

MÓDULO 2 | PROGRAMACION COBOL– Métodos de Manipulación de Datos

1 METODOS DE MANIPULACIÓN DE DATOS

En esta sección analizaremos los métodos utilizados para manipulación de datos básicos, como ser:

- Cálculo de máximos y mínimos,
- Cálculo de promedio,
- Ordenamiento de un conjunto de elementos según un orden preestablecido
- Corte de Control

1.1 Cálculo de Máximos y Mínimos

Un máximo, es el límite superior a que puede llegar algo. Es decir, si C es un numero tal que $C > C_1, C_2 \dots C_n$, entonces C sería el **máximo**.

Un mínimo sería el limite inferior a que se puede reducir algo. Es decir, si C es numero tal que $C < C_1, C_2 \dots C_n$, entonces C sería el **mínimo**.

Para explicar mejor lo anteriormente citado se lo explicara a través de un ejemplo.

Ejemplo:

Buscar entre los siguientes 10 números de 2 cifras los valores máximos y mínimos:

10	20	05	30	14	29	08	09	19	18
----	----	----	----	----	----	----	----	----	----

En este caso el valor máximo es el 30, y el mínimo el 5.

Traducido a COBOL, ambos casos se resuelven con un ciclo que deberá recorrer secuencialmente y hasta el fin un conjunto de datos que puede surgir de un archivo, vector, tabla, etc.; ademas se deberá declarar una variable auxiliar a la cual por cada iteración se le deberá mover el valor leído si se cumpliera la condición que buscamos (que sea mayor o menor que el valor de la variable auxiliar, dependiendo el caso). Cuando terminemos de leer todos los datos, la variable auxiliar tendrá almacenado el valor que deseamos obtener:

	1	2	3	4	5	6	7	
12345678901	2345678901	2345678901	2345678901	2345678901	2345678901	2345678901	2345678901	23...
	<pre> MOVE ZEROES TO AUXILIAR PERFORM VARYING INDICE FROM 1 BY 1 UNTIL INDICE > 10 IF VECTOR(INDICE) > AUXILIAR MOVE VECTOR(INDICE) TO AUXILIAR END-IF END-PERFORM </pre>							

Búsqueda del máximo – La variable auxiliar se inicializa con cero.

Búsqueda del mínimo—La variable auxiliar debe inicializarse con el máximo valor que le permita su declaración, en este caso $MAX = 99$

El promedio, de una cantidad finita de números, es una media estadística muestral determinada por la suma de todos ellos dividida entre el número de sumandos.

Matemática	-> 5
Lengua	-> 8
C. Sociales	-> 9
C. Naturales	-> 8

Traducido a COBOL, se resuelve con un ciclo que deberá recorrer secuencialmente y hasta el fin un conjunto de datos que puede surgir de un archivo, vector, tabla, etc.; además se deberán declarar las variables "Promedio" que guardará el resultado, un acumulador en el que se irán sumando los valores leídos y una variable auxiliar que va a contar la cantidad de valores sumados. Al finalizar el ciclo, se deberá dividir el acumulador por la variable.

En este ejemplo se conoce la cantidad de elementos y es utilizada en la condición del ciclo.

Cálculo del promedio

1.3 Métodos de Ordenamiento: Burbuja

Este algoritmo funciona comparando cada elemento de la lista que va a ser ordenada con el siguiente, intercambiándolos de posición si estuvieran en el orden equivocado. Es necesario recorrer varias veces toda la lista para que no se necesiten más intercambios, lo cual significa que la lista está ordenada.

Para ordenar una lista en forma ascendente se debe recorrer $n - 1$ veces la lista (siendo n el total de elementos que la componen), comparando elementos adyacentes de dos en dos. Si un elemento es mayor que el que está en la siguiente posición, se intercambian. El nombre del método es una analogía que hace referencia a los elementos menos pesados que suben a la vez que los mas pesados bajan.

Utilizando términos técnicos, se debe utilizar una variable auxiliar y dos ciclos anidados. El ciclo externo controla las iteraciones del ciclo interno hasta el límite de tamaño de la lista (MAX). El ciclo interno es el que realiza la comparación e intercambio de posiciones entre los elementos adyacentes, recorriendo la lista hasta $MAX - I$ (siendo I el índice del ciclo externo que nos indica la posición en la lista).

A continuación se muestra el algoritmo:

Ciclo externo

Ciclo interno

Comparación de elementos adyacentes

MAX = 8 -> Es la longitud del vector para este ejemplo.

I = J = 1 -> Valor inicial de los índices.

