

T
001,6425
L864

**ESCUELA SUPERIOR POLITECNICA
DEL LITORAL**

ESCUELA DE COMPUTACION

ESPOL - MATH

Elaborado por:

GISELLA LOPEZ MONCAYO

GUAYAQUIL-ECUADOR

DEDICATORIA

A MIS PADRES, GRACIAS A ELLOS HE
CULMINADO UNA VEZ MAS OTRA ETAPA MUY
IMPORTANTE EN MI VIDA

AGRADECIMIENTO

"LOS GRANDES CONOCIMIENTOS ENGENDRAN LAS GRANDES DUDAS, Y LA SOLUCION DE ELLAS SON EL DESCUBRIMIENTO DE LA VERDAD".....AGRADEZCO MUY ESPECIALMENTE A LAS SIGUIENTES PERSONAS, QUE DE UNA U OTRA MANERA ME HICIERON DUDAR MUCHAS VECES EN LA REALIZACION DE ESTE PROYECTO:

JORGE FERNANDEZ
MILTON TORRES
CARLOS PAREDES

A ELLOS, MUCHAS GRACIAS !!!!!

"NO DIO A ESCOGER LA NATURALEZA AL PADRE QUE HIJO QUISIERA TENER, NI AL HIJO QUE PADRE: PERO SI DIO A ESCOGER AL AMIGO", POR ESO MIS PROFUNDOS AGRADECIMIENTOS AL ING. XAVIER SALINAS, QUIEN ADEMAS DE ORIENTARME MUY ACERTADAMENTE EN ESTE PROYECTO, SUPO SER UN GRAN AMIGO !

DIRECTOR DE PROYECTO

A handwritten signature in black ink, appearing to read 'Xavier Salinas Andrade', is written over a horizontal dashed line.

ING. XAVIER SALINAS ANDRADE

DECLARACION EXPRESA

LA RESPONSABILIDAD POR LOS HECHOS, IDEAS Y DOCTRINAS EXPUESTOS EN ESTE PROYECTO, CORRESPONDEN EXCLUSIVAMENTE A SU AUTOR, Y EL PATRIMONIO INTELECTUAL DEL MISMO, A LA ESCUELA SUPERIOR POLITECNICA DEL LITORAL.



Handwritten signature of Gisella Lopez Moncayo, written in black ink over a horizontal dashed line. The signature is stylized, with the first letter 'G' being large and prominent, followed by 'isella', 'Lopez', and 'Moncayo' in a cursive script.

GISELLA LOPEZ MONCAYO

ESFOL - MATH

MANUAL DEL USUARIO

GISELLA LOPEZ MONCAYO

Guayaquil, Mayo de 1985

CONTENIDO

1. GENERALIDADES

1.1	ESPOL-MATH	1
1.2	LA ESCALA DE MEMORIA AUTOMATICA	1
1.3	MODO DE OPERACION	2
1.4	FUNCIONES DE UN NUMERO	2
1.5	FUNCIONES DE DOS NUMEROS	2
1.6	CALCULOS EN CADENA	3
1.7	NOTACION R P N	8
1.8	INGRESO DE NUMEROS	8
1.9	PROYECCION DE NUMEROS NEGATIVOS	9
1.10	TERMINACION DEL ESPOL-MATH	10

2. OPERACIONES GENERALES

2.1	TECLA OPERACIONAL (ENTER)	11
2.2	CONTROL DEL FORMATO DE PRESENTACION (FIXn)	13
2.3	EL USO DE PI (PI)	14
2.4	MANIPULACION DEL CONTENIDO DE LA ESCALA	
2.4.1	ASCENDER LA ESCALA (RUP)	14
2.4.2	DESCENDER LA ESCALA (RDO)	15
2.4.3	INTERCAMBIAR X CON Y (XCY)	16
2.4.4	BORRAR EL REGISTRO X (CLX)	16
2.4.5	BORRAR EL STACK (CLST)	17
2.5	CONSERVACION DEL REGISTRO X (LSTX)	17

3. FUNCIONES NUMERICAS

3.1	VALOR ABSOLUTO (ABS)	20
3.2	ENTERO (INT)	20
3.3	FRACCION (FRAC)	21
3.4	MODULO (MOD)	22
3.5	CAMBIO DE SIGNO (CHS)	23

4. OPERACIONES ARITMETICAS

4.1	ADICION (ADD +)	24
4.2	SUBTRACCION (SUB -)	26
4.3	MULTIPLICACION (MULT X)	26
4.4	DIVISION (DIV /)	27
4.5	RECIPROCO (INV 1/X)	28
4.6	ELEVACION AL CUADRADO (SQ)	28
4.7	RAIZ CUADRADA (SQRT)	29

5. OPERACIONES TRIGONOMETRICAS

5.1	SENO (SIN)	30
5.2	ARCOSENO (ASIN)	30
5.3	COSENO (COS)	30
5.4	ARCCOSENO (ACOS)	31
5.5	TANGENTE (TAN)	31
5.6	ARCOTANGENTE (ATAN)	31
5.7	CONVERSION A HORAS-MINUTOS-SEGUNDOS O GRADOS-MINUTOS-SEGUNDOS (HMS)	31
5.8	CONVERSION A HORAS DECIMALES O GRADOS DECIMALES (HR)	32
5.9	SUMA Y RESTA DE ANGULOS Y TIEMPOS (HMS+ HMS-)	32

6. OPERACIONES EXPONENCIALES Y LOGARITMICAS

6.1	EXPONENCIACION (YX)	34
6.2	LOGARITMO NATURAL (LN)	34
6.3	ANTILOGARITMO NATURAL (ALN)	34
6.4	LOGARITMO COMUN (LOG)	35
6.5	ANTILOGARITMO COMUN (ALOG)	35

7. OPERACIONES ESTADISTICAS

7.1	ACUMULACIONES (A+)	36
7.2	CORRECTOR DE ACUMULACIONES (A-)	40
7.3	MEDIA (MD)	40

8. OPERACIONES DE REGISTROS SECUNDARIOS

8.1	ALMACENAMIENTO DE NUMEROS (STOn)	42
8.2	RECUPERACION DE NUMEROS (RCLn)	43
8.3	BORRADO DE REGISTROS SECUNDARIOS (CLR)	43
8.4	ARITMETICA (STO+n STO-n STOxn STO/n)	44

9. OPERACIONES PORCENTUALES

9.1	PORCIENTO (% POR)	46
9.2	DIFERENCIA PORCENTUAL (%CH DPOR)	47

APENDICE A

FUNCIONES Y OPERACIONES LAST X	49
--------------------------------	----

APENDICE B

MENSAJES DE ERROR	50
-------------------------	----

APENDICE C

FUNCIONES DE UN NUMERO	51
FUNCIONES DE DOS NUMEROS	51

APENDICE D

INDICE DE FUNCIONES	52
---------------------------	----

1. GENERALIDADES

1.1 EL ESPOL-MATH

El ESPOL-MATH es un sistema desarrollado para proporcionar al usuario la facilidad de ejecutar operaciones típicas de las calculadoras de bolsillo, desde un terminal del computador IBM 4341.

Permite realizar una secuencia de operaciones matemáticas sin necesidad de programarlo en algún lenguaje de alto nivel.

Además el ESPOL-MATH es una ayuda didáctica para el uso de las calculadoras que trabajan en base a la notación RPN (REVERSE POLISH NOTATION). Se usan como modelos para el diseño de este sistema las calculadoras HEWLETT-PACKARD.

El ESPOL-MATH puede realizar operaciones aritméticas, estadísticas, trigonométricas, logarítmicas y exponenciales, además de tener la facilidad de manipular y alterar valores numéricos mantenidos en una escala de memoria automática (STACK) de cinco registros primarios (X, Y, Z, T y W) y en 7 registros de almacenamiento secundario (R1 a R7).

1.2 LA ESCALA DE MEMORIA AUTOMÁTICA

El almacenamiento automático de los resultados intermedios es la razón que permite al ESPOL-MATH simplificar la solución de las más complejas ecuaciones.

