UNIVERSIDAD AUTONOMA DE NUEVO LEON.

FACULTAD DE INGENIERIA MECÁNICA Y ELECTRICA.

TAREA 4: ALGORITMO HEURISTICO PARA PROBLEMA DE MOCHILA

PRESENTADO POR:

JESUS JAVIER MORENO VAZQUEZ 1619830

HORA: V4-V6

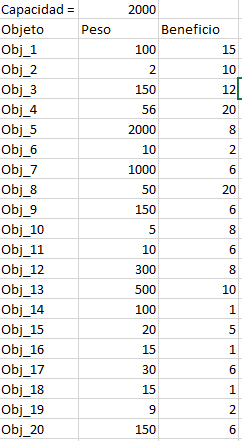
PROFESOR: DR. MARIA ANGELICA SALAZAR

SAN NICOLAS DE LOS GARZA, NUEVO LEON, A 5 DE FEBRERO DEL 2016

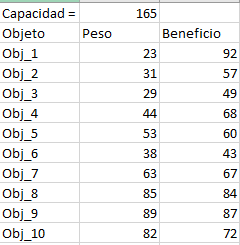
**DESCRIPCION GENERAL DEL PROBLEMA**

Se tiene una lista de ‘n’ objetos, cada uno tiene un peso y un beneficio respectivamente, también se tiene una “mochila” la cual tiene una capacidad ‘K\_max’. El objetivo del problema es introducir tantos objetos a la mochila buscando aumentar al máximo el beneficio sin exceder la capacidad. Se presentan 9 casos para testear el algoritmo desarrollado:

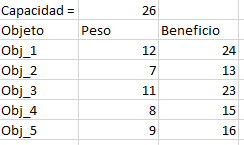
**CASO CLASE**: Propuesto en Clase



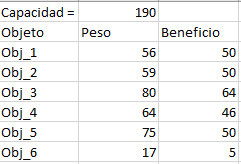
**CASO 1:** 10 Objetos con una capacidad máxima de 165



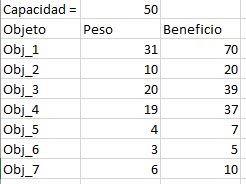
**CASO 2:** 5 objetos y una capacidad 26



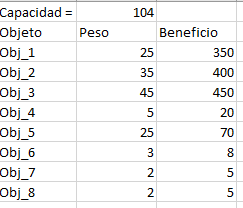
**CASO 3:** 6 objetos y una capacidad de 190

****

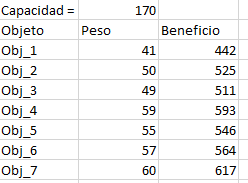
**CASO 4:** 7 objetos y una capacidad de 50



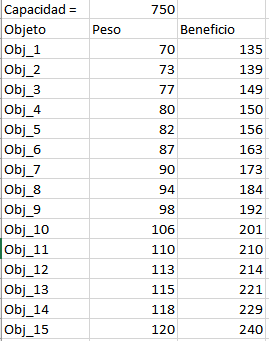
**CASO 5:** 8 objetos y una capacidad de 104



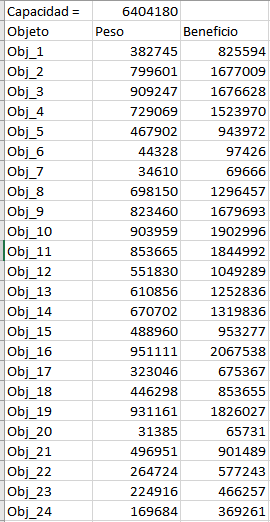
**CASO 6:** 7 Objetos con una capacidad de 170



**CASO 7:** 15 Objetos con una capacidad de 750



**CASO 8:** 24 Objetos y una capacidad de 6404180



**ALGORITMO PROPUESTO:**

1.- Lectura de Datos

2.- Sacar el promedio de los pesos.

3.- Ordenar objetos en una nueva lista respecto al beneficio de manera descendiente sin tomar en cuenta aquellos objetos cuyos pesos excedan el promedio.

