

Control 4

Llenar la mochila

El objetivo de este control es afianzar las habilidades necesarias para el desarrollo sistemático de algoritmos iterativos con cierto grado de sofisticación.

El problema

Se dispone de una mochila que tiene un aguante de k Kg. Asimismo, se dispone de una hilera de productos con los que se puede llenar la mochila. Las únicas restricciones para llenar la mochila son que (i) únicamente pueden ponerse en la mochila productos que aparecen en posiciones consecutivas en la hilera; y (ii) el peso total de los productos no puede exceder el aguante k de la mochila.

Debes especificar y desarrollar sistemáticamente un algoritmo iterativo eficiente que, dado un vector de enteros $\text{int } a[n]$, en el que cada posición representa un producto en la hilera y cada valor representa el peso de dicho producto, y un aguante k para la mochila, encuentre el número máximo de productos con los que puede llenarse la mochila. Puede asumirse que (i) al menos hay un producto; (ii) tanto los pesos de los productos como el aguante de la mochila son positivos; y (iii) ningún peso de producto excede el aguante de la mochila.

Por ejemplo, el siguiente vector a :

10	5	1	5	1	1	15	2	3	4
----	---	---	---	---	---	----	---	---	---

representa una hilera de 10 productos. El primer producto pesa 10 kg, el segundo 5 Kg, el tercero 1 Kg ... y así sucesivamente. Si el aguante total de la mochila es $k = 20$ kg, bajo la restricción de que únicamente pueden elegirse productos consecutivos en la hilera, como máximo se podrá guardar en la mochila 5 productos, los correspondientes a las casillas sombreadas.

Trabajo a realizar

Para realizar el control se proporciona un archivo `control4.cpp` que contiene un programa que lee por la entrada estándar vectores de enteros, así como el aguante de la mochila, invoca a una función `llena_mochila` sobre dichos datos, que calcula el número máximo de productos consecutivos que puede almacenar la mochila, e imprime el valor devuelto.

A continuación, se muestra un ejemplo de entrada procesable por este programa, y de salida producida (suponiendo una implementación adecuada de `llena_mochila`). Cada entrada se muestra en dos líneas: la primera indica el vector de pesos de productos, la segunda el aguante de la mochila. Los vectores se introducen indicando, como primer valor, el número de elementos, y, a continuación, enumerando los distintos elementos. El final de los casos de prueba se indica con una línea que contiene únicamente -1.

Entrada	Salida
10 10 5 1 5 1 1 15 2 3 4	5
20	3
5 10 20 1 10 9	
20	
-1	

Tu trabajo consiste en:

- Especificar y diseñar el algoritmo, rellenando los comentarios incluidos en el archivo.
- Implementar este algoritmo.
- Entregar `control4.cpp` a través del juez en línea de la asignatura.

Importante:

- Únicamente se evaluarán aquellas entregas que superen satisfactoriamente los casos de prueba del juez, sin superar el tiempo máximo previsto para ello.
- No modificar el código proporcionado. Únicamente deben responderse a los distintos apartados, en el interior de los comentarios, e implementar la función `llena_mochila` (sí pueden incluirse funciones auxiliares, si se considera necesario).