Los registros primarios se encuentran "apilados" como estantes, uno encima del otro, y pueden almacenar hasta cinco resultados intermedios, hasta que los necesita, insertándolos luego en los cálculos. Este sistema hace que el proceso de operar a través de los problemas, sea tan natural como si estuviera operando con lápiz y papel, pero el ESPOL-MATH efectúa la parte más pesada.

Así es como se ven los registros de la escala de memoria automática:

```
                STACK
                -----
W |             |
  |             |
-----
T |             |
  |             |
-----
Z |             |
  |             |
-----
Y |             |
  |             |
-----
X |             |
  |             |
-----
```


1.3 MODO DE OPERACION

Para poder realizar operaciones el usuario debe ingresar primero numeros al STACK y ejecutar las funciones con ellos. En este caso se necesita escribir el numero y pulsar la tecla ENTER, repitiendo este procedimiento hasta completar los operandos en la escala. Cuando se ingresa un nuevo numero, el valor del reg. X es elevado automaticamente al siguiente registro del STACK, es decir, se desplazan hacia el registro inmediato superior.

Sin embargo, la escala tambien realiza automaticamente algunos movimientos que le aportan eficiencia de calculo y facilidad de operacion, y son los que almacenan automaticamente los resultados intermedios, despues de completar un calculo, cualquier nuevo digito que ingrese sera parte de un nuevo numero. Asi mismo, la escala descendera automaticamente los numeros cuando realiza una operacion de dos numeros, es decir, bajan al registro inmediato inferior.

1.4 FUNCIONES DE UN NUMERO

Las funciones de un numero operan solamente sobre el reg. X. Los contenidos de los registros Y, Z, T y W no se ven afectados cuando se ejecuta la funcion de un numero, ya que las funciones que operan solamente sobre el reg. X no producen rotacion de la escala.

Para utilizar estas funciones:

1. Se ingresa el numero
2. Se pulsa ENTER para colocarlo en el reg. X
3. Se ingresa la funcion
4. Se pulsa ENTER para su ejecucion

Ejemplo de funciones de un solo numero:

Reciproco	INV o $1/X$
Raiz cuadrada	SQRT
Entero	INT
Fraccion	FRAC

El resto de estas funciones se encuentran descritas en el APENDICE C.

1.5 FUNCIONES DE DOS NUMEROS

A fin de que puedan efectuarse funciones de dos numeros, deben estar presentes dos cantidades. Ambos

numeros deben estar ingresados a la calculadora en los registros X y Y, por lo que estas funciones trabajan con sus valores.

Para utilizar estas funciones:

1. Se ingresa el numero
2. Se pulsa ENTER para colocarlo en el reg. X
3. Se ingresa el segundo numero
4. Se pulsa ENTER para separar el primer numero del segundo
5. Se ingresa la funcion
6. Se pulsa ENTER para su ejecucion

Ejemplo de funciones de dos numeros:

Adicion	ADD	o	+
Substraccion	SUB	o	-
Multiplicacion	MULT	o	x
Division	DIV	o	/

El resto de estas funciones se encuentran en el APENDICE C.

1.6 CALCULOS EN CADENA

La rapidez y simplicidad de operacion del sistema logico del ESPOL-MATH se evidencia durante la resolucion de calculos en cadena. Aun durante los mas largos calculos, solo se debe realizar una operacion por vez, pudiendo ver los resultados a medida que van siendo producidos.

Por ejemplo, para resolver $(17 - 5) \times 4$

Si se estuviera operando con lapiz y papel, se calcularia primero el resultado intermedio de $(17 - 5)$

$$17 - 5 = 12$$

... y luego multiplicaria dicho resultado por 4

$$12 \times 4 = 48$$

Con el ESPOL-MATH se opera con el problema de la misma forma, de una operacion por vez. Primero resuelve el resultado intermedio ...

SE INGRESA

17

ENTER

APARECE

W	:	:
T	:	:
Z	:	:
Y	:	:
X	:	17.00

SE INGRESA

5
ENTER

APARECE

```

-----
W |          |
-----
T |          |
-----
Z |          |
-----
Y | 17.00    |
-----
X |  5.00    |
-----

```

-
ENTER

```

-----
W |          |
-----
T |          |
-----
Z |          |
-----
Y |          |
-----
X | 12.00    |
-----

```

... y luego se resuelve la respuesta final. No se necesita pulsar ENTER de nuevo para almacenar el resultado intermedio, el ESPOL-MATH lo almacena automáticamente cuando se ingresa el número siguiente. Ahora se completa el problema multiplicando por 4 el resultado intermedio.

SE INGRESA

4
ENTER

APARECE

```

-----
W |          |
-----
T |          |
-----
Z |          |
-----
Y | 12.00    |
-----
X |  4.00    |
-----

```


SE INGRESA

5
ENTER

APARECE

```

-----
W |          |
-----
T |          |
-----
Z |          |
-----
Y |  6.00    |
-----
X |  5.00    |
-----

```

+
ENTER

```

-----
W |          |
-----
T |          |
-----
Z |          |
-----
Y |          |
-----
X | 11.00    |
-----

```

Ahora efectue (9 - 3)

SE INGRESA

9
ENTER

APARECE

```

-----
W |          |
-----
T |          |
-----
Z |          |
-----
Y | 11.00    |
-----
X |  9.00    |
-----

```

SE INGRESA

3
ENTER

APARECE

```

-----
W |          |
-----
T |          |
-----
Z | 11.00    |
-----
Y |  9.00    |
-----
X |  3.00    |
-----

```

-
ENTER

```

-----
W |          |
-----
T |          |
-----
Z |          |
-----
Y | 11.00    |
-----
X |  6.00    |
-----

```

Ahora se multiplican las respuestas intermedias para obtener la respuesta final.

SE INGRESA

x
ENTER

APARECE

```

-----
W |          |
-----
T |          |
-----
Z |          |
-----
Y |          |
-----
X | 66.00    |
-----

```

Se observa que no se necesita escribir ni recordar las respuestas intermedias antes de multiplicar. El ESPOL-MATH almacena automaticamente los resultados intermedios y los entrega en el momento de la multiplicacion.

1.7 NOTACION RPN

Cuando se esta utilizando el ESPOL-MATH, el sistema RPN (REVERSE POLISH NOTATION) ofrece las siguientes ventajas:

- Opera con una sola funcion por vez. El ESPOL-MATH disminuye la extension de los problemas, en lugar de hacerlos mas complejos.
- Los resultados intermedios aparecen a medida que son calculados. No hay calculos ocultos y se puede controlar cada paso a medida que se avanza.
- Los resultados intermedios son manejados automaticamente.. Cuando se resuelve un problema no es necesario escribir largas respuestas intermedias.
- Se pueden efectuar los calculos de la misma manera que lo haria con lapiz y papel. No necesita pensar el problema por adelantado.
- En los calculos no es necesario preocuparse por los parentesis, el RPN elimina la necesidad de ingresar parentesis

1.8 INGRESO DE NUMEROS

El ESPOL-MATH reserva una capacidad de hasta 15 posiciones para los numeros en las variables del STACK, LAST X y de los registros secundarios. En estas 15 ubicaciones se incluyen los digitos, el signo y el punto decimal.

El ESPOL-MATH acepta dos tipos de numeros:

- reales
- punto flotante

Cuando se trata de un numero real, los digitos deben ser escritos consecutivamente, permitiendo el ingreso de un punto seguido de los decimales, en el caso de que los tuviera, de lo contrario se puede omitir el punto si se tratara de un numero entero sin decimales. Si el numero es negativo se antepone el signo - antes de escribir cualquier digito. No se escribe el signo + para los positivos, el ESPOL-MATH lo asumira. En otras palabras, el ingreso de numeros reales se lo realiza en la misma forma tal cual lo representa.

A diferencia del anterior, para ingresar un numero de punto flotante se lo debe escribir bajo el siguiente formato:

[~] . X . [x...] E s XX

- Es opcional. Solamente se lo digita cuando el numero es negativo.

