `knight.cpp` igual que `warrior.cpp`.
- `Makefile` el archivo `makefile` debe poder compilar los tres archivos `main.cpp`, `warrior.cpp` y `knight.cpp` con solo aplicar el comando `make all` en la terminal debe quedar el ejecutable `output`. Además agregue en el archivo `MakeFile` el comando `clean`: para borrar archivos de `output`.

Todos los archivos deben estar guardados en una carpeta llamada `ApellidoNombreApellidoNombre_Taller1_ParteB`, y se debe comprimir la carpeta en un archivo llamado `ApellidoNombreApellidoNombre_Taller1_ParteB.zip`.

La calificación de su ejercicio se realizara con la siguiente serie de comandos en `binder`.

```
unzip ApellidoNombreApellidoNombre_Taller1_ParteB.zip
cd ApellidoNombreApellidoNombre_Taller1_ParteB
make all
.\output
make clean
```

Copie la siguiente función `main()` en su archivo `main.cpp`, y entregue la función `main` tal como aparece,

```
int main ()
{
    cout << "Test 1:" << endl;
    Warrior husband;
    Knight wife;
    cout << fight(husband, wife) << endl;
    cout << husband.is_alive() << endl;
    cout << wife.is_alive() << endl;
    cout << "\n";
    cout << "Test 2:" << endl;
    Warrior chuck;
    Warrior bruce;
    Knight carl;
    Warrior dave;
    Warrior mark;
    cout << fight(chuck, bruce) << endl;
    cout << fight(dave, carl) << endl;
    cout << chuck.is_alive() << endl;
    cout << bruce.is_alive() << endl;
    cout << carl.is_alive() << endl;
    cout << dave.is_alive() << endl;
```

```
cout << fight(carl, mark) << endl;
cout << carl.is_alive() << endl;
return 0;
}
```

El resultado esperado de poner en la terminal `make all` seguido de `.\output` debe ser:

```
Test 1:
0
0
1
```

```
Test 2:
1
0
1
0
1
0
0
0
```

Nota: El programa debe ser llamado `ApellidoNombreApellidoNombre_Taller1_ParteB.zip` donde `Apellido` y `Nombre` debe reemplazarlos con el primer apellido y nombre de la pareja. El programa debe correr en binder, si el programa demora más de 10 segundos en correr se considera incorrecto.