

consultoría

sa

✉ Avenida de Mayo 666 - Piso 6  
☎ +54 11 5235 3555  
✉ info@sistemasactivos.com

proyectos

# Guía de Ejercicios

- Octubre 2007 -

## Módulo I

### Problema #1

Defina una estructura en memoria conveniente para almacenar una fecha en formato Día-Mes-Año de la forma DD-MM-AAAA (2 dígitos para el día, 2 dígitos para el mes, 4 dígitos para el año). Compile el código fuente requerido para verificar la sintaxis.

### Problema #2

Realice el código fuente de un programa que pida al usuario las tres partes de la fecha del Problema #1, y las presente con los guiones en pantalla. Considerando que la salida del mismo deberá ser como sigue.

```

I N G R E S E   E L   D I A :
0 6
I N G R E S E   E L   M E S :
0 3
I N G R E S E   E L   A Ñ O :
2 0 0 5
L A   F E C H A   D E L   D I A   E S   0 6 - 0 3 - 2 0 0 5

```

Compile y ejecute el programa y verifique que el resultado sea el esperado.

### Problema #3

Se tiene la siguiente estructura de memoria para sostener una fecha en formato DD-MM-AA (2 dígitos para cada parte de la fecha).

```

0 1   F E C H A .
      0 2   D I A   P I C   9 ( 0 2 ) .
      0 2   F I L L E R   P I C   X ( 1 )
                               V A L U E   " / " .
      0 2   M E S   P I C   9 ( 0 2 ) .
      0 2   F I L L E R   P I C   X ( 1 )
                               V A L U E   " / " .
      0 2   A N I O   P I C   9 ( 0 2 ) .

```

Al final del año 2000, la misma sufrió una modificación incorporando a la estructura el concepto de siglo (de forma que para el año 2000 se asuma valor de siglo 20 lo que tiene un fin práctico ya que el siglo en realidad debería ser 19), como rediseñaría dicha estructura para incorporar la variable siglo. Compile el código fuente requerido para verificar la sintaxis.

#### Problema #4

Se tiene la presente declaración de variables:

7	7	E	S	T	A	D	O	-	C	I	V	I	L	P	I	C	X	.
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

La variable ESTADO-CIVIL, se utiliza de modo que su valor representa:

Valor	Significado
C	Casado
S	Soltero
V	Viudo
D	Divorciado

Incorpore los niveles 88 para definir las variables booleanas que se correspondan con estas situaciones. Compile el código fuente requerido para verificar la sintaxis.

#### Problema #5

Se tiene una variable RIESGO, que representa una categorización del 1 al 9, se pide realizar la definición de datos (incluyendo las variables booleanas necesarias) sabiendo que la variable se utilizará en un procedimiento de aprobación que aprobará a todas las operaciones que presenten RIESGO menor a 3. Compile el código fuente requerido para verificar la sintaxis.

#### Problema #6

Codificar la definición de variables útil para recibir el resultado a una pregunta que debe ser respondida con (S/N) y adicionarle las variables booleanas SI, NO, RESPUESTA-VALIDA con sus respectivos valores. Compile el código fuente requerido para verificar la sintaxis.

#### Problema #7

La estructura de una cuenta bancaria generalmente es la siguiente 999-99-99999-9 donde los primeros tres dígitos identifican a la sucursal bancaria, los dos que siguen definen al producto (Caja de ahorro, préstamo prendario, préstamo hipotecario, etc.), luego está el número de la cuenta propiamente dicho y finalmente un dígito verificador que comprueba la consistencia del dato completo.

Defina la estructura de variables más conveniente para almacenar una cuenta bancaria con el uso previsto e incorpore las variables booleanas para caja de ahorro, préstamo prendario, préstamo hipotecario y cuenta corriente, sabiendo que los códigos a los que responden son 40, 50, 60 y 45 respectivamente. Compile el código fuente requerido para verificar la sintaxis.

#### Problema #8

Se pide definir una variable capaz de almacenar un número que toma valor en el rango -45834 al +49234. ¿Cuántos bytes ocuparía en cada una de las tres formas de almacenamiento posibles?