X . x...	Representa a los digitos. Solamente permite un numero entero, seguido del punto, y a su vez de los numeros punto flotante. La cantidad de estos ultimos dependen del signo del numero, es decir, si el numero es positivo se pueden escribir hasta 9 digitos, si es negativo solo hasta 8. Recuerde que solamente se cuenta con 15 posiciones.
E	Textualmente significa "Elevado a la". Representa el simbolo de exponenciacion.
s	Representa el simbolo de la potencia. Se digita + o -
XX	Representa el valor de la potencia. Necesariamente se lo debe escribir con 2 digitos.

NOTA: Los simbolos [] no se los digita. Indican solamente que su contenido es opcional.

Existen numeros que se los puede ingresar en cualquiera de los dos formatos, por ejemplo:

<u>REAL</u>	<u>PUNTO FLOTANTE</u>
948	0.948E+03
-35.42	-3.542E+01
0.0813	8.13E-02

Existen otros en cambio, que por la magnitud de la potencia no pueden ser representados en 15 posiciones como numeros reales, por ejemplo:

4.97E+52
-0.86E-34

La capacidad maxima de los valores numericos que puede soportar el ESPOL-MATH es:

$$10^{75} < x < 10^{75} \\ -10^{-75} > x > -10^{-75}$$

1.9 PROYECCION DE NUMEROS NEGATIVOS

Cuando el numero o el signo de la exponenciacion es negativo, se digita el signo -. En el momento de la proyeccion no aparecera tal simbolo sino la letra N a la derecha del STACK, para los numeros negativos, y la letra n a continuacion de la letra E, en el caso de la potencia negativa.

Ejemplos:

SE INGRESA

-3.12

ENTER

-0.123E-13

ENTER

APARECE

3.12 N

0.123En13 N

1.9 TERMINACION DEL ESPOL-MATH

Para finalizar el sistema del ESPOL-MATH, se digita solamente la palabra FIN y se pulsa la tecla ENTER.

2. OPERACIONES GENERALES

2.1 TECLA OPERACIONAL (ENTER)

Cuando se esta ingresando numeros, se debe indicar que seha finalizado el ingreso de uno y esta listo para ingresar el siguiente. Esto se hara mediante la tecla ENTER.

Ademas de permitirle saber al ESPOL-MATH que ha finalizado de ingresar un numero, pulsando ENTER tambien se desplaza el numero dentro de la escala, por lo que dicha tecla produce un ascenso de los numeros dentro de la escala.

Para ver como opera la tecla ENTER, se ingresa los numeros del 1 al 5.

SE INGRESA

1
ENTER

APARECE

```

-----
W |          |
-----
T |          |
-----
Z |          |
-----
Y |          |
-----
X | 1.00      |
-----

```

2
ENTER

```

-----
W |          |
-----
T |          |
-----
Z |          |
-----
Y | 1.00      |
-----
X | 2.00      |
-----

```

El numero en el reg. X es empujado hacia el reg. Y.

SE INGRESAAPARECE

3

ENTER

W			

T			

Z		1.00	

Y		2.00	

X		3.00	

Los numeros de X y Y son desplazados hacia los registros Y y Z respectivamente, colocandose el numero recién ingresado en el reg. X, y así sucesivamente.

SE INGRESAAPARECE

4

ENTER

W			

T		1.00	

Z		2.00	

Y		3.00	

X		4.00	

Cuando se pulsa la tecla ENTER sin haber ingresado ningún valor o alguna función, el ESPOL-MATH asume que se ha ingresado el mismo valor que contenía el reg. X, es decir, la escala asciende automáticamente duplicando el reg. X en el reg. Y.

SE PULSAAPARECE

ENTER

W		1.00	

T		2.00	

Z		3.00	

Y		4.00	

X		4.00	

<u>SE INGRESA</u>	<u>APARECE</u>
5	
ENTER	W : 2.00 :
	T : 3.00 :
	Z : 4.00 :
	Y : 4.00 :
	X : 5.00 :

Como los registros se empujan hacia arriba, una vez llenada la escala y se ingresa un nuevo numero, se observa que el valor en el reg. W se pierde por la parte superior de la escala.

2.2 CONTROL DEL FORMATO DE PRESENTACION (FIXn)

El ESPOL-MATH le proporciona la posibilidad de controlar el formato de como se verán los numeros en la presentacion.

Se debe especificar un numero, en el rango de 0 a 9, para la cantidad de posiciones decimales que se verán despues del punto. El control del formato solo altera la forma en que se presenta el numero. El numero internamente es almacenado completamente sin modificarse.

Se selecciona digitando FIX seguido por un numero que especifica la cantidad de lugares decimales.

Cuando se inicia el ESPOL-MATH sin haber ejecutado esta operacion, presentara solo 2 digitos despues del punto decimal.

Ejemplos:

<u>SE INGRESA</u>	<u>APARECE</u>
23.7589	
ENTER	23.75
FIX4	
ENTER	23.7589
FIX6	
ENTER	23.758900

Si el numero de decimales a mostrarse es mayor a la cantidad de digitos despues del punto decimal que se encuentra almacenado, el ESPOL-MATH lo completara con ceros.

2.3 EL USO DE PI

El ESPOL-MATH provee el valor de PI como una constante fija con una precision de 10 digitos (3.141592654). Ingrese simplemente PI cada vez que lo necesite en los calculos.

Por ejemplo, calcular la superficie de Ganimedes, una de las 12 lunas de Jupiter, utilizando la formula $A=\pi d^2$. El diametro de Ganimedes (d) es de 3200 millas.

<u>SE INGRESA</u>	<u>APARECE</u>
3200	
ENTER	3200.00
ENTER	3200.00
x	
ENTER	1024000.00
PI	
ENTER	3.14
x	
ENTER	32169908.78

2.4 MANIPULACION DEL CONTENIDO DE LA ESCALA

2.4.1 ASCENDER AL ESCALA (RUP)

La funcion RUP permite el desplazamiento ascendente en forma circular de los datos de la escala de memoria automatica.

Para ver como opera la funcion RUP, asumamos que tenemos cargada el STACK con los numeros del 1 al 5, de la siguiente manera:

```

-----
W | 1.00      |
-----
T | 2.00      |
-----
Z | 3.00      |
-----
Y | 4.00      |
-----
X | 5.00      |
-----

```

Ahora se ingresa la funcion RUP:

SE INGRESA

APARECE

RUP

ENTER

W	2.00
T	3.00
Z	4.00
Y	5.00
X	1.00



Cuando se ingresa RUP los contenidos de la escala suben un registro. El ultimo numero (en el reg. W) rota hacia el reg. X.

2.4.2 DESCENDER LA ESCALA (RDD)

La funcion RDD permite el desplazamiento descendente en forma circular de los datos de la escala de memoria automatica.

Para ver como opera la funcion RDD, se toma el resultado del ejemplo anterior.

W	2.00
T	3.00
Z	4.00
Y	5.00
X	1.00

SE INGRESA

APARECE

RDD

ENTER

W	1.00
T	2.00
Z	3.00
Y	4.00
X	5.00



SE INGRESAAPARECE

RDO

ENTER

W	5.00
T	1.00
Z	2.00
Y	3.00
X	4.00



Cuando se ingresa RDO los contenidos de la escala bajan un registro. El primer número (en el reg. X) rota hacia el reg. W.

2.4.3 INTERCAMBIO X CON Y (XCY)

La función XCY intercambia los contenidos de los registros X y Y, sin modificar los contenidos de los registros Z, T y W.

Si se ingresa XCY sin haber modificado los datos del ejemplo anterior, los números quedarían así:

SE INGRESAAPARECE

XCY

ENTER

W	1.00
T	2.00
Z	3.00
Y	5.00
X	4.00

2.4.4 BORRAR EL REGISTRO X (CLX)

Cuando se ejecuta la función CLX el reg. X es llevado a cero.

Continuando con el ejemplo anterior se ingresa la función CLX.

SE INGRESA

CLX

ENTER

APARECE

```

-----
W | 1.00      |
-----
T | 2.00      |
-----
Z | 3.00      |
-----
Y | 5.00      |
-----
X | 0.00      |
-----

```

2.4.5 BORRAR EL STACK (CLST)

Para borrar todos los registros del STACK (X, Y, Z, T y W) se ingresa la funcion CLST.

