APUNTE ITERACIONES, ARREGLOS Y MODO INDIRECTO ORGANIZACIÓN Y ARQUITECTURA DE COMPUTADORAS	COMISIÓN 2 2015

Iteraciones

Existen muchos problemas que requieren iteraciones para ser resueltos, es decir, requieren realizar la misma operación varias veces hasta llegar al resultado final. Por ejemplo, supongamos que queremos realizar una multiplicación sin utilizar la operación 'producto'.

Calcular A*B se podría realizar sumando B veces el valor A (o viceversa). Pensándolo como un programa podría ser:

1. Inicializar un acumulador R en cero.
2. Verificar si B es cero.
3. Si B es cero, salir.
4. Asignar R ← R + A ; Recordemos que la primera vez R es cero
5. Asignar B ← B - 1 ; Al ir restando 1 a B nos acercamos al final del problema
6. Volver al paso 2

Esto se puede resolver con lo que conocemos de la arquitectura Q, ya que con saltos condicionales podemos realizar el paso 3.

Ejercicio 1: intentá resolver el problema anterior utilizando lo que conoce de Q (¡sin leer la solución a continuación!).

Solución:

Dale, en serio, intentá resolverlo sin leer la solución, que se encuentra en algún lado en este apunte.
Pista: si imprimís este apunte, nunca vas a encontrar las soluciones.

Arreglos

Los arreglos son colecciones de elementos, contenidas en posiciones de memoria consecutivas. Todos los elementos ocupan lo mismo ya que si el primer elemento ocupa una celda, el resto también lo hará. Son, a su vez por definición, los equivalentes en programación de las matrices y vectores de las matemáticas. Precisamente, una gran motivación para usar arreglos es que hay mucha teoría detrás de ellos que puede ser utilizada en el diseño de algoritmos para resolver problemas verdaderamente interesantes.

El tamaño de un arreglo estará definido por la cantidad de elementos en el mismo.
Ejemplo:

Posición	Contenido
0X000A	1100
0X000B	A5EE
0X000C	0000
0X000D	AFAF
0X000E	FFFF

UN ELEMENTO POR CELDA
TAMAÑO 5.

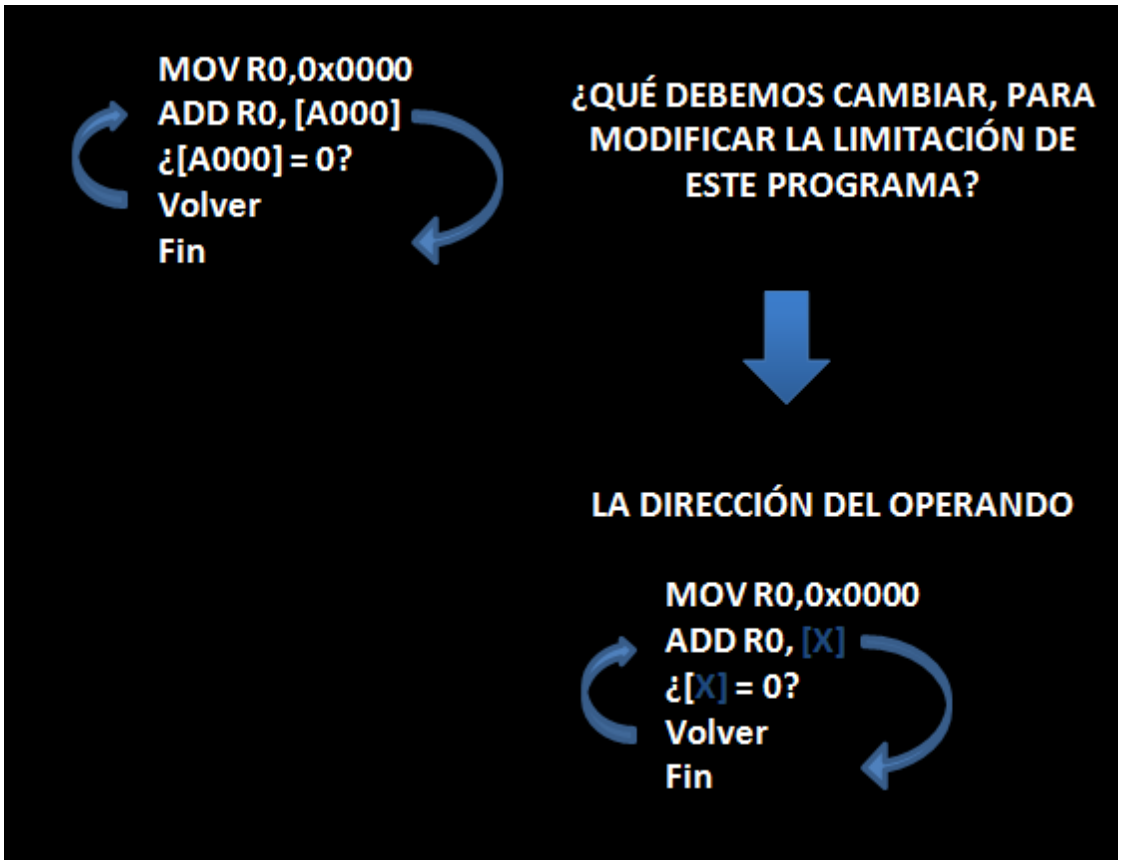
¿Cuándo finaliza un arreglo?