#### Problema #9

Realice un programa que pregunte el nombre y apellido de una persona y los presente en pantalla. Compile y ejecute el programa y verifique que el resultado sea el esperado.

### Problema #10

Realice un programa que pregunte por un monto, y luego informe el IVA correspondiente a dicho monto (el 21%).

### Problema #11

Las variables definidas con PIC X tienen la propiedad de poder ser tratadas por partes, por ejemplo si se tiene una variable NOMBRE PIC X(20). Puedo obtener las primeras 5 letras del nombre haciendo uso de la variable NOMBRE(1:5) donde el 1 indica el número de carácter a partir del cual esta la información de interés y el 5 indica la cantidad de caracteres que me interesan. Aplique esta propiedad para mostrar las partes de una fecha almacenada en una variable FECHA PIC X(10) VALUE '2006-04-18'. La salida del programa debe quedar entonces:

A	Ñ	O	:	2	0	0	6
M	E	S	:	0	4		
D	I	A	:	1	8		

Compile y ejecute el programa y verifique que el resultado sea el esperado.

### Problema #12

Modifique el programa del problema anterior para presentar la fecha en formato DD/MM/AAAA. Compile y ejecute el programa y verifique que el resultado sea el esperado.

### Problema #13

Los registros tienen la propiedad de mostrar la información de todas las variables que lo componen como una cadena alfanumérica del tamaño que ocupa en bytes la información que contiene. De esta forma, a través de la siguiente estructura se puede analizar el byte de zoneado.

0	1	Z	O	N	E	A	D	O	.						
0	2	N	U	M		P	I	C	S	9	(	0	1	)	.

Realice un programa que establezca el valor -3 a la variable NUM y presente en pantalla la variable ZONEADO. Compile y ejecute el programa y verifique que el resultado sea el esperado.

### Problema #14

Extienda el programa anterior para componer la tabla de valores de los bytes de zoneado que se corresponden con los valores del -9 al 9. Compile y ejecute el programa y verifique que el resultado sea el esperado. Conserve la salida del programa en un archivo agregando en la invocación del programa (RUNCOBOL PROGRAMA > ARCHIVO.TXT).

### Problema #15

Modifique la definición de variables del problema anterior para verificar cual sería el efecto de declarar la variable numérica como empaquetada decimal (PACKED-DECIMAL). Compile y ejecute el programa y verifique que el resultado sea el esperado.

### Problema #16

Teniendo en cuenta la siguiente estructura de memoria.

0 1	P R A C T I C A .		
0 2	N U M 1	P I C	S 9 ( 0 5 ) .
0 2	N U M 2	P I C	S 9 ( 0 7 ) .
0 3	N U M 3	P I C	9 ( 0 4 ) .

Un programa presento el valor de la variable PRACTICA y obtuvo la siguiente salida.

2	3	4	1	{	3	2	4	8	9	5	N	5	3	7	6
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Determine los valores de las variables NUM1, NUM2 y NUM3. Se recomienda hacer uso de la tabla de valores obtenida en el Problema #14.

### Problema #17

Teniendo en cuenta la siguiente estructura de memoria.

0	1	P R A C T I C A .
0	2	N U M 1 P I C S 9 ( 0 5 ) .
0	2	N U M 2 P I C S 9 ( 0 7 ) .
0	3	N U M 3 P I C 9 ( 0 4 ) .

Un programa presento el valor de la variable PRACTICA y obtuvo la siguiente salida.

5 3 8 9 R 1 2 6 7 9 5 R 4 3 2 6

Determine los valores de las variables NUM1, NUM2 y NUM3. Intente hacer automáticamente la conversión a través de un programa que cargue el valor de la variable PRACTICA y muestre por pantalla los valores de las variables NUM1, NUM2 y NUM3. Compile y ejecute el programa y verifique que el resultado sea el esperado.