Consideremos los valores almacenados anteriormente para ejecutar esta funcion.

SE INGRESA

CLST

ENTER

APARECE

```

-----
W | 0.00      |
-----
T | 0.00      |
-----
Z | 0.00      |
-----
Y | 0.00      |
-----
X | 0.00      |
-----

```

2.5 CONSERVACION DEL REGISTRO X (LSTX)

Ademas de los cinco registros que almacenan automaticamente los resultados intermedios, el ESPOL-MATH cuenta tambien con un registro automatico separado, el registro LAST X.

Este registro conserva el ultimo valor que estaba en el registro X antes de ejecutar alguna funcion.

Por ejemplo:

SE INGRESAAPARECELAST X

3.4

ENTER

W	
T	
Z	
Y	
X	3.40

0.00

5.20

ENTER

W	
T	
Z	
Y	3.40
X	5.20

0.00

+

ENTER

W	
T	
Z	
Y	
X	8.60

5.20

Para llevar nuevamente el contenido del reg. LAST X al reg. X, se debe ejecutar la funcion LSTX.

SE INGRESAAPARECELAST X

LSTX

ENTER

W	
T	
Z	
Y	8.60
X	5.20

5.20

En este caso, el registro LAST X mantiene su valor anterior, se observa que no fue cambiado por 8.60.

Las funciones que actualizan al registro LAST X se encuentran descritas en el APENDICE A.

3. FUNCIONES NUMERICAS

3.1 VALOR ABSOLUTO (ABS)

Algunos calculos requieren el valor absoluto, o magnitud de un numero. Para obtener el valor absoluto de un numero que se encuentra en el registro X, ejecute la funcion ABS.

Por ejemplo, para calcular el valor absoluto de -3:

SE INGRESA

-3
ENTER

APARECE

```

-----
W |           |
-----
T |           |
-----
Z |           |
-----
Y |           |
-----
X |  3.00      | N
-----
  
```

LAST X

```

-----
| 0.00      |
-----
  
```

ABS
ENTER

```

-----
W |           |
-----
T |           |
-----
Z |           |
-----
Y |           |
-----
X |  3.00      |
-----
  
```

```

-----
| 3.00      | N
-----
  
```

3.2 ENTERO (INT)

Cuando se ejecuta la funcion INT, extrae y presenta la porcion entera del numero encontrado en el registro X truncando la parte fraccional.

Para ver como opera la funcion INT, se ingresa un numero con decimales.

SE INGRESA

1.234

ENTER

APARECE

W	
T	
Z	
Y	
X	1.234

LAST X

0.00

INT

ENTER

W	
T	
Z	
Y	
X	1.00

1.234

Cuando se ejecuta INT la parte fraccionaria del numero se pierde. El numero completo es conservado por el registro LAST X.

3.3 FRACCION (FRAC)

Cuando se ejecuta la funcion FRAC extrae y presenta la porcion fraccionaria del numero encontrado en el registro X, truncando la parte entera.

Para ver como funciona FRAC se ingresa otro numero con decimales, conservando los valores del ejemplo anterior.

SE INGRESA

56.78

ENTER

APARECE

```

-----
W |          |
-----
T |          |
-----
Z |          |
-----
Y | 1.00      |
-----
X | 56.78     |
-----

```

LAST X

```

-----
| 1.234      |
-----

```

FRAC

ENTER

```

-----
W |          |
-----
T |          |
-----
Z |          |
-----
Y | 1.00      |
-----
X | 0.78      |
-----

```

```

-----
| 56.78      |
-----

```

Cuando se ejecuta FRAC, la parte entera del numero se pierde. El numero completo es conservado por el registro LAST X.

3.4 MODULO (MOD)

Cuando se ejecuta la funcion MOD, se divide el registro Y por el registro X, y el residuo de la division es llevado al registro X.

Por ejemplo, hallar 128 modulo 10.

SE INGRESA

128

ENTER

APARECE

```

-----
W |          |
-----
T |          |
-----
Z |          |
-----
Y |          |
-----
X | 128.00    |
-----

```

LAST X

```

-----
| 0.00       |
-----

```

SE INGRESAAPARECELAST X

10

ENTER

```

-----
W |          |
-----
T |          |
-----
Z |          |
-----
Y | 128.00   |
-----
X | 10.00    |
-----

```

```

-----
| 0.00      |
-----

```

MOD

ENTER

```

-----
W |          |
-----
T |          |
-----
Z |          |
-----
Y | 128.00   |
-----
X | 8.00     |
-----

```

```

-----
| 10.00     |
-----

```

3.5 CAMBIO DE SIGNO (CHS)

Ingresando la función CHS se puede cambiar el signo de cualquier número, positivo o negativo distinto de cero, que se encuentre en el registro X.

Por ejemplo, se ingresa la función CHS tomando en cuenta los valores del ejemplo anterior:

SE INGRESAAPARECELAST X

CHS

ENTER

```

-----
W |          |
-----
T |          |
-----
Z |          |
-----
Y | 128.00   |
-----
X | 8.00     | N
-----

```

```

-----
| 10.00     |
-----

```

4. OPERACIONES ARITMETICAS

4.1 ADICION (ADD o +)

La funcion de adiccion suma el contenido del registro X con el contenido del registro Y, colocando el resultado en el registro X, esta operacion se encuentra dentro de la clasificacion de funciones de dos numeros, operando sobre los registros X y Y, lo que origina un descenso de la escala.

Para ejecutar esta operacion, se puede digitar la palabra ADD o el simbolo +. Cualquiera de las dos efectuan la funcion de adiccion.

Se realiza la operacion $12+15$, pero antes de efectuarla se pulsa nuevamente ENTER y se ingresa la funcion RDO, para poder visualizar mejor el ejemplo:

<u>SE INGRESA</u>	<u>APARECE</u>	<u>LAST X</u>
12		
ENTER	W	0.00
	T	
	Z	
	Y	
	X 12.00	
15		
ENTER	W	0.00
	T	
	Z	
	Y 12.00	
	X 15.00	
ENTER	W	0.00
	T	
	Z 12.00	
	Y 15.00	
	X 15.00	

SE INGRESAAPARECELAST X

RDO

ENTER

```

-----
W | 15.00 |
-----
T | 0.00 |
-----
Z | 0.00 |
-----
Y | 12.00 |
-----
X | 15.00 |
-----

```

```

-----
| 0.00 |
-----

```

Ahora se ejecuta la operacion suma:

SE INGRESAAPARECELAST X

+

ENTER

```

-----
W | 15.00 |
-----
T | 15.00 |
-----
Z | 0.00 |
-----
Y | 0.00 |
-----
X | 27.00 |
-----

```

```

-----
| 15.00 |
-----

```

Se observa que la escala desciende automaticamente durante los calculos que relacionan entre si a los registros X y Y, por cuanto es una operacion de dos numeros, el ultimo valor que contenia el registro X antes de realizar la funcion es conservado en LAST X.

Esta accion de descenso se produce tambien con las funciones de "-", "x" y "/". El numero en W es duplicado en W y cae en T, el numero en T cae en Z, el numero en Z cae en Y, y los numeros en Y y X se suman para dar la respuesta en X.

4.2 SUBTRACCION (SUB o -)

La funcion de substraccion resta el contenido del registro X del contenido en el registro Y, colocando el resultado en el registro X. Esta funcion es considerada de dos numeros, por lo tanto produce un descenso de la escala. Para ejecutarla se ingresa la palabra SUB o el simbolo -.

Para ejecutar esta funcion se toman los datos del ejemplo anterior, restandole el valor de 19.

<u>SE INGRESA</u>	<u>APARECE</u>	<u>LAST X</u>
19		
ENTER	W 15.00	15.00
	T 0.00	
	Z 0.00	
	Y 27.00	
	X 19.00	
-		
ENTER	W 15.00	19.00
	T 15.00	
	Z 0.00	
	Y 0.00	
	X 8.00	

El valor que anteriormente contenia el registro X es conservado por el registro LAST X.

