

Examen I

1. *(6,5 puntos)* Los estudiantes del Laboratorio de Algoritmos II el día de San Valentín quieren descifrar el mensaje secreto de un amigo/a especial. A medida que se lee el mensaje, cada letra debe ser insertada en la cola de prioridades. En caso de conseguir un asterisco debe desencolar el elemento con menor prioridad (la prioridad viene dada por el orden lexicográfico donde $a < b$ y así sucesivamente). El mensaje se formará con los elementos extraídos de la cola de prioridades.

Por ejemplo, si el mensaje es el siguiente:

*USVZ**RE*T**LE**

El mensaje resultante será:

SUERTE

Para resolver el problema **deberá hacer uso de una cola de prioridad implementada con heaps**.

Un ejemplo de los archivos de entrada y salida sería:

Entrada:

Archivo de entrada (no poseerá caracteres especiales, ñ o acentos, ningún espacio y todas las letras estarán en mayúscula):

VT*ZQ*M

Salida:

La salida correspondiente será:

TQ

La solución del problema deberá encontrarse en un archivo llamado mensaje.py.

Nota: Deberá usar como representación de arreglos el módulo arrayT.py

2. (8,5 puntos) Los estudiantes del Laboratorio de Algoritmos II le quieren hacer una torta sorpresa a la Prof. Catherine para celebrar su cumpleaños. Para ello deberán seguir una receta en la cual se indican los ingredientes que se necesitan y su cantidad. Como no todos los ingredientes son fáciles de conseguir, también se indicará si es posible conseguirlos o no. El archivo de entrada tendrá el siguiente formato:

```

n
Ingrediente_1 [tab] Cantidad de Ingrediente [tab] Si existe/No existe
Ingrediente_2 [tab] Cantidad de Ingrediente [tab] Si existe/No existe
...
Ingrediente_n [tab] Cantidad de Ingrediente [tab] Si existe/No existe
  
```

Como los estudiantes deben hacer la torta rápidamente para poder hacer la sorpresa, deberán ordenar los ingredientes de la siguiente manera: primero los ingredientes que si se poseen, luego los que no. Dentro de este orden deberán ordenar de manera descendente por cantidad, y si dos ingredientes tienen la misma cantidad, entonces se ordenarán alfabéticamente según el nombre de los ingredientes.

El arreglo que se le pasará como parámetro al algoritmo de ordenamiento será un arreglo de tripletas donde la primera posición será el ingrediente, la segunda la cantidad de ingredientes y la tercera si posee o no el ingrediente.

En caso de que haya más de la mitad de los ingredientes que no se puedan encontrar, deberá aparecer que no es posible preparar la torta. En caso contrario deberán imprimir que sí es posible preparar la torta.

El archivo de salida poseerá el siguiente formato:

```

Ingrediente_1 [tab] Cantidad de Ingrediente [tab] Si existe/No existe
Ingrediente_2 [tab] Cantidad de Ingrediente [tab] Si existe/No existe
...
Ingrediente_n [tab] Cantidad de Ingrediente [tab] Si existe/No existe
La torta se puede preparar/La torta no se puede preparar

```

Un ejemplo de dicho ejercicio sería el siguiente:

Entrada:

Archivo de entrada (no poseerá caracteres especiales, ñ o acentos):

```

5
Mantequilla 1 Si
Harina 0 Si
Azucar 0 No
Chocolate 4 Si
Vainilla 4 Si

```

Salida:

La salida correspondiente será:

```

Chocolate 4 Si
Vainilla 4 Si
Mantequilla 1 Si
Harina 0 Si
Azucar 0 No
La torta se puede preparar

```

La solución del problema 2 deberá encontrarse en un archivo llamado torta.py.

Nota: Deberá usar como representación de arreglos el módulo arrayT.py.

Condiciones de la entrega

Cree un archivo comprimido del tipo “tgz” llamado ExamI-X.tgz, donde X es su número de carné. Su contenido serán los archivos que utilizó para la realización del examen. Debe entregar el archivo a su respectivo profesor el día 16 de febrero de 2017 antes de las 4:30 p.m.