### Problema #18

Quando se define una variable PIC 9(04) BINARY se esta definiendo una variable de 2 bytes que puede almacenar números entre 0 y 9999. En particular los valores entre el 0 y el 255. Se almacenan en su segundo bytes, la representación en pantalla de ese bytes se hace a través de la tabla ASCII. Aplique todo lo utilizado hasta el momento para obtener a través de un programa que utilice una conveniente definición de memoria el carácter ASCII que se corresponde con el valor 66. Compile y ejecute el programa y verifique que el resultado sea el esperado.

### Problema #19

Realice un programa dividido en los párrafos 100000-CARGA, 200000-CALCULO y 300000-RESULTADO, que obtenga los valores de 5 importes, totalice su valor y calcule su IVA e informe monto total e IVA.

## Estructuras de decisión.

### Problema #1

Realizar un programa que pida dos números al usuario y los muestre ordenados de menor a mayor. Compile y ejecute el programa y verifique que el resultado sea el esperado.

### Problema #2

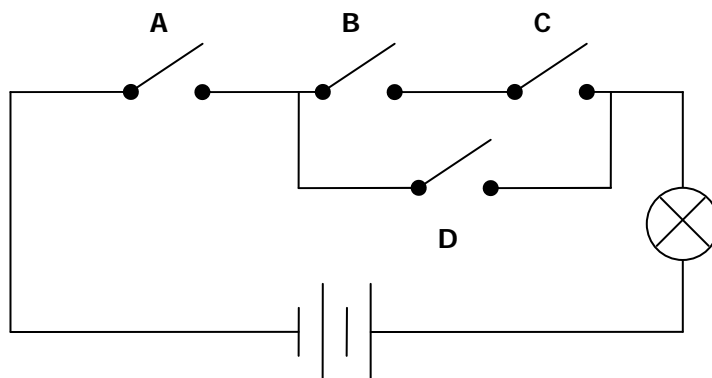
Se tienen las siguientes reglas de ingreso a un club:

- Altura mínima (1,60m para las mujeres; 1,70m para los hombres).
- Peso máximo (60Kg para las mujeres; 90Kg para los hombres).
- No tener antecedentes cardiacos.

Hacer un programa que pida al usuario los datos personales de los postulantes y los califique en APTOS y NO APTOS. Compile y ejecute el programa y verifique que el resultado sea el esperado.

### Problema #3

Se tiene una instalación eléctrica como la que sigue:



Si la lámpara (representada por el círculo) enciende solo cuando existe al menos un camino cerrado en el circuito, hacer un programa que pida al usuario el estado de las 4 llaves (Abiertas – Cerradas) e informe en que estado debería estar la lámpara. Compile y ejecute el programa y verifique que el resultado sea el esperado.

### Problema #4

Hacer un programa que pida al usuario las longitudes de los tres lados de un triángulo y luego dictamine si se trata de un triángulo EQUILÁTERO, ISÓSCELES ó ESCALENO. Compile y ejecute el programa y verifique que el resultado sea el esperado.

### Problema #5

Hacer un programa que pida al usuario las longitudes de los cuatro lados de una figura geométrica y califique la figura como un CUADRADO, RECTÁNGULO, ó ninguno de los dos anteriores. Compile y ejecute el programa y verifique que el resultado sea el esperado.

### Estructuras de repetición.

#### **Problema #1**

Realizar un programa que presente en pantalla los números del 1 al 10. Compile y ejecute el programa y verifique que el resultado sea el esperado.

#### **Problema #2**

Realizar un programa que presente en pantalla los números pares del 1 al 16. Compile y ejecute el programa y verifique que el resultado sea el esperado.

#### **Problema #3**

Realizar un programa que presente en pantalla los números impares del 1 al 30. Compile y ejecute el programa y verifique que el resultado sea el esperado.

#### **Problema #4**

Codificar un programa que calcule el factorial de un número ingresado por el usuario y lo informe en pantalla. Compile y ejecute el programa y verifique que el resultado sea el esperado.

#### **Problema #5**

Realizar un programa que calcule el máximo de 20 números ingresados por el usuario. Compile y ejecute el programa y verifique que el resultado sea el esperado.