4.3 MULTIPLICACION (MULT o x)

Esta funcion multiplica el contenido del registro X con el del registro Y, ubicando el resultado en el registro X, produciendo tambien un descenso de la escala por considerarse dentro de las funciones de dos numeros. Para su ejecucion se ingresa la palabra MULT o el simbolo x.

Se toman los valores del ejemplo anterior para multiplicarlo por 3.

SE INGRESAAPARECELAST X

3

ENTER

```

-----
W | 15.00 |
-----
T | 0.00 |
-----
Z | 0.00 |
-----
Y | 8.00 |
-----
X | 3.00 |
-----

```

```

-----
| 19.00 |
-----

```

X

ENTER

```

-----
W | 15.00 |
-----
T | 15.00 |
-----
Z | 0.00 |
-----
Y | 0.00 |
-----
X | 24.00 |
-----

```

```

-----
| 3.00 |
-----

```

Se observa que el valor del registro X es conservado por el registro LAST X antes de ejecutarse la función de multiplicación.

4.4 DIVISION (DIV o /)

Esta función divide el contenido del registro X para el contenido del registro Y, colocando el resultado en el registro X, para ejecutar se puede ingresar la palabra DIV o el símbolo /, ambas realizarán la misma operación. La función de división produce un descenso de la escala por ser considerada de dos números.

Como ejemplo se ingresa el número 2 para dividirlo con el valor del ejemplo anterior.

SE INGRESAAPARECELAST X

2

ENTER

```

-----
W | 15.00 |
-----
T | 0.00 |
-----
Z | 0.00 |
-----
Y | 24.00 |
-----
X | 2.00 |
-----

```

```

-----
| 3.00 |
-----

```

SE INGRESA

/

ENTER

APARECE

```

-----
W | 15.00 |
-----
T | 15.00 |
-----
Z | 0.00 |
-----
Y | 0.00 |
-----
X | 12.00 |
-----

```

LAST X

```

-----
| 2.00 |
-----

```

Cuando se ejecuta una funcion de dos numeros, el valor original del registro X es conservado por el registro LAST X.

4.5 RECIPROCO (INV o 1/x)

Esta funcion calcula el reciproco de un numero en el registro X. Para su operacion se ingresa la palabra INV o el simbolo 1/x.

Para ejecutar un ejemplo de esta funcion, se consideran los valores dejados por la operacion anterior.

SE INGRESA

INV

ENTER

APARECE

```

-----
W | 15.00 |
-----
T | 15.00 |
-----
Z | 0.00 |
-----
Y | 0.00 |
-----
X | 0.08 |
-----

```

LAST X

```

-----
| 12.00 |
-----

```

El resultado es colocado en el registro X, cuyo valor original es conservado en el reg. LAST X. Esta funcion es considerada de un numero.

4.6 ELEVACION AL CUADRADO (SQ)

Para elevar al cuadrado un numero en el registro X, ejecute la funcion SQ.

Se ingresa el numero 5 como valor inicial.

SE INGRESAAPARECELAST X

5

ENTER

W	
T	
Z	
Y	
X	5.00

0.00

SQ

ENTER

W	
T	
Z	
Y	
X	25.00

5.00

El resultado es colocado en el registro X, pero su contenido anterior es conservado en el registro LAST X. Esta operacion se encuentra dentro de la clasificacion de funciones de un numero.

4.7 RAIZ CUADRADA (SQRT)

Para calcular la raiz cuadrada de un numero en el registro X, ejecute la funcion SQRT.

Se toma el valor anterior para ejecutarla.

SE INGRESAAPARECELAST X

SQRT

ENTER

W	
T	
Z	
Y	
X	5.00

25.00

El resultado de esta operacion de un numero, es colocado en el registro X, conservando su valor anterior en LAST X.

5. OPERACIONES TRIGONOMETRICAS

Estas operaciones son consideradas funciones de un numero, operando solo sobre el registro X, en donde se colocara el resultado. El registro LAST X conserva el valor inicial del registro X antes de ejecutar alguna operacion trigonométrica.

5.1 SENO (SIN)

La funcion SIN calcula el seno del contenido del registro X.

<u>SE INGRESA</u>	<u>APARECE</u>	<u>LAST X</u>
467		
ENTER	467.00	0.00
SIN		
ENTER	0.95	467.00

5.2 ARCOSENO (ASIN)

La funcion ASIN calcula el arcoseno del contenido del registro X.

<u>SE INGRESA</u>	<u>APARECE</u>	<u>LAST X</u>
0.56E-08		
ENTER	0.56E+08	0.00
ASIN		
ENTER	0.32E+06	0.56E+08

5.3 COSENO (COS)

La funcion COS calcula el coseno del contenido del registro X.

<u>SE INGRESA</u>	<u>APARECE</u>	<u>LAST X</u>
38.65		
ENTER	38.65	0.00
COS		
ENTER	0.78	38.65

5.4 ARCCOSEN (ACOS)

La funcion ACOS calcula el arcocoseno del contenido del registro X.

<u>SE INGRESA</u>	<u>APARECE</u>	<u>LAST X</u>
0.52		
ENTER	0.52	0.00
ACOS		
ENTER	58.66	0.52

5.5 TANGENTE (TAN)

La funcion TAN calcula la tangente del contenido del registro X.

<u>SE INGRESA</u>	<u>APARECE</u>	<u>LAST X</u>
0.64E-43		
ENTER	0.64En43	0.00
TAN		
ENTER	0.11En44	0.64En43

5.6 ARCOTANGENTE (ATAN)

La funcion ATAN calcula el arcotangente del contenido del registro X.

<u>SE INGRESA</u>	<u>APARECE</u>	<u>LAST X</u>
0.78		
ENTER	0.78	0.00
ATAN		
ENTER	37.95	0.78

5.7 CONVERSION A HORAS-MINUTOS-SEGUNDOS (HMS)

Utilizando el ESPOL-MATH, se puede transformar un tiempo especificado en horas decimales al formato de horas-minutos-segundos mediante la ejecucion de la funcion HMS.

Para ejecutar esta funcion se ingresa la palabra HMS, y como es considerada de un numero, solo operara sobre el registro X, colocando en este ultimo el resultado. El registro LAST X mantiene el valor anterior en el momento de la operacion.

Por ejemplo, para transformar 21.57 horas en HH-MM-SS:

<u>SE INGRESA</u>	<u>APARECE</u>	<u>LAST X</u>
21.57		
ENTER	21.57	0.00
HMS		
ENTER	21.34	21.57
FIX4		
ENTER	21.3412	21.5700

Esto quiere decir 21 horas, 34 minutos y 12 segundos. Al ingresar la función de FIX4 se cambia el formato de presentación. Recuerde que la pantalla no se conmuta automáticamente para ver más de dos dígitos después de la coma decimal, a menos que se le indique. Si utiliza la función FIX con un número mayor a 4, los decimales son completados con ceros al final del número, los mismos que no deben ser tomados en cuenta.

5.8 CONVERSION A HORAS DECIMALES (HR)

Para cambiar de horas-minutos-segundos a horas decimales se ejecuta la función HR, el resultado se almacena en el registro X, conservando su valor anterior en el registro LAST X.

Cuando se ingresa un tiempo en el formato horas-minutos-segundos, los dígitos que indican las horas se encuentran a la izquierda del punto decimal, mientras que los dígitos que especifican los minutos, segundos y fracciones de segundo, se encuentran a la derecha del punto decimal.

Se conservan los valores y el FIX del ejemplo anterior.

<u>SE INGRESA</u>	<u>APARECE</u>	<u>LAST X</u>
167.221568		
ENTER	167.2215	21.5700
HR		
ENTER	167.3710	167.2215

Utilizando las funciones HMS y HR se puede convertir también ángulos especificados en grados con fracción decimal en grados-minutos-segundos y viceversa.

5.9 SUMA Y RESTA DE ANGULOS Y TIEMPOS (HMS+ y HMS-)

Para sumar o restar horas o ángulos decimales, ingrese simplemente los números que corresponden a las horas o ángulos decimales y ejecute + o -.