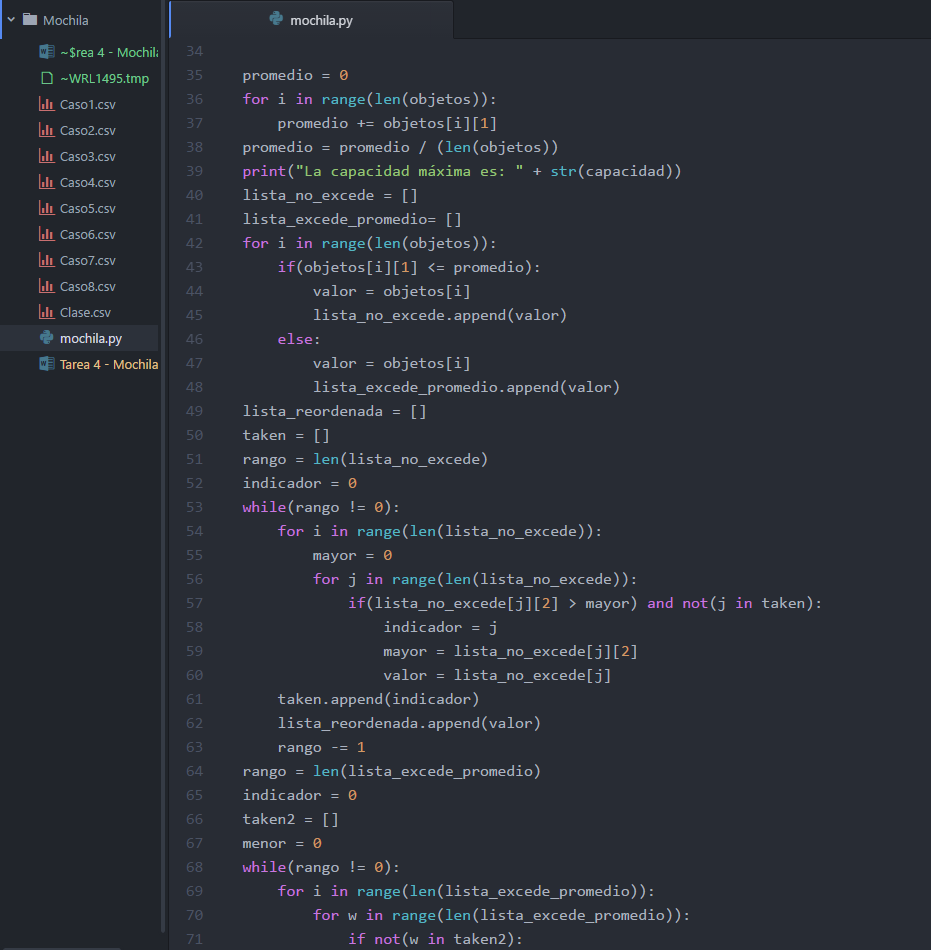
4.- Agregar los objetos restantes (aquellos que exceden el promedio) pero ahora respecto al peso de manera ascendente.

5.- Seleccionar el primer elemento que no haya sido seleccionado.

6.- Asignar dicho elemento si y solo si no excede la capacidad

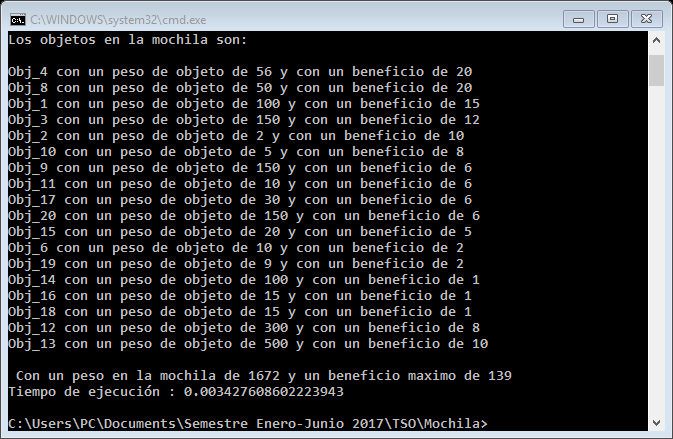
7.- Si aún hay espacio, sigue introduciendo objetos (paso 5), si no, termina la ejecución.

8.- Reportar resultados.

