Algoritmos voraces

Objetivos

- Diseñar un algoritmo que siga una estrategia voraz para resolver un problema.
- Implementar el algoritmo desarrollado utilizando C++.
- Implementar la lectura de datos desde un fichero.
- Trabajar en grupo.

Actividades (Equipos de 2 o 3 alumnos)

Actividad 1:

Se debe resolver el problema de la Mochila fraccionaria. En este problema tenemos un total de `n` objetos cada uno de ellos con un peso y un valor y una mochila (o recipiente) que es capaz de soportar un peso máximo `M`. Teniendo en cuenta que los objetos se pueden fraccionar, se quiere encontrar la mejor selección de objetos a introducir en la mochila de forma que el valor que lleve sea máximo y no se sobrepase el peso máximo `M`.

Una implementación de este algoritmo es:

```
función Mochila(M:real+,O: &elemento[1..n],P:&real+[1..n],V:& real+[1..n]): real[1..n]

      X:[0,1][1..n]
      peso:real+
      i:entero+

      parai←1 hasta n hacer
            Xi← 0
      fpara

      OrdenarDecreciente_V/P(O,V,P)

      i← 1
      peso ← 0

      mientras peso < M  y  i≤ n hacer
            si peso + Pi ≤ M
                  Xi← 1
                  peso ← peso + Pi
            si no
                  Xi ← (M-peso)/Pi
                  peso ← M
            fsi
            i ← i + 1
      fmientras

      devolver X

ffunción
```

Implementa un algoritmo según el pseudocódigo que se proporciona. Crea un programa que pida por teclado el peso máximo de la mochila y el número total de elementos (para cada uno de ellos se deberá solicitar su peso y valor). Una vez introducidos los valores se resolverá el problema invocando a la función anterior y se mostrará por pantalla la solución obtenida (objetos incluidos y valor de la mochila).

Actividad 2:

Utilizar la información del Anexo para sustituir la parte del programa de la actividad 1 que solicita los elementos por teclado por una función que lea la entrada desde la información almacenada en un fichero de texto. Definir al menos 3 ficheros de entrada diferentes variando los valores de `n` y `M`.

A continuación, se detalla un ejemplo de fichero de entrada, que deberá seguir el formato descrito en los comentarios que se ofrecen junto a cada línea:

entrada.txt

```
15 // M
3 // n
8 6 3 // P (pesos de los elementos)
40 24 18 // V (valores de los elementos)
```

El programa deberá pedir al usuario el nombre del fichero que se desea leer. Se puede asumir que todos los ficheros de entrada se encontrarán en la misma carpeta que el ejecutable.

Anexo: Lectura de un fichero

Para realizar la lectura de un fichero en C++ se pueden utilizar diferentes funciones. En esta práctica se propone el uso de las funciones open(), getline(), strtok()… como se muestra en el siguiente ejemplo donde se implementa la lectura de un vector de enteros de tamaño n (primera línea el tamaño del vector y la segunda sus valores separados por espacios, tabuladores o retornos de carro).

```
#include<iostream>
#include<fstream>
#include<stdlib.h>
#include<string.h>

using namespace std;

int main() {
	ifstream fDatos;
	char linea[200], *tok;
	int n, i;
	int *vector;

	// Abrir fichero de nombre "entrada.txt"
	fDatos.open("entrada.txt");

	//Lectura de la primera linea
	fDatos.getline(linea, 200);
	tok=strtok(linea, " \t\r\n");
	n = atoi(tok); // convertir el token a entero
	vector = new int[n+1];

	//Lectura de la segunda linea
	fDatos.getline(linea, 200);
	tok=strtok(linea, " \t\r\n");
	i=1;
	vector[i] = atoi(tok);
	for(i=2 ; i<=n ; i++)
	{
		tok=strtok(NULL, " \t\r\n");
		vector[i] = atoi(tok);
	}

	cout << "n = " << n << "\n";
	cout << "vector -> [ " << vector[1];
	for(i=2 ; i<=n ; i++)
	cout << " , " << vector[i];
	cout << " ]\n\n";

	fDatos.close();
	cout<< "FIN...\n\n";
	return 0;
}
```

Modo de entrega

La práctica se realizará en equipos de **dos o tres alumnos** y se entregarán los siguientes ficheros con los nombres que se indican.

Archivo comprimido:	practica6.zip
Contenido del archivo:	p6_a1.cpp, p6_a2.cpp, a2_entrada1.txt, a2_entrada2.txt, a2_entrada3.txt Cada fichero contiene el código fuente de las actividades correspondientes y los nombres de los miembros del equipo.

Todos los componentes del equipo entregarán el archivo practica6.zip en la tarea llamada “Práctica 6: Algoritmos voraces”, dentro del campus virtual.

Fecha fin de entrega: Lunes, 19 de abril de 2021 a las 15:00.

Evaluación

La calificación total de la actividad es 0,2 puntos (0,1 puntos cada actividad).

A continuación, se indica el sistema de evaluación:

- **Opción A: La práctica se entrega durante la sesión de prácticas.**

Cada alumno/-a del equipo entregará la práctica (practica6.zip) en la tarea de la web de la asignatura. Antes de subir la tarea a la web se debe recibir el visto bueno del profesorado y todos los miembros del equipo deben estar presentes y explicar cualquier aspecto que se solicite. Se evaluará cogiendo al azar la práctica de uno de los miembros del equipo, de forma que dicha práctica será la que se corrija.

Si hay algún miembro del equipo que no entrega la práctica en la tarea, no responde correctamente a las preguntas realizadas por el profesorado o no está presente…, no se le calificará según esta opción y puede optar a la calificación según la opción B.

- **Opción B: La práctica se entrega en horario posterior a la sesión de prácticas.**

A esta opción optarán los equipos y miembros de equipos que no cumplen los requisitos para ser evaluados según la opción A. Cada miembro del equipo debe entregar tanto la práctica en la tarea de la web de la asignatura como el programa que funcione correctamente. Además, se deberá grabar un vídeo donde participen todos los integrantes del equipo (duración 3-5 minutos) donde se explique cómo se ha realizado la actividad y se ejecute el programa. Este vídeo deberá subirse también a la tarea (tamaño máximo 20MB).

Se calificará cogiendo al azar la práctica de uno de los miembros del equipo, de forma que dicha práctica será la que se corrija. La calificación máxima que se puede obtener es un 0,1 (0,05 cada actividad). Los programas deben seguir las especificaciones que se dan.

- **Opción C: 0 puntos**

- El alumno/a:
  - No entrega la práctica en la tarea de la web de la asignatura.
  - No asiste a la sesión de prácticas cumpliendo con las condiciones establecidas.
- El programa no funciona correctamente y no realiza lo que se pide.
- Se entrega bajo la opción B y no se adjunta vídeo.
- Se detecta copia con otras prácticas. La nota será un 0 en esta práctica para todas las prácticas implicadas, aun cuando la práctica haya sido valorada previamente de forma positiva por parte del profesorado.