Para sumar horas-minutos-segundos o grados-minutos-segundos utilice la función HMS+, de lo contrario, para restarlos ejecute HMS-. Considerado esta función de dos números, opera sobre los registros X y Y, colocando el resultado en el registro X.

Por ejemplo, hallar la suma de 45 horas, 10 minutos, 50.76 segundos y 24 horas, 49 minutos, 10.95 segundos, y luego restar del resultado 7 horas, 23 minutos, 11 segundos.

<u>SE INGRESA</u>	<u>APARECE</u>	<u>LAST X</u>
45.105076		
ENTER	45.10	0.00
24.491095		
ENTER	24.49	0.00
HMS+		
ENTER	70.00	24.49

Primero se suma 45.105076 con 24.491095 en el formato de HH-MM-SS, cuyo resultado es 24.49

7.2311		
ENTER	7.23	24.49
HMS-		
ENTER	70.00	24.49

Se ingresa 7.2311 en el formato de HH-MM-SS para ser restado del resultado intermedio de 24.49

FIX6		
ENTER	62.365071	7.231100

Se ejecuta la función de FIX6 para poder visualizar el resultado de 62 horas 36 minutos y 50.71 segundos.

6. OPERACIONES EXPONENCIALES Y LOGARITMICAS

6.1 EXPONENCIACION (YX)

La funcion YX se utiliza para elevar numeros a potencias. El uso de YX permite elevar un numero real positivo o negativo a cualquier potencia real. Esto es, la potencia puede ser positiva o negativa, un entero o un numero real. El resultado es colocado en el registro X, produciendo un descenso de la escala por ser considerada funcion de dos numeros, conservando su valor anterior en el registro LAST X.

Por ejemplo:

<u>SE INGRESA</u>	<u>APARECE</u>	<u>LAST X</u>
345		
ENTER	345.00	0.00
22		
ENTER	22.00	0.00
YX		
ENTER	0.67E+56	22.00

6.2 LOGARITMO NATURAL (LN)

La funcion LN se utiliza para calcular el logaritmo natural del registro X, en base "e" (2.718281838184)

Como es considerada funcion de un numero no origina un descenso de la escala, ubicando el resultado en el registro X. El registro LAST X conserva su valor anterior.

Por ejemplo:

<u>SE INGRESA</u>	<u>APARECE</u>	<u>LAST X</u>
0.66		
ENTER	0.66	0.00
LN		
ENTER	0.41 N	0.66

6.3 ANTILOGARITMO NATURAL (ALN)

La funcion ALN eleva "e" (2.718281838184) a la potencia del valor en el registro X.

Esta funcion considerada de un numero, trabaja sobre el registro X, colocando en este ultimo el resultado. El valor inicial lo conserva LAST X

Por ejemplo:

<u>SE INGRESA</u>	<u>APARECE</u>	<u>LAST X</u>
75		
ENTER	75.00	0.00
ALN		
ENTER	0.37E+33	75.00

6.4 LOGARITMO COMUN (LOG)

La función LOG calcula el logaritmo común en base 10 del valor en el registro X. El resultado es colocado en el registro X, conservando en LAST X su valor inicial. Forma parte de las funciones de un número.

Por ejemplo:

<u>SE INGRESA</u>	<u>APARECE</u>	<u>LAST X</u>
67		
ENTER	67.00	0.00
LOG		
ENTER	1.82	67.00

6.5 ANTILOGARITMO COMUN (ALOG)

La función ALOG eleva 10 a la potencia del valor en el registro X, colocando en este último el resultado. Considerada función de un número, no produce descenso de la escala. Su valor anterior se lo encuentra en LAST X.

Por ejemplo:

<u>SE INGRESA</u>	<u>APARECE</u>	<u>LAST X</u>
33		
ENTER	33.00	0.00
ALOG		
ENTER	0.10E+34	33.00

7. OPERACIONES ESTADISTICAS

7.1 ACUMULACIONES (A+)

La ejecución de la función A+ le permite disponer, automáticamente de ciertos valores obtenidos a partir de los contenidos en los registros X y Y, para utilizarlos en problemas estadísticos.

Para esta función se utilizan únicamente seis de los siete registros de almacenamiento secundario.

Antes de comenzar cualquier cálculo utilizando la función A+, se debe borrar previamente los registros de almacenamiento utilizados en acumulaciones, pulsando la función CLR (refiérase a la sección 8.3)

Cuando se ingresa un número y se ejecuta A+, se efectúa cada una de las siguientes operaciones:

	<u>OPERACION</u>	<u>RESULTADO EN:</u>
1.	El contenido del registro X se suma al contenido de R1.	R1
2.	El cuadrado del contenido del registro X se suma al contenido de R2.	R2
3.	El contenido del registro Y se suma al contenido de R3.	R3
4.	El cuadrado del contenido del registro Y se suma al contenido de R4.	R4
5.	El contenido del registro X se multiplica por el contenido del registro Y y el producto se suma al contenido de R5.	R5
6.	Suma el valor de uno al contenido del registro R6.	R6
7.	Después que el ESPOL-MATH realiza todas las operaciones anteriores (1. a 6.), la suma de pares de valores, que se encuentran en el R6, es colocado en el registro X.	

Cuando se ejecuta A+, la escala y los contenidos de los registros de almacenamiento estadísticos cambian...

... de este esquema ...

STACK		LAST X	REGISTROS SECUNDARIOS	
W	w	0.00	R1	0.00
T	t		R2	0.00
Z	z		R3	0.00
Y	y		R4	0.00
X	x		R5	0.00
			R6	0.00
			R7	0.00

... a esta figura ...

STACK		LAST X	REGISTROS SECUNDARIOS	
W	w	x	R1	$\sum x$
T	t		R2	$\sum x^2$
Z	z		R3	$\sum y$
Y	y		R4	$\sum y^2$
X	n	El numero de pares de valores	R5	$\sum xy$
			R6	$\sum n$
			R7	0.00

Para utilizar cualquier sumatoria individualmente, en cualquier momento, se puede recuperar los contenidos de un registro de almacenamiento estadístico, llevándolos al registro X, por medio de la función RCL y la dirección del registro estadístico (se la explicara mas adelante).

Por ejemplo, hallar $\sum x$, $\sum x^2$, $\sum y$, $\sum y^2$ y $\sum xy$ para los pares de valores de X y Y que se enumeran a continuacion:

y	7	5	9
x	5	3	8

SE INGRESA	STACK	LAST X	REGISTROS SECUNDARIOS
7 ENTER	W : -----	: 0.00 -----	R1 : 0.00 -----
5 ENTER	T : -----		R2 : 0.00 -----
	Z : -----		R3 : 0.00 -----
	Y : 7.00 -----		R4 : 0.00 -----
	X : 5.00 -----		R5 : 0.00 -----
			R6 : 0.00 -----
			R7 : 0.00 -----
A+ ENTER	W : -----	: 5.00 -----	R1 : 5.00 -----
	T : -----		R2 : 25.00 -----
	Z : -----		R3 : 7.00 -----
	Y : 7.00 -----		R4 : 49.00 -----
	X : 1.00 -----		R5 : 35.00 -----
			R6 : 1.00 -----
			R7 : 0.00 -----
5 ENTER	W : -----	: 5.00 -----	R1 : 5.00 -----
3 ENTER	T : 7.00 -----		R2 : 25.00 -----
	Z : 1.00 -----		R3 : 7.00 -----
	Y : 5.00 -----		R4 : 49.00 -----
	X : 3.00 -----		R5 : 1.00 -----
			R6 : 0.00 -----
			R7 : -----

SE INGRESA	STACK	LAST X	REGISTROS SECUNDARIOS
A+ ENTER	W : 1.00 : T : 7.00 : Z : 1.00 : Y : 5.00 : X : 2.00 :	: 3.00 :	R1 : 8.00 : R2 : 34.00 : R3 : 12.00 : R4 : 74.00 : R5 : 50.00 : R6 : 2.00 : R7 : 0.00 :
9 ENTER	W : 1.00 : T : 5.00 : Z : 2.00 : Y : 9.00 : X : 8.00 :	: 3.00 :	R1 : 8.00 : R2 : 34.00 : R3 : 12.00 : R4 : 74.00 : R5 : 50.00 : R6 : 2.00 : R7 : 0.00 :
8 ENTER	W : 1.00 : T : 5.00 : Z : 2.00 : Y : 9.00 : X : 3.00 :	: 8.00 :	R1 : 16.00 : R2 : 98.00 : R3 : 21.00 : R4 : 155.00 : R5 : 122.00 : R6 : 3.00 : R7 : 0.00 :

7.2 CORRECTOR DE ACUMULACIONES (A-)

Si se descubre que uno de los valores esta equivocado, despues de haber ejecutado la funcion A+, se puede corregir las sumatorias utilizando la funcion A- (sumatoria menos) como sigue:

- Se ingresa el par de datos "incorrectos" dentro de los registros X y Y.
- Se ejecuta A- para borrar el dato incorrecto.
- Se ingresa los valores correctos para X y Y. (Si uno de los valores de un par de datos es incorrecto, se deben borrar ambos valores y reingresarlos).
- Se ejecuta A+.

