

EJERCICIOS SERIE 8 | RUTINAS - Ejercicios Integradores

PAUTAS GENERALES Y RECOMENDACIONES

- Sé claro y prolijo. Es muy importante que el código sea lo más claro y legible posible.
- Es muy importante que los identificadores de funciones y variables sean coherentes. El identificador debe ser suficientemente descriptivo.
- Poné una línea en blanco entre las definiciones de función para simplificar la lectura del programa.
- Las expresiones matemáticas complejas pueden representarse en varios pasos.
- Documentá correctamente las funciones y módulos que desarrolles.
- Documentá partes del código cuyo significado pudiera no quedar del todo claro.
- No documentes en exceso, pero tampoco ahorres documentación necesaria. La documentación debe ser breve y concisa.

1 EJERCICIOS INTEGRADORES

1.1 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.

1.2 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 "-".

1.3 Problema #3

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

1.4 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.

1.5 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.

1.6 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