



## Tarea 1

### IIC2133 - Estructuras de datos y algoritmos

Segundo semestre, 2015

**Entrega:** Lunes 7 de Septiembre

## Objetivos

- Utilizar los conceptos básicos de C para resolución de un problema complejo.
- Comprender las ventajas y desventajas de los distintos tipos de Tablas de Hash.

## Problema: Pizza Planeta

La empresa global Pizza Planeta ha solicitado ayuda con el siguiente problema:

Ad portas de inaugurar su nueva sucursal en el centro del universo, y en vista de la cantidad de variedades de pizzas posibles ( $\infty$ ) y el flujo de clientes, se han encontrado con que necesitan un sistema computacional capaz de soportar las siguientes operaciones:

- Cliente de nombre X ordena N pizzas de tipo Y
- Cocinero avisa que la pizza de tipo Y está lista, debe entregarse a quien la pidió, en orden FIFO<sup>1</sup>

Para esto, la empresa quiere hacer cotizaciones de los distintos sistemas que podrían soportar esto, en particular los siguientes:

- Tabla de Hash con direccionamiento cerrado usando encadenamiento (Closed addressing with chaining)
- Tabla de Hash con direccionamiento abierto usando sondeo lineal. (Open addressing with linear probing)
- Table de Hash con direccionamiento abierto usando doble hashing. (Open addressing with double hashing)

Las operaciones se harán mediante los siguientes comandos escritos en consola:

- ASK X N Y : El cliente de nombre X pide N pizzas de tipo Y
- RDY Y : El cocinero avisa que una pizza de tipo Y está lista.
- END : Cambio de turno de la tienda. Fin del test.

Cuando se haga una orden deberás registrarla, y cuando el cocinero avise que una pizza está lista deberás llamar al cliente que la solicitó, imprimiendo su nombre en consola. No importa si el cliente está esperando más de una pizza, se llamará por cada una de sus pizzas que esté lista.

Deberás incluir un informe autocontenido detallado mostrando los resultados de tus pruebas, justificando seriamente cada uno de ellos en base a la complejidad de las operaciones de cada tabla. Si algo no calza, propón una razón que lo explique. Presenta la información de manera amigable.

---

<sup>1</sup>Dentro de los que pidieron ese tipo de pizza. Si alguien está desde antes, esperando otro tipo, será llamado cuando esté listo el tipo el pidió

Ten en mente que dado que las pizzas pueden ser arbitrariamente grandes, pero no superarán los 2048 caracteres.

En el sitio web del curso encontrarás un archivo `.h` que contiene las funciones de hash que deberás usar en tu programa. Esto es para que todas las tareas trabajen en las mismas condiciones.

## Bonus: +10 % de la nota final

Escribe la definición formal de una Pizza. Extiende esta definición al conjunto de todas las Pizzas posibles. Demuestra que la cardinalidad de este conjunto es  $\infty$ . ¿Es contable? Justifica.

## Input

Tu programa se probará en Linux (Ubuntu 14.04 32-bit con 4GB de RAM) cargando los archivos a la consola mediante el siguiente comando:

```
$ ./tuPrograma modo < test.txt
```

`tuPrograma` es tu programa compilado

`modo` elige el modo de tabla de hash a usar:

- 0 = Tabla de Hash con direccionamiento cerrado usando encadenamiento.
- 1 = Tabla de Hash con direccionamiento abierto usando sondeo lineal.
- 2 = Tabla de Hash con direccionamiento abierto usando doble hashing.

`<` indica que el archivo se leerá como si hubiera sido copiado y luego pegado a la consola.

`test.txt` es el archivo que contiene una sucesión de comandos para probar.

Notese que el uso del pipe `<` es solo para automatizar el testing. De todas maneras tu programa debiera poder funcionar leyendo los comandos desde la consola.

## Output

Tu programa deberá imprimir en consola el orden en el que se llama a los clientes según cuando estan listas sus pizzas. Por ejemplo, si tu programa recibe la siguiente sucesión de comandos

```
ASK Juan 1 Pepperonni
ASK Rob 1 Napolitana
ASK María 2 Pepperonni
RDY Pepperonni
ASK Tulio 1 Pepperonni
RDY Pepperonni
RDY Pepperonni
RDY Pepperonni
RDY Napolitana
END
```

Deberá imprimir esto:

```
Juan
María
María
Tulio
```

Rob

No imprimas más cosas, ya que estorban a la hora de corregir. Puedes imprimir en consola para probar tu programa, pero la versión que entregues sólo debe imprimir el resultado. Si imprimes otras cosas se descontará de tu nota.

## Evaluación

El 70 % de la nota dependerá de una evaluación experimental de tu programa y considerará eficiencia, relativa a los resultados del resto del curso. El 30 % restante se obtendrá desde el informe. Si el código C que entregues no funciona correctamente la nota máxima a obtener será un 3.5

Si tu programa demora más de 5 minutos será cortado y se considerará como que fallaste ese test. Los programas se probarán con varios test, de dificultad creciente.

Se asignará medio punto a la nota del código correspondiente al uso de memoria, el cual se calculará de la siguiente manera: Sea  $x$  la cantidad de bytes solicitados, e  $y$  la cantidad de bytes devueltos, este punto equivale a  $P = 1 - \frac{x-y}{x}$ , con lo que se puede llegar a una nota de 7.5

## Entrega

Sigue al pie de la letra estas instrucciones. Ellas tienen por objetivo facilitar la labor del ayudante.

- Entrega tu informe en formato PDF en el buzón correspondiente habilitado en el sitio web del curso.
- Entrega el código de tu programa en el buzón correspondiente habilitado en el sitio web del curso.
- El archivo entregado para el código debe ser de extensión `.zip` y debe generar un directorio con tu número de alumno. Tu programa deberá compilar al hacer los siguientes comandos:

```
$ cd tuNumeroDeAlumno
tuNumeroDeAlumno$ make
```

ó en su defecto

```
$ cd tuNumeroDeAlumno
tuNumeroDeAlumno$ gcc tarea1.c -std=c99 -o tarea1 -lm
```

en el segundo caso tu `main()` debe estar en un archivo llamado `tarea1.c`, y en ambos casos el resultado de la compilación deberá ser un ejecutable llamado `tarea1`

Se recomienda probar esto en la Máquina Virtual y en caso de que tu tarea necesite parámetros adicionales para compilar correctamente, tener un Makefile que se encargue de eso.

Se descontará puntaje de la nota final de la tarea si las reglas de entrega no son cumplidas.