7.3 MEDIA (MD)

La funcion MD se utiliza para calcular la media (promedio aritmetico) de los valores X y Y, acumulados en los registros secundarios.

Cuando se ejecuta MD:

- 1) Se calcula la media de X empleando los datos acumulados en el primero y ultimo de los registros estadisticos. (Estos son los registros que contienen Σx y n ; definidos como R1 y R6). El valor resultante de la media de X es llevado al registro X.
- 2) Se calcula la media de Y empleando los datos acumulados en el tercero y ultimo de los registros estadisticos. (Estos son los registros que contienen Σy y n ; definidos como R3 y R6). El valor resultante de la media de Y es llevado al registro Y. Para su utilizacion, ejecutando simplemente XCY puede llevar el valor de Y al registro X.

La forma mas sencilla de acumular los datos requeridos por la funcion MD, es mediante la utilizacion de la funcion A+ como se explicara anteriormente. Si se ingresa MD manteniendo los valores del ejemplo previo, se obtendria lo siguiente:

<u>SE INGRESA</u>		STACK	LAST X	REGISTROS SECUNDARIOS
MD ENTER	W	1.00	3.00	R1 16.00
	T	5.00		R2 98.00
	Z	2.00		R3 21.00
	Y	7.00		R4 155.00
	X	5.33		R5 122.00
				R6 3.00
				R7 0.00

8. OPERACIONES SOBRE REGISTROS SECUNDARIOS

8.1 ALMACENAMIENTO DE NUMEROS (STOn)

Para almacenar un número que está en el registro X en algún registro de almacenamiento secundario, ingrese la función STO seguido de un número, dentro del rango de 1 a 7, que le indicará la dirección del registro.

Por ejemplo:

SE INGRESA	STACK	LAST X	REGISTROS SECUNDARIOS
224500 ENTER	W ----- T ----- Z ----- Y ----- X 224500.00 -----	0.00 -----	R1 0.00 ----- R2 0.00 ----- R3 0.00 ----- R4 0.00 ----- R5 0.00 ----- R6 0.00 ----- R7 0.00 -----
STO4 ENTER	W ----- T ----- Z ----- Y ----- X 224500.00 -----	0.00 -----	R1 0.00 ----- R2 0.00 ----- R3 0.00 ----- R4 224500.00 ----- R5 0.00 ----- R6 0.00 ----- R7 0.00 -----

Se observa que cuando se almacena un número, este simplemente se copia en el registro; así 224500 también permanece en el registro X. El almacenamiento de un número no cambia los contenidos de la escala de memoria automática.

8.2 RECUPERACION DE NUMEROS (RCLn)

Los numeros son recuperados de los registros de almacenamiento y llevados al registro X en la misma forma en que fueron almacenados.

Para recuperar un numero de un registro de almacenamiento secundario, ingrese la funcion RCL seguido de la direccion del registro.

Por ejemplo, si mantenemos el valor almacenado por la funcion STO, al ingresar RCL nos quedaria de la siguiente manera:

SE INGRESA	STACK	LAST X	REGISTROS SECUNDARIOS
RCL4 ENTER	W	0.00	R1 0.00
	T		R2 0.00
	Z		R3 0.00
	Y 224500.00		R4 224500.00
	X 224500.00		R5 0.00
			R6 0.00
			R7 0.00

Al recuperar un numero dentro del registro X se produce el desplazamiento hacia arriba de los contenidos de la escala. Esto es, el valor previo del registro X es llevado al registro Y; el que estaba en Y al registro Z, y así sucesivamente, perdiendose el contenido del registro W.

8.3 BORRADO DE REGISTROS SECUNDARIOS (CLR)

Aun cuando haya recuperado los contenidos de un registro de almacenamiento, el numero tambien permanece en el registro secundario.

Se puede borrar los registros de almacenamiento mediante tres metodos:

- Para borrar los contenidos de un solo registro secundario, se almacena simplemente otro numero alli. El numero original es borrado y reemplazado por el nuevo.
- Para borrar un registro de almacenamiento, reemplace el numero por cero. Por ejemplo, ingrese el 0 al registro X y luego ejecute STO al registro a ser borrado.

- Para borrar todos los registros de almacenamiento a la vez, ejecute la función CLR.

8.4 ARITMETICA CON LOS REGISTROS DE ALMACENAMIENTO (STO+n, STO-n, STOxn Y STO/n)

Se puede realizar aritmetica sobre todos los registros de almacenamiento, ejecutando STO seguida por la función aritmetica, seguida a su vez por la dirección del registro (1 a 7).

<u>OPERACION</u>	<u>RESULTADO</u>
STO+1	El número en el registro X se suma a los contenidos del registro R1 y la suma es colocada dentro de R1.
STO-2	El número en el registro X se resta de los contenidos de R2 y la diferencia es colocada dentro de R2.
STOx3	El número en el registro X se multiplica por los contenidos de R3, y el producto es colocado en R3.
STO/4	El número en R4 se divide por el número en el registro X, y el cociente es colocado en R4.

Por ejemplo:

<u>SE INGRESA</u>	<u>STACK</u>	<u>LAST X</u>	<u>REGISTROS SECUNDARIOS</u>
15			
ENTER	W	0.00	R1 0.00
STO6			
ENTER	T		R2 0.00
3			
ENTER	Z		R3 0.00
	Y 15.00		R4 0.00
	X 3.00		R5 0.00
			R6 15.00
			R7 0.00

SE INGRESA	STACK	LAST X	REGISTROS SECUNDARIOS
STOx6			
ENTER	W	0.00	R1 0.00
	T		R2 0.00
	Z		R3 0.00
	Y 15.00		R4 0.00
	X 3.00		R5 0.00
			R6 45.00
			R7 0.00

9. OPERACIONES PORCENTUALES

9.1 PORCIENTO (% o POR)

Esta operacion es una funcion de dos numeros que permite calcular porcentajes, utilizando los contenidos de los registros X y Y. Para su operacion se puede ingresar la palabra POR o el simbolo %.

Para ejecutar esta funcion se debe ingresar la palabra POR o el simbolo %, cualquiera de las dos realizan la misma operacion.

Para hallar el porcentaje de un numero:

- Se digita el numero base.
- Se pulsa ENTER.
- Se digita el numero que representa la tasa por ciento.
- Se pulsa ENTER.
- Se digita la funcion %
- Se pulsa ENTER.

Por ejemplo, casi el 94% del peso de un tomate es agua. Si un tomate en particular pesa 500 gramos, que cantidad de su peso es agua ?

<u>SE INGRESA</u>	<u>APARECE</u>	<u>LAST X</u>
500		
ENTER	W	94.00
94		
ENTER	T	
%		
ENTER	Z	
	Y 500.00	
	X 470.00	

Cuando se ejecuta %, el contenido de la escala cambia

<u>... de este ...</u>	<u>... a este ...</u>
W	W
T	T
Z	Z
Y 500.00 Base	Y 500.00 Base
X 94.00 Tasa	X 470.00 Respuesta

Se observa que la respuesta calculada se sobrescribe sobre la tasa por ciento en el registro X, y el número base es conservado en el registro Y. A pesar de ser una función de dos números no produce un descenso de la escala.