## Módulo II

### Vectores y Redefines.

#### Problema #1

Realizar un programa que pida al usuario los números para cargar los valores de una matriz de 5 x 5 por filas y luego calcule el valor de la suma de los elementos de la diagonal principal. Compile y ejecute el programa y verifique que el resultado sea el esperado.

#### Problema #2

Realizar un programa que calcule los primeros 20 términos de la serie de Fibonacci. La cual se desarrolla de la siguiente manera: 1, 1, 2, 3, 5, 8, 13,...; Los primeros 2 números siempre son unos, y luego el resto se calcula como la suma de los dos anteriores. Compile y ejecute el programa y verifique que el resultado sea el esperado.

#### Problema #3

Realizar un programa que cargue una matriz por filas y luego la muestre en pantalla por columnas. Compile y ejecute el programa y verifique que el resultado sea el esperado.

#### Problema #4

Realizar un programa que cargue una matriz con números enteros en orden ascendente en forma de espiral como se muestra en la figura. Compile y ejecute el programa y verifique que el resultado sea el esperado.

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

#### Problema #5

Implemente una función para convertir las calificaciones del 1 al 10 al sistema de calificaciones de la "A" a la "F"; siendo las correspondencias:

- A = 10.
- B = 9 y 8.
- C = 7, 6, 5 y 4.
- D = 3.
- E = 2.
- F = 1.

**NOTA:** Dicha función debe plantearse mediante un vector precargado.

Compile y ejecute el programa y verifique que el resultado sea el esperado.



## Registros y Redefines.

### Problema #1

Mediante la utilización de un registro cargar Nombre, Apellido y tres notas de un alumno, luego calcular el promedio e informarlo por pantalla con el siguiente formato:

Apellido, Nombre: Promedio (el promedio debe tener el siguiente formato 9(2)v9(2) y debe ser mostrado con coma)

### Problema #2

Armar una estructura de memoria que nos permita tener un vector precargado con la cantidad de días de cada mes del año. Compile y verifique que el resultado sea el esperado.

### Problema #3

Utilizando la estructura de memoria del ejercicio anterior realice un programa que devuelva por pantalla la cantidad de días que tiene un mes de un determinado año donde ambos son ingresados por teclado (lo que debe ingresar es el número de mes y el año, y debe considerar la posibilidad de que el año sea bisiesto). Compile y ejecute el programa y verifique que el resultado sea el esperado.

### Problema #4

Defina una estructura de memoria de la forma más adecuada para poder tomar por teclado una cadena, la cual represente a una función matemática (el formato de la función debe ser "**num1 operador num2**" donde **num1** sea 9(4), **num2** sea 9(4) y **operador** pueda ser "+", "-", "/", "\*").

Luego escriba el programa que resuelva la función ingresada y devuelva el resultado por pantalla. Compile y ejecute el programa y verifique que el resultado sea el esperado.

### Problema #5

A partir del ejercicio anterior genere un nuevo programa donde los números puedan tener una longitud variable.

### Problema #6

Defina una estructura de memoria que contenga el nombre, apellido, DNI, teléfono, dirección, estado civil (Casado= "C", Soltero= "S", Viudo= "V", Divorciado= "D"), hijos (Si= "S", No="N") y grupo y factor sanguíneo (Grupo (A, B, O) y Factor (+ o -)).

Haga un programa que lea los datos de teclado y los muestre ordenadamente en pantalla. Compile y ejecute el programa y verifique que el resultado sea el esperado.

## Métodos de manipulación de datos (1ra parte).

### Problema #1

- Cargar los siguientes 10 números en un vector y buscar el máximo y el mínimo.  
10, 20, 2, 5, 4, 89, 32, 88, 99, 0.

### Problema #2

Al vector cargado en el ejercicio anterior, calcular el promedio y mostrarlo por pantalla.  
Tener en cuenta que al definir la variable donde se calculará el promedio contemple        decimales.

