

Práctica 1

Introducción al Simulador de la Computadora Mejorada

Ejercicio 1: (1 punto)

Realizar un programa en SiCoMe que contenga las siguientes subrutinas:

- Subrutina 1: Dado un número almacenado en el acumulador, calcular su complemento a 2. Al finalizar la subrutina, el resultado debe quedar en el acumulador.
- Subrutina 2: Dado un número en el acumulador, calcular su valor absoluto, el resultado debe quedar en el acumulador.

El código inicial del programa debe comprobar el correcto funcionamiento de ambas subrutinas.

Ejercicio 2: (1,5 puntos)

Realizar un programa en SiCoMe que contenga una subrutina que utilice dos posiciones de memoria predefinidas como punteros a dos números A y B. Al ser invocada, la subrutina comprueba si $A > B$, en cuyo caso intercambia los valores de ambas posiciones de memoria; en caso de que $A \leq B$, no hace nada. Para codificar la subrutina, se pueden reservar las posiciones de memoria que se consideren oportunas.

Ejemplo:

Dirección	Significado	Valor
0	Ptr. A	2
1	Ptr. B	3
2	Valor A	9
3	Valor B	0



Dirección	Significado	Valor
0	Ptr. A	2
1	Ptr. B	3
2	Valor A	0
3	Valor B	9

El código inicial del programa debe realizar pruebas de invocación de la subrutina para comprobar ambos casos.

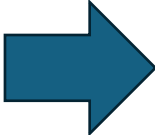
Ejercicio 3: (2 puntos)

Se tiene un vector de N números alineados en la memoria. Realizar un programa en SiCoMe que, **mediante un bucle**, sustituya todos los elementos del vector por su valor absoluto. Para su codificación, se pueden reservar las posiciones de memoria que se consideren oportunas.

Práctica 1

Introducción al Simulador de la Computadora Mejorada

Ejemplo:

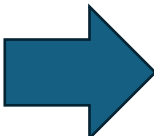
Dirección	Significado	Valor		Dirección	Significado	Valor
4	T[0]	0005		4	T[0]	0005
5	T[1]	FFFF		5	T[1]	0001
6	T[2]	0009		6	T[2]	0009
7	T[3]	FFF2		7	T[3]	000D
8	T[4]	FFFF		8	T[4]	0001

Ejercicio 4: (2,5 puntos) :

Se tiene un vector de N números alineados en la memoria. Realizar un programa en SiCoMe que contenga una subrutina que utilice una posición de la memoria predefinida con un índice i . Al finalizar la subrutina, debe quedar en el acumulador el valor i -ésimo de la tabla (indexación desde 0). En el caso de que el índice se salga de la tabla, debe quedar en el acumulador el valor 7FFF. Para codificar la subrutina, se pueden reservar las posiciones de memoria que se consideren oportunas.

El código inicial del programa debe realizar pruebas de invocación de la subrutina para comprobar ambos casos.

Ejemplo:

Dirección	Significado	Valor		i	ACUMULADOR
0	Ptr. Inicial del vector	3		1	6
1	Ptr. Final del vector	8		9	7FFF
2	Índice (i)	i			
3	V[0]	5			
4	V[1]	6			
...			
8	V[15]	FFFF			


Práctica 1

Introducción al Simulador de la Computadora Mejorada

Ejercicio 5: (3 puntos):

Se tiene un vector de n elementos alineados en la memoria. Considerando que es una memoria cíclica, realizar un programa que desplace todos los números a la posición anterior, quedando el primero en la posición del último. Por ejemplo:

Dirección	Significado	Valor
8	V[0]	0005
9	V[1]	0001
A	V[2]	FFFD
B	V[3]	000A
C	V[4]	000B
D	V[5]	FFFF



Dirección	Significado	Valor
8	V[0]	0001
9	V[1]	FFFD
A	V[2]	000A
B	V[3]	000B
C	V[4]	FFFF
D	V[5]	0005

Para codificar el programa se pueden reservar las posiciones de memoria que se consideren oportunas. El programa tiene que funcionar para cualquier tabla de tamaño n .

Cada ejercicio deberá ir acompañado de un ejemplo que verifique el correcto funcionamiento del mismo; esto es condición indispensable para la evaluación de la práctica.