Como el peso total del tomate está aun en el registro Y y el peso del agua del tomate está en el registro X, el peso del remanente se puede obtener simplemente por sustracción.

SE INGRESA

ENTER

APARECE

```

-----
W |           |
-----
T |           |
-----
Z |           |
-----
Y |           |
-----
X | 30.00      |
-----

```

LAST X

```

-----
| 470.00      |
-----

```

El peso en gramos de la materia sólida
del tomate de 500 gramos.

9.2 DIFERENCIA PORCENTUAL (%CH o DPOR)

Esta función es una operación de dos números que calcula el porcentaje de incremento o decremento del número en el registro Y con el número en el registro X.

Se puede ingresar la palabra DPOR o el símbolo %CH para su ejecución. A pesar de ser operación de dos números no se produce un descenso de la escala.

Para hallar la diferencia porcentual:

- Se digita el número base.
- Se pulsa ENTER.
- Se digita el segundo número.
- Se pulsa ENTER.
- Se digita la función %CH.
- Se pulsa ENTER.

La diferencia porcentual se calcula como $\%CH = [(x - y) * 100] / y$

Ejemplo: Un cultivador de tomates ha hallado que puede disminuir la cantidad de agua en los tomates que está cultivando. Sus tomates promedio pesan alrededor de 500 gramos y ha descubierto que solo 430 gramos de ese peso total es agua, en comparación con los 470 gramos de agua del ejemplo anterior. ¿Cuál es la diferencia porcentual de la cantidad de agua entre el tomate promedio y los indicados anteriormente?

<u>SE INGRESA</u>	<u>APARECE</u>	<u>LAST X</u>
470		
ENTER	470.00	0.00
430		
ENTER	430.00	0.00
%CH		
ENTER	8.51 N	430.00

El porcentaje de disminucion en peso de agua en los tomates del presente ejemplo es de -8.51

APENDICE AFUNCIONES Y OPERACIONES LAST X

<u>FUNCION</u>	<u>CONSERVACION X</u> <u>EN LAST X</u>	<u>FUNCION</u>	<u>CONSERVACION X</u> <u>EN LAST X</u>
ABS	si	MD	si
ADD	si	MOD	si
ACOS	si	MULT	si
ALN	si	PI	-
ALOG	si	POR	si
ASIN	si	RCLn	-
ATAN	si	RDO	-
A+	si	RUP	-
A-	si	SIN	si
CHS	-	SQ	si
CLR	-	SQRT	si
CLST	-	STOn	-
CLX	-	STOxn	-
COS	si	STO+n	-
DIV	si	STO-n	-
DPOR	si	STO/n	-
FIXn	-	SUB	si
FRAC	si	TAN	si
HMS	si	X	si
HMS+	si	XCY	-
HMS-	si	YX	si
HR	si	1/X	si
INT	si	+	si
INV	si	-	si
LN	si	/	si
LOG	si	%	si
LSTX	-	%CH	si

APENDICE BMENSAJES DE ERROR

FUERA DE RANGO

- Un numero ha excedido la capacidad de calculo o de almacenamiento del ESPOL-MATH.

(sobrepaso) > (1 x 10 ** 75)
 (1 x 10 ** -75) < (sobrepaso) > (-1 x 10 ** -75)
 (sobrepaso) < (-1 x 10 ** 75)

- La direccion del Registro Secundario no se encuentra dentro del rango de 1 a 7.

ERROR DE DATOS

El ESPOL-MATH ha intentado efectuar una operacion sin sentido

- %CH cuando Y = 0
- MD cuando R6 = 0
- STO/n cuando X = 0
- / cuando X = 0
- DIV cuando X = 0
- YX cuando Y = 0 y X <= 0 o,
Y < 0 y X no es entero
- 1/X cuando X = 0
- SQRT cuando X < 0
- LN cuando X <= 0
- ALN cuando X > 174.673 o,
X < -180.218
- LOG cuando X <= 0
- SIN cuando !X! > 2**100
- COS cuando !X! > 2**100
- TAN cuando !X! > 2**100 o,
X = (2*J+1)*90 J=1,2,3...7x10**27
- ASIN cuando !X! > 1
- ACOS cuando !X! > 1

FUNCION NO DEFINIDA

- Se ha ingresado una funcion no reconocida por el ESPOL-MATH.

INVALIDO PARA EXPONENCIACION

- Se ha ingresado un numero de punto flotante con un formato equivocado.

FIX INSUFICIENTE

- La cantidad de decimales de algun numero de la escala, LAST X o de los registros secundarios no pueden ser proyectados en 15 posiciones

APENDICE CFUNCIONES DE UN NUMERO

ABS
ACOS
ALN
ALOG
ASIN
ATAN
CHS
COS
FRAC
HMS
HR
INT
INV
LN
LOG
SIN
SQ
SQRT
TAN
1/X

FUNCIONES DE DOS NUMEROS

ADD
A+
A-
DIV
DPOR
HMS+
HMS-
MOD
MULT
POR
SUB
X
YX
+
-
/
%
%CH

APENDICE DINDICE DE FUNCIONES

		PAG.
ABS	Valor absoluto	20
ADD	Adicion	24
ACOS	Arcocoseno	31
ALN	Antilogaritmo natural	34
ALOG	Antilogaritmo comun	35
ASIN	Arcoseno	30
ATAN	Arcotangente	31
A+	Acumulaciones	36
A-	Corrector de acumulaciones	40
CHS	Cambio de signo	23
CLR	Borrado de registros	43
CLST	Borrado del STACK	17
CLX	Borrado del registro X	16
COS	Coseno	30
DIV	Division	27
DPOR	Diferencia porcentual	47
FIN	Terminacion del ESPOL-MATH	19
FIXn	Control de presentacion	13
FRAC	Parte fraccional	21
HMS	Conversion a HH.MM.SS	31
HMS+	Suma de HH.MM.SS	32
HMS-	Resta de HH.MM.SS	32
HR	Conversion a horas decimales	32
INT	Parte entera	20
INV	Reciproco	28
LN	Logaritmo Natural	34
LOG	Logaritmo comun	35
LSTX	Conservacion reg. X	17
MD	Promedio aritmetico	40
MOD	Modulo	22
MULT	Multiplicacion	26
PI	PI	14
POR	Porciento	46
RCLn	Recuperacion	43
RDO	Descenso de la escala	15
RUP	Ascenso de la escala	14
SIN	Seno	30
SQ	Elevacion al cuadrado	28
SQRT	Raiz cuadrada	29
STOn	Almacenamiento	42
STOXn	Multiplicacion	44
STO+n	Suma	44
STO-n	Resta	44
STO/n	Division	44

		PAG.
SUB	Substraccion	26
TAN	Tangente	31
X	Operador de multiplicacion	26
XCY	Intercambio X con Y	16
XY	Exponenciacion	34
1/X	Inverso	28
+	Operador suma	24
-	Operador resta	26
/	Operador de division	27
%	Operador porciento	46
%CH	Operador diferencia porcentual	47

ESPOL - MATH

MANUAL DE IMPLEMENTACION

GISELLA LOPEZ MONCAYO

Guayaquil, Mayo de 1985

CONTENIDO

1. <u>INTRODUCCION</u>	1
2. <u>DESCRIPCION DE PROGRAMAS</u>	2
2.1 MATH00	2
2.1.1 INICIALIZACION	4
2.1.2 PROYECCION DE LA PANTALLA	5
2.1.3 FINALIZACION	5
2.1.4 IDENTIFICACION Y VERIFICACION	5
2.1.5 EJECUCION	7
2.2 MATH01	8
2.3 MATH02	10
2.4 MATH03	12
2.5 MATH04	13
2.6 MATH05	15
2.7 MATH06	16
2.8 MATH07	18
2.9 MATH08	20
APENDICE A	
LISTADO DEL PROGRAMA MATH00	23
LISTADO DEL PROGRAMA MATH