

CS2013 - Programación III
Simulacro de Práctica Calificada #3 (PC3)
2023 - 2

Profesor: Rubén Rivas

Hash - 7 puntos

Diseñar y desarrollar el template de función **can_text_split** que confirme si es posible dividir un texto en palabras contenidas en una lista (contenedor generico).

Caso de uso #1

```
vector<string> vec = {"este", "texto", "UTEC", "prueba"};  
string text1 = "estetexto";  
cout << boolalpha << can_text_split(text1, vec) << endl;
```

Caso de uso #2

```
list<string> lst = {"este", "texto", "UTEC", "prueba"};  
string text2 = "estetextoUTEC";  
cout << boolalpha << can_text_split(text2, lst) << endl;
```

Caso de uso #3

```
cout << boolalpha  
    << can_text_split<string>("prueba",  
        {"este", "texto", "UTEC", "prueba"}) << endl;
```

Caso de uso #4

```
cout << boolalpha  
    << can_text_split<wstring>(L"estaprueba",  
        {L"esta", L"texto", L"UTEC", L"prueba"}) << endl;
```

Heap - 4 puntos

Desarrollar un functor llamada **nearest_spheres** que utilizando un heap, reciba una colección de esferas (**sphere**) cuyo radio (expresado **ms**) es **r** y peso específico (**kg/m³**) es **sw** y devolver la primeras **n** esferas cuyo peso es el más cercano al peso promedio de todas las esferas.

```
// Crear functor
nearest_spheres<int, double> ns;
// Agregar esferas
ns.add_sphere(10, 2); // #1
ns.add_sphere(5, 1);  // #2
ns.add_sphere(8, 2);  // #3
ns.add_sphere(20, 1); // #4
ns.add_sphere(15, 2); // #5
// Obteniendo esferas en orden: #1, #3, #5
std::vector<sphere<int, double>> spheres = ns(3);
// Mostrando resultado
std::cout << ns.average_weight() << std::endl;
std::cout << spheres;
```

```
// Crear functor
nearest_spheres<double, double> ns;
// Agregar esferas
ns.add_sphere(15.5, 2.5); // #1
ns.add_sphere(5.3, 1.3);  // #2
ns.add_sphere(8.7, 2.3);  // #3
ns.add_sphere(20.8, 1.7); // #4
ns.add_sphere(15.1, 2.8); // #5
ns.add_sphere(12.4, 2.8); // #6
ns.add_sphere(24.1, 2.6); // #7
ns.add_sphere(11.1, 2.4); // #8
ns.add_sphere(2.1, 2.4);  // #9
ns.add_sphere(4.1, 2.4);  // #10
// Obteniendo esferas en orden
std::vector<sphere<double, double>> spheres = ns(5);
// Mostrando resultado
std::cout << ns.average_weight() << std::endl;
cout << spheres;
```