

Estructura de Computadores I

Práctica 3 (curso 2014/2015)

Actividad **Evaluable** (5% sobre la nota final)

Fecha de entrega: domingo 29 de marzo de 2015

(Hasta las 23:55h a través de Campus Extens)

Introducción

El método de **ordenación por selección** es uno de los algoritmos más sencillos para ordenar un vector de N elementos. Este algoritmo consiste en encontrar el menor de todos los elementos del vector e intercambiarlo con el que está en la primera posición. Esta operación se repite para la segunda posición del vector y así sucesivamente hasta ordenar todo el vector.

De forma general, el algoritmo funciona de la siguiente forma:

- Para cada una de las posiciones k del vector:
 - Se busca el mínimo elemento entre la posición k (incluida) y el final del vector de elementos.
 - Se intercambia este valor con el elemento de la posición k .

A modo de ejemplo, considérese el siguiente vector de 6 componentes, el cual se desea ordenar. En cada una de las iteraciones, se indica:

- En **rojo**, la posición para la cual estamos buscando el correspondiente valor mínimo.
- En **azul**, los valores del vector que se tienen en cuenta en la búsqueda del valor mínimo en esa iteración.
- En **verde**, el valor mínimo encontrado.

Así pues, entre una iteración y la siguiente, se intercambian los valores en rojo y verde, quedando la ejecución del algoritmo de la siguiente forma:

5	3	8	10	1	4	Se cambia el 5 por 1
1	3	8	10	5	4	En este caso, coincide el mínimo y el actual
1	3	8	10	5	4	Se cambia el 8 por el 4
1	3	4	10	5	8	Se cambia el 10 por el 5
1	3	4	5	10	8	Se cambia el 10 por el 8
1	3	4	5	8	10	El vector está totalmente ordenado

Para más información sobre el algoritmo, se puede consultar la página siguiente:
http://es.wikipedia.org/wiki/Ordenamiento_por_selecci%C3%B3n

Enunciado

En esta tercera práctica de programación en lenguaje ensamblador 68k, debéis generar un programa que implemente el algoritmo de ordenación por selección. El programa tendrá como entrada un número natural **N** y un vector **V** de tamaño **N**, que contendrá el vector de elementos a ordenar. Tras la ejecución del algoritmo, **V** contendrá el vector de elementos ordenado de menor a mayor. Adicionalmente, el programa debe generar un vector **VIN** de tamaño **N** que incluirá los elementos del vector **V** en orden inverso, es decir, ordenados de mayor a menor.

La cabecera del programa debe ser inexcusablemente la siguiente:

	ORG	\$1000
N:	EQU	6
V:	DC.B	5, 3, 8, 10, 1, 4
VIN:	DS.B	N
	DS.W	0

El resto de variables que sean necesarias para implementar el algoritmo deberán declararse tras la última sentencia de la cabecera. El programa tiene que funcionar para cualquier **N**, donde **N** es un valor entre 2 y 255. Nótese que **V** está definido como un vector de bytes, y por tanto cada uno de sus elementos puede tomar un valor entre 0 y 255.

Los programas que no respeten las anteriores restricciones no podrán conseguir la calificación de aprobado.

A modo de ejemplo, tras la ejecución del algoritmo con la cabecera indicada arriba, los vectores **V** y **VIN** deben quedar en memoria de la siguiente forma:

V	01	03	04	05	08	0A
VIN	0A	08	05	04	03	01

Entrega

La práctica debe entregarse de forma electrónica antes de las **23:55 horas** del domingo día **29 de marzo de 2015**, a través de un enlace que se dispondrá en la página web de la asignatura a tal efecto.

Para realizar la práctica se pueden formar **grupos de hasta 2 personas**. Sólo en casos excepcionales se podrá optar por formar grupos de 3 personas y, en tal caso, los alumnos deberán plantearle al profesor la formación de dicho grupo por adelantado para que evalúe si el grupo es viable.

A través de Campus Extens, debéis entregar únicamente el **fichero con el código fuente del programa en formato .X68**. El nombre de dicho fichero debe tener el siguiente formato: **dni1-dni2.X68**, donde **dni1** y **dni2** son los números de DNI/NIE/pasaporte de los alumnos/as que componen el grupo. En caso de tratarse de un grupo de 3 personas, el identificador del tercer miembro debe concatenarse a los otros números siguiendo el

mismo patrón. Si se respeta correctamente la nomenclatura de los ficheros, únicamente es necesario que se suba el fichero a la plataforma a través de **un único miembro del grupo**. No son necesarias dos copias del mismo código para el mismo grupo.

Evaluación

La nota de la práctica estará especificada con un valor entre 0 y 10, con dos decimales. Los aspectos más relevantes que se evaluarán serán los siguientes:

- **Ejecución correcta del programa** para cualquier valor ***N*** y vector ***V***, de acuerdo con las especificaciones de este documento.
- **Claridad del código fuente**. Se debe presentar el **código indentado**, debe haber una elección adecuada de **nombre de variables y etiquetas**, y, finalmente, debe haber suficientes, pero no excesivos, **comentarios**.
- Relacionado con el punto anterior, se deberá tener además en cuenta que **ninguna línea de código (con comentarios incluidos) podrá tener una longitud mayor a 80 caracteres**. En los siguientes recuadros, se muestran dos programas: el primero de ellos sería penalizado ya que posee varias líneas con más de 80 caracteres; el segundo programa, en cambio, no sería penalizado ya que ninguna de sus líneas sobrepasa los 80 caracteres.

El **no cumplimiento** de alguno de los requisitos especificados en este enunciado implicará una **penalización** en la calificación final de la práctica.

```

; ;La siguiente línea tiene una longitud de 80 caracteres!
*-----
A: DC.W 40
B: DC.W 13
C: DS.W 1
START:
    ; Este programa realiza la suma en complemento a 2 de dos variables de tipo word, denominadas A y B.
    ; Además, guarda el resultado de la operación en otra variable, también de tipo word, denominada C.
    MOVE.W A,D0
    ADD.W B,D0
    MOVE.W D0,C
    SIMHALT
END START

```

```

; ;La siguiente línea tiene una longitud de 80 caracteres!
*-----
A: DC.W 40
B: DC.W 13
C: DS.W 1
START:
    ; Este programa realiza la suma en complemento a 2 de dos variables de
    ; tipo word, denominadas A y B. Además, guarda el resultado de la
    ; operación en otra variable, también de tipo word, denominada C.
    MOVE.W A,D0
    ADD.W B,D0
    MOVE.W D0,C
    SIMHALT
END START

```