1	2	3	4	5	6	7
123456	890123456789012345678901234567890123456789012345678901234567890123...					
PERFORM VARYING I FROM 1 BY 1 UNTIL I > MAX						
PERFORM VARYING J FROM 1 BY 1 UNTIL J > MAX - I						
IF ITEM(J) > ITEM(J+1)						
MOVE ITEM(J) TO AUXILIAR						
MOVE ITEM (J+1) TO ITEM(J)						
MOVE AUXILIAR TO ITEM(J+1)						
END-IF						
END-PERFORM						
END-PERFORM						

Algoritmo de ordenamiento de burbuja ascendente

- Primera iteración del ciclo externo: I vale 1 y J va incrementándose hasta $MAX - I$ cuyo resultado, en este caso, es 7. A medida que J incrementa su valor, compara elementos adyacentes y los intercambia si el valor en el orden J es mayor al de la posición $J + 1$. Al completar la primera vuelta del ciclo I , el elemento mayor (en este caso 8) quedará al final de la lista:

INDICE	VECTOR	I=1							RESULTADO DE LA ITERACION
		J=1	J=2	J=3	J=4	J=5	J=6	J=7	
1	8	4	4	4	4	4	4	4	4
2	4	8	2	2	2	2	2	2	2
3	2		8	3	3	3	3	3	3
4	3			8	5	5	5	5	5
5	5				8	1	1	1	1
6	1					8	7	7	7
7	7						8	6	6
8	6							8	8

2. Segunda iteración del ciclo externo: I vale 2 y en este caso $MAX - I$ vale 6. Al igual que en el paso anterior, a medida que J incrementa su valor, compara elementos adyacentes y los intercambia si el valor del ítem J es mayor al de la posición $J + 1$. Al completar la primera vuelta del ciclo I , el elemento mayor (en este caso 7) quedará al final de la lista:

INDICE	VECTOR	I=2						RESULTADO DE LA ITERACION
		J=1	J=2	J=3	J=4	J=5	J=6	
1	4	2	2	2	2	2	2	2
2	2	4	3	3	3	3	3	3
3	3		4	4	4	4	4	4
4	5			5	1	1	1	1
5	1				5	5	5	5
6	7					7	6	6
7	6						7	7
8	8							8

3. Tercera iteración del ciclo externo: I vale 3 y $MAX - I$ vale 5. Al completar otra vuelta del ciclo I , el elemento mayor (en este caso 6) quedará al final de la lista, como indica la tabla:

INDICE	VECTOR	I=3					RESULTADO DE LA ITERACION
		J=1	J=2	J=3	J=4	J=5	
1	2	2	2	2	2	2	2
2	3	3	3	3	3	3	3
3	4		4	1	1	1	1
4	1			4	4	4	4
5	5				5	5	5
6	6					6	6
7	7						7
8	8						8

4. Cuarta iteración del ciclo externo: I vale 4 y $MAX - I$ es 4. Al completar otra vuelta del ciclo I , el elemento mayor (en este caso 5) quedará al final de la lista, como indica la tabla:

INDICE	VECTOR	I=4				RESULTADO DE LA ITERACION
		J=1	J=2	J=3	J=4	
1	2	2	2	2	2	2
2	3	3	1	1	1	1
3	1		3	3	3	3
4	4			4	4	4
5	5				5	5
6	6					6
7	7					7
8	8					8

5. Quinta iteración del ciclo externo: I vale 5 y $MAX - I$ es 3. Al completar otra vuelta del ciclo I , el elemento mayor (en este caso 4) quedará al final de la lista, como indica la tabla:

INDICE	VECTOR	I=5			RESULTADO DE LA ITERACION Y FINAL
		J=1	J=2	J=3	
1	2	1	1	1	1
2	1	2	2	2	2
3	3		3	3	3
4	4			4	4
5	5				5
6	6				6
7	7				7

- No conocemos la cantidad de movimientos que tiene cada cliente, entonces no se puede armar un ciclo que cuente la cantidad de comprobantes ingresados.
- No existe la misma cantidad de comprobantes para todos los clientes.

Conocidas las restricciones, debemos resolver cómo armar una rutina que se repita mientras mientras no cambie el dato de comparación, en este caso el número del cliente.

Este procedimiento se denomina comúnmente ***rutina de corte de control***, acumula los importes leídos mientras el número del cliente sea *igual al anterior*, cuando sea distinto deberá mostrar la suma acumulada y limpiar la variable utilizada para realizar el mismo proceso con otro cliente.

Para que este procedimiento funcione correctamente, es necesario contemplar que se den las siguientes condiciones:

- Los datos deben estar ordenados por el concepto por el cual realizamos el agrupamiento (en nuestro caso, número de cliente)
- Al cambiar el concepto por el cual se controla el corte (número del cliente) se deben emitir los totales y borrarse los acumulados.
- Utilizar una variable auxiliar para guardar el dato por el cual se está controlando el número de cliente, y compararlo contra el que se está leyendo y así verificar si debemos cumplir con el punto anterior.

El usuario que opere con el módulo que tomamos como ejemplo deberá ingresar por teclado por cada movimiento de cuenta (depósito o extracción de dinero), el número del cliente que realiza la operación y el monto por el cuál opera; al finalizar la carga de operaciones de un cliente, se imprimirá por pantalla el saldo de sus movimientos. Para finalizar, debe cargar un número de cliente igual a cero