### Problema #3

Hacer un programa que cargue en un vector 10 registros con la siguiente estructura:

01 Notas.

05 Alumnos	Occurs 10 times.
10 Legajo	Pic 9(02).
10 Nota	Pic 9(02).

Repetir legajos en la carga, de modo que halla varios registros con un mismo legajo en el vector. Ej. 2, 1, 8, 2, 3, 3, 1, 4.... Una vez finalizada la carga, efectuar un ordenamiento del vector por la clave legajo, utilizando el método de ordenamiento de Burbuja, mostrando al final de cada ciclo de ordenamiento el orden en que van quedando los legajos en el vector.

Una vez ordenado los registros, leer el vector efectuando un corte de control por legajo para calcular el promedio de notas de cada legajo, la máxima nota y la mínima nota para ese legajo y mostrarlo por pantalla.



#### Problema #4

Realizar un programa que genere un listado por pantalla del archivo 'CUENTAS.DAT' con el siguiente formato:

```
*-----*
| NRO.CUENTA | COD.CLI. | DISPONIBLE CUENTA |
*-----*
| 99.99999.9 | 99999999 | -99999999999999,99 |
| 99.99999.9 | 99999999 | -99999999999999,99 |
| 99.99999.9 | 99999999 | -99999999999999,99 |
| 99.99999.9 | 99999999 | -99999999999999,99 |
| 99.99999.9 | 99999999 | -99999999999999,99 |
*-----*
```

#### Problema #5

Realizar un programa que obtenga el máximo y el mínimo disponible en cuenta del archivo de cuentas, además que identifique cuales son los códigos de cliente que se corresponden con dichos montos. Debe informar todo esto en pantalla.

#### Problema #6

Realizar un programa que obtenga el monto promedio de las facturas de los servicios del archivo de servicios e informe en pantalla.

#### Problema #7

Realizar un programa que lea el archivo de cuentas y el de servicios e identifique los clientes poseedores de las cuentas sobre las que se realizarán los debitos. Estos datos deben informarse en un archivo de salida 'SALDOS.DAT' con el siguiente formato:

- Cliente (Numérico de 8 enteros)
- Monto de la cuenta (15 enteros y 2 decimales con signo).

Los archivos de entrada deben ser abiertos y recorridos solo una vez cada uno.

#### Problema #8

Se tienen dos archivos de entrada el de servicios, y el de cuentas Bancarias. (Son los cargados en ejercicios anteriores)

El funcionamiento general de las cuentas es el siguiente: Los dos dígitos del producto identifican el tipo de cuenta, '01' para las cajas de ahorro, '02' para las cuentas corrientes. El cuenta es único en si mismo y el dígito verificador es solo para comprobar la consistencia del numero de cuenta.

El programa debe realizar los debitos de las facturas sobre las cuentas en el sistema, basado en lo siguiente, las cajas de ahorro por definición no pueden quedar en rojo con lo cual los debitos que generen esta situación deben ser rechazados y las cuentas corrientes no presentan restricciones.

La salida del mismo, debe ser un archivo de cuentas con sus montos actualizados, y un archivo de rechazos con los servicios que no fueron debitados.

**NOTA:** SUPONER EN ESTA INSTANCIA QUE LOS ARCHIVOS VIENEN ORDENADOS DE LA SIGUIENTE MANERA:

- SERVICIOS (Ascendente por numero de cuenta)

- CUENTAS (Ascendente por numero de cuenta)

Los archivos de SERVICIOS y CUENTAS son de acceso secuencial.

Los archivos de entrada deben ser abiertos y recorridos solo una vez cada uno.

## Problema #9

Realizar un programa que genere un listado del archivo 'CUENTAS.DAT' en un archivo de longitud de registro igual a 80 BYTES con el siguiente formato:

FECHA 99/99/9999

PAGINA 1

### LISTADO DE CUENTAS

```
*-----*-----*-----*
| CUENTA | CLIENTE | MONTO |
*-----*-----*-----*
|XX-XXXXX-X|99999999|-99999999999999,99|
.
.
|XX-XXXXX-X|99999999|-99999999999999,99|
*-----*-----*-----*
```

FECHA 99/99/9999

PAGINA 2

### LISTADO DE CUENTAS

```
*-----*-----*-----*
| CUENTA | CLIENTE | MONTO |
*-----*-----*-----*
|XX-XXXXX-X|99999999|-99999999999999,99|
.
.
|XX-XXXXX-X|99999999|-99999999999999,99|
*-----*-----*-----*
```

Notar que se debe tomar la fecha de la computadora y presentarla en el encabezado del detalle. Además, se requiere un contador de número de página y considerar que el encabezado debe imprimirse en cada nueva pagina, el criterio que se debe utilizar para identificar una nueva página es que las líneas grabadas en el archivo sean 60, entonces cada 60 líneas tendremos una nueva página.