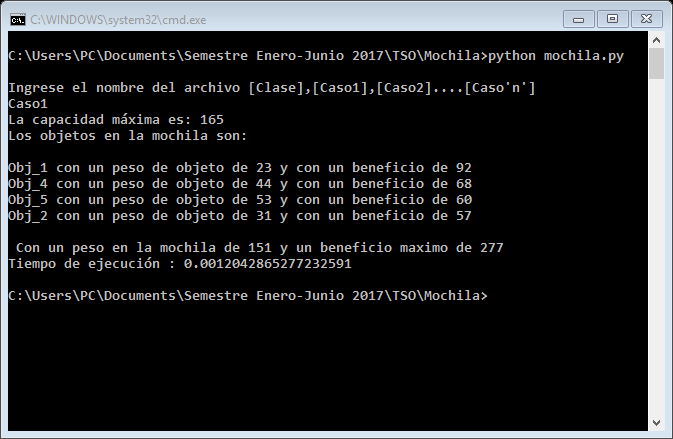
****

**RESULTADOS**

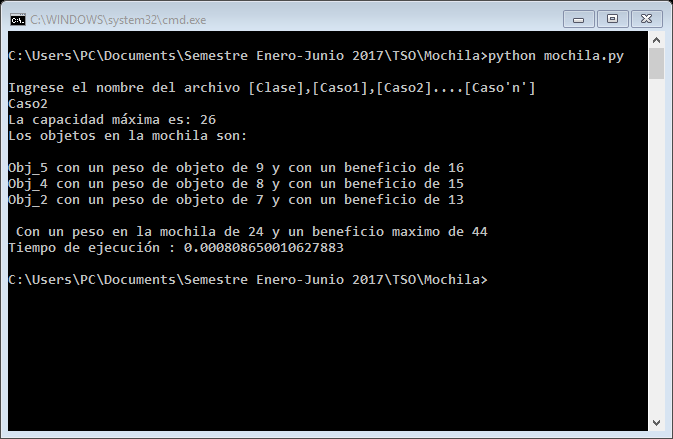
Clase:



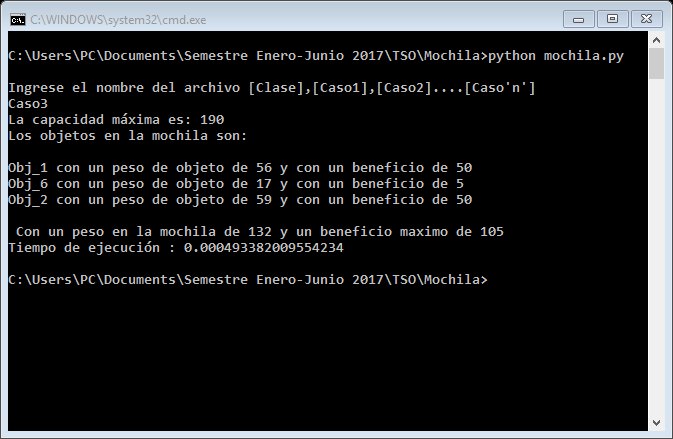
Caso1:



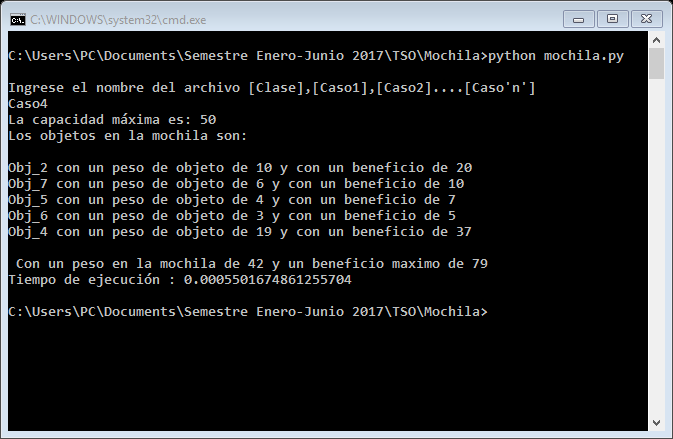
Caso2:



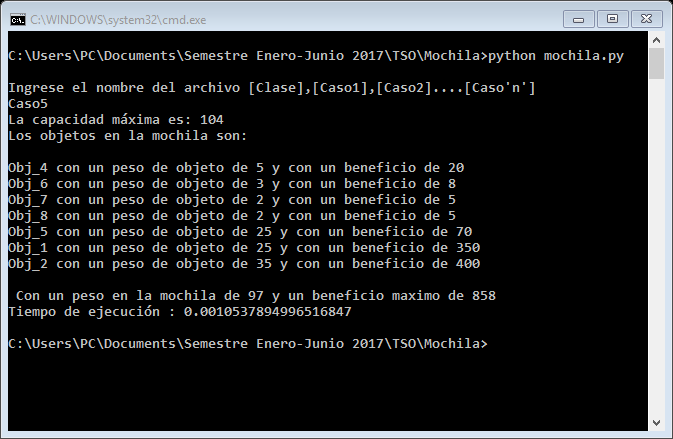
Caso3:



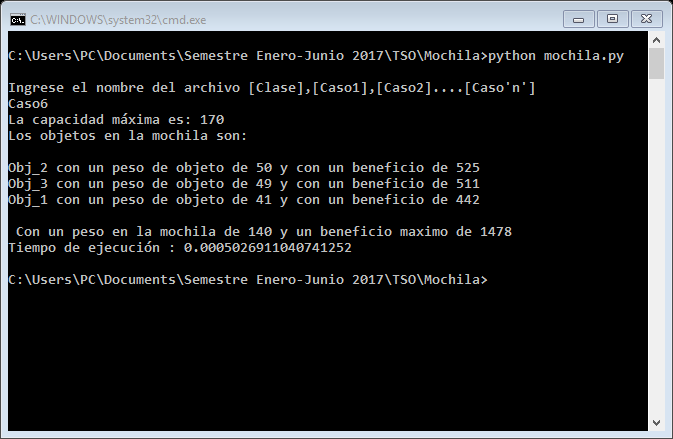
Caso4:



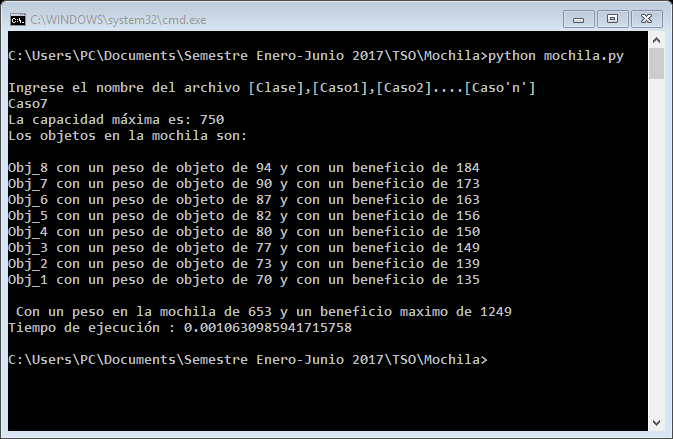
Caso5:



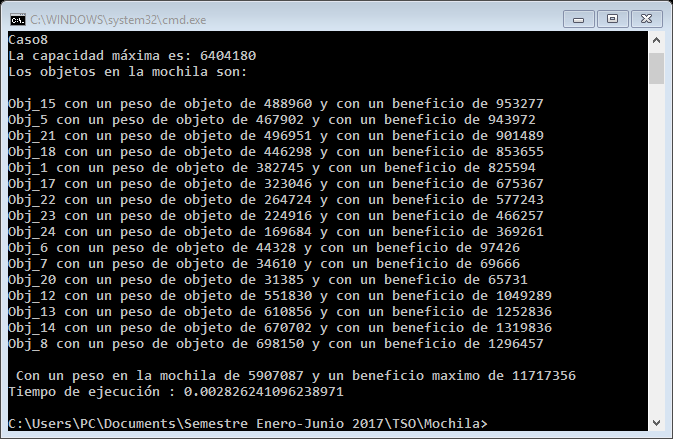
Caso6:



Caso7:



Caso8:



**CARACTERISTICAS DEL SISTEMA**

Procesador: Intel® Core i5-4590 CPU @ 3.30 GHz

Memoria Instalada (RAM): 8.00 GB (7.88 GB utilizable)

Sistema Operativo: Windows 10 Pro

Tipo de sistema: Sistema operativo de 64 bits, procesador x64

**CONCLUSIONES**

Este algoritmo fue mucho más sencillo de codificar que los realizados anteriormente, no se presentó ninguna dificultad, sin embargo, recomiendo realizar estas tareas con tiempo y revisar siempre los resultados, ya que en este ejemplo tuve un error de lógica en el que no seleccionaba bien los objetos que se introducían a la mochila, error que solo ocurría en uno de los 9 casos revisados y que no hubiese notado sin revisar todos los resultados.

**BIBLIOGRAFIA**

* https://www.python.org/doc/