

## Trabajo Práctico 1 Organización del Procesador

### Cód 1949 - Año 2010

#### Transformación y ordenamiento

Uno de los procedimientos más comunes y útiles en el procesamiento de datos es la clasificación u ordenamiento de los mismos. En esta oportunidad se nos presenta un problema a solucionar.

Tenemos una lista de números enteros positivos desordenada. Estos números están en diferente formato: binario, octal y decimal. Cada número en la lista es identificable con dos dígitos iniciales que representan en qué codificación están escritos, y además tienen un delimitador como marca de fin del números (guión medio). Los números tienen una extensión de cadena total de 7 caracteres: dos para la codificación, 4 para el número en sí mismo, 1 para el delimitador de fin de número. Los dígitos iniciales pueden ser:

00	Binario
01	Octal
10	Decimal

Veamos un ejemplo de la lista:

000101-000111-001010-010077-001001-100031-000010-101234-001111-010000-010100-#

Además la cadena posee una marca final para identificar el fin del proceso en el recorrido por los números. En el ejemplo, la cadena traducida a decimal sería la siguiente:

5	7	10	63	9	31	2	1234	15	0	64
---	---	----	----	---	----	---	------	----	---	----

Dado el módulo C (main.c) publicado en Moodle, **implemente** en lenguaje NASM las siguientes funciones:

```
void decode(short int *input, unsigned int i_size, unsigned short int* numbers, unsigned int *n_size)
```

```
void bubble_sort(unsigned short int * numbers, unsigned int p_size)
```

1) Decode: Construye un arreglo (*numbers*) con los datos dados en *input*, decodificando cada número en su representación decimal.

2) Bubble Sort: Dado un arreglo de números desordenado (*numbers*) ordena el arreglo utilizando el algoritmo de ordenamiento bubble sort.

Luego de ejecutar su programa la salida sería:

0	2	5	7	9	10	15	31	63	64	1234
---	---	---	---	---	----	----	----	----	----	------

Especificaciones:

- Punto 1) Para la rutina *decode* deberá definir una subrutina que dado un número codificado retorne el decimal correspondiente a su codificación. Luego utilizar dicha subrutina para transformar el arreglo codificado en su representación decimal. Deberá imprimir por pantalla el

arreglo convertido a números decimales.

- Punto 2) Para la rutina *bubble\_sort* deberá definir una subrutina que dado un arreglo, ordene el mismo usando el algoritmo de ordenamiento bubble sort y muestre por pantalla el arreglo ordenado.

Grupos de trabajo:

- Los grupos deberán ser de tres (3) alumnos.

Fechas de entrega:

- 29/10 :
  - Entrega del punto 1.
  - Modo de corrección: se ejecutará el archivo ejecutable, sin revisión de código fuente.
- 09/11
  - Entrega del proyecto completa.
  - Modo de corrección: se ejecutará el archivo ejecutable, con revisión de código fuente.

Consideraciones:

- Formato de la entrega: Archivo comprimido donde incluirán un archivo de texto (integrantes.txt) con nombre y apellido de los integrantes del grupo, el ejecutable de su programa y el código fuente del programa.
- Forma de entrega: Uno de los integrantes del grupo deberá subir el archivo comprimido a moodle, en la opción habilitada para tal fin.
- Aprobación: Aprobarán el proyecto aquellos grupos que hayan cumplido con las fechas de entrega y con las especificaciones del programa pedido para cada entrega.