## Métodos de manipulación de datos (2da. parte)

### Problema #1

Cargar el archivo CUENTAS con la siguiente información:

Nro. Cliente	Nombre	Saldo
1	Perez	\$40
2	Alvarez	\$30
3	Breither	\$12.5
4	Di Augusto	\$100.25
5	Meloso	\$60
6	Mars	\$32.10
7	Torres	\$23
8	Duffaur	\$15

Hacer un programa que permita al usuario ingresar el patrón de búsqueda (por Nro. de cliente o nombre), ingresar el nro. de cliente o nombre (depende del patrón seleccionado) y mostrar por pantalla el registro completo (incluyendo el saldo de dicho usuario). En caso de no encontrar el registro, mostrar un mensaje alusivo.

### Problema #2

Utilizando el archivo del ejercicio anterior y el archivo SERVICIOS con la siguiente información:

Nro. cliente	Cod. servicio	Monto
1	CTI	\$10
1	AGA	\$15
3	CTI	\$45
4	CTI	\$50
4	AGA	\$70
5	CTI	\$20
9	AGA	\$20

Hacer un programa que realice un apareo entre el archivo CUENTAS y SERVICIOS por numero de cliente. Para cada cliente restar el monto total de sus servicios al saldo de su cuenta si y solo si dispone de un saldo mayor o igual al monto total de sus servicios (la suma total de los servicios para ese cliente), grabar el registro en el archivo de CUENTAS con el saldo actualizado, en caso de que el total de montos de servicio supere al saldo de CUENTAS para ese cliente, grabar el registro del cliente en el archivo RECHAZOS con el siguiente formato:

#### RECHAZOS

Nro. Cliente, Nombre, Saldo actual de la cuenta, Imp. Deuda.

Si un nro. de cliente del archivo CUENTAS no se encuentra en el archivo SERVICIOS, grabar el registro en el archivo INCIDENCIAS con el siguiente formato:

Nro. Cliente Tabla Ausencia.

En el campo "Tabla ausencia" se grabará el nombre de la tabla en donde no se registra el dato.

EN el caso de que un nro. de cliente de la tabla SERVICIOS no se encuentre en la tabla CUENTAS, se grabara un registro en la tabla INCIDENCIAS con el formato indicado anteriormente.

## Archivos Indexados.

### Problema #1

Realizar un programa para cargar datos en el archivo CUENTASIX.DAT tomando como origen los datos del archivo CUENTAS.DAT, utilizar la siguiente estructura de registro:

- Número de cuenta (Alfanumérico de 8 caracteres). Campo Clave.
- Código de cliente (Numérico de 8 enteros sin signo).
- Monto de la cuenta (15 enteros y 2 decimales con signo).

### Problema #2

Realizar un programa que tome por teclado los valores de la clave del archivo CLIENTESIX.DAT y busque en dicho archivo el registro correspondiente a esa clave y lo muestre, en caso de que no exista información para los datos ingresados se debe mostrar un mensaje de error.

### Problema #3

Realizar un programa que tome por teclado dos números de cuenta distintos y utilice el archivo CUENTASIX.DAT para realizar un listado que contenga los registros cuyo valor en el campo clave se encuentre entre los valores ingresados anteriormente. Considerar que si no se encuentra ningún registro que cumpla esta condición no se debe generar el listado sino que se debe mostrar un mensaje de error.

### Problema #4

Realizar un programa que efectúe el ABMC (alta, baja, modificación y consulta) del archivo CLIENTESIX.DAT considerando los errores pertinentes en cada caso.