Podemos, tener dos posibilidades:

1. Podemos conocer el tamaño del arreglo y por lo tanto la cantidad de elementos.
2. Conocemos la condición de fin del arreglo, por ejemplo, un elemento con valor FFFF indica el final (y no es un elemento del arreglo).

Realizamos un ejercicio para poner en práctica lo visto:

A partir de la celda A000 tenemos un arreglo que contiene los alquileres de películas de un videoclub y que finaliza con el primer valor 0. Realizar un programa que sume todos los valores de dichos alquileres.



Podemos concluir que para adecuar nuestra situación debemos introducir otro modo de direccionamiento**.**

Modo de direccionamiento Indirecto:
En los modos de direccionamiento indirecto, el operando de la instrucción indica la localización de la dirección efectiva del operando. A su vez el modo de direccionamiento indirecto puede adquirir diferentes formas, según cual sea el lugar donde se encuentre la dirección del operando.

Indirecto registro:
En el operando se especifica el registro que indica donde se encuentra la dirección de memoria en la que se encuentra el operando o donde hay que dejar el resultado.

Ejemplo:

```
MOV R0, [R5]
```

En este caso , podemos especificar que estaremos accediendo a memoria, lectura una vez.

Indirecto memoria:
En el operando se especifica una dirección a memoria, cuyo contenido a su vez es otra dirección de memoria, donde se encuentra el operando o donde hay que dejar el resultado.

Ejemplo:

```
MOV R0, [[FFFF]]
```

En este caso, podemos especificar, que estaremos realizando 2 accesos a memoria.

Ahora si, luego de los contenidos abordados, ya te encontrarás en condiciones de realizar el siguiente ejercicio:

Ejercicio 2: A partir de la celda A000 tenemos un arreglo que contiene los alquileres de películas de un videoclub y que finaliza con el primer valor 0. Realizar un programa que sume todos los valores de dichos alquileres.

Solución: ¿Otra vez te apresurás a buscar la respuesta? Intentá hacerlo y cuando termines buscá bien porque en este apunte, vas a encontrar el programa resuelto. **Pista**: si imprimís este apunte, nunca vas a encontrar las soluciones.

Ejercicio 3: Supongamos un arreglo A, cuya dirección de inicio se encuentra en R0 y su dirección de fin en R1. Dicho arreglo contiene valores en BSS(16). Debe construir una rutina llamada '*invierte'*, la cual debe armar un nuevo arreglo B que empiece en la dirección B000, que tenga el mismo contenido que A pero en orden inverso (al de A). Por ejemplo, si en A se encuentran los valores 0x0004, 0x0F76 y 0x0099, entonces en el arreglo B deben quedar los valores 0x0099, 0x0F76 y 0x0004.

Solución: Hmmm, no sé si vas a encontrar la solución a este problema en el apunte. De verdad, pensalo y acordate que algo de lo visto en el apunte te va a ser útil.

Soluciones

Ejercicio 1 (Multiplicación utilizando saltos condicionales):

Ejercicio 2 (Sumar valores en un arreglo)