### Problema #5

Realizar un programa que genere el archivo SERVICIOIX.DAT con una clave principal y una alternativa, utilizando la siguiente estructura:

- Código de servicio (Alfanumérico de 3 caracteres). Clave principal.
- Número de cuenta (8 enteros). Clave alternativa.
- Descripción del servicio (Alfanumérico de 30 caracteres).
- Periodo facturado (Alfanumérico de 6 caracteres – Formato AAAAMM).
- Monto facturado (5 enteros y 2 decimales con signo).

Considere que bajo estas condiciones no podría existir más de un registro con la misma clave principal pero si podría existir más de un registro con la misma clave alternativa.

### Problema #6

Realizar un programa que pida por pantalla la selección de la clave por la cual se va a acceder al archivo SERVICIOIX.DAT (principal o alternativa) y en base a lo seleccionado solicite los valores correspondientes para el acceso al mismo y muestre los datos del o de los registros correspondientes con formato de listado.

## Integración

### Problema #1

Realizar un programa que pida al usuario que ingrese una frase, luego, que le pida que ingrese una palabra, en este punto el programa debe informar si esa palabra se usa o no en la frase y cuantas veces es utilizada. Si la palabra se usaba en la frase, pedir al usuario la palabra por la que debe ser reemplazada, y mostrar la frase con el reemplazo realizado.

### Problema #2

Realizar un programa que lea una operación numérica con el formato como sigue:

- $123 + 24 + 1 =$
- $24 - 10 - 23 =$
- $167 + 23 - 1 =$

El mismo debe interpretar la operación y ejecutarla informando el resultado. NOTA: Debe implementarse algún método por el cual se pase del formato alfanumérico del dato ingresado al formato numérico con el cual podamos operar. Además, tener en cuenta que siempre habrá un lugar en blanco entre cada dato y la operación, al igual que el signo "=" para cerrar la misma. Las operaciones válidas serán "+" y "-".

### Problema #3

Agregarle al programa anterior las operaciones "\*", "/", "(" y ")".

### Problema #4

Realizar un programa SORT para ordenar archivos, el mismo debe ordenar archivos de hasta 100 Bytes de longitud de registro y 50 registros. El orden del mismo debe implementarse a través de un vector. Y la decisión de sobre que campo establecer el orden, mediante una cadena de las siguientes características:

`SORT FIELDS=(P,L,O)`

Donde "P" es la posición inicial, "L" es la longitud, y "O" es el orden. Ejemplo:

`SORT FIELDS=(2,3,A)`

Este ejemplo ordenará de forma ascendente el campo que comienza en el segundo byte y tiene una longitud de 3 bytes. Otro ejemplo:

`SORT FIELDS=(1,10,D)`

Este otro ejemplo ordenará de forma descendente por los primeros 10 bytes del registro.



### Problema #5

Generar una rutina para calcular el determinante de una matriz de 2 x 2. Utilizar la siguiente área de comunicación como guía:

01 EDET-AREA.

05 EDET-I-AREA.

10 EDET-MAT.

15 EDET-FIL

OCCURS 2 TIMES.

20 EDET-COL

OCCURS 2 TIMES.

25 EDET-NUM

PIC 9(02).

05 EDET-O-AREA.

10 EDET-DET

PIC 9(04).

05 EDET-S-AREA.

10 EDET-COD-RET

PIC 9(01).

88 EDET-OK

VALUE 0.

88 EDET-ERROR

VALUE 9.

10 EDET-REF

PIC X(60).

Donde se ingresará con el área cargada con la matriz, y la rutina devolverá el valor del determinante junto con EDET-OK, ó EDET-ERROR con alguna explicación conveniente en EDET-REF.

### Problema #6

Realizar una rutina para validar fechas, y establecer su área de comunicación. Luego, probarla en los casos que se presentarán a continuación. Para la fecha:

- 08/08/2005 => OK
- 29/02/2005 => ERROR
- 29/02/2008 => OK
- 00/03/2004 => ERROR
- 12/13/2000 => ERROR
- 01/02/1970 => OK
- 31/06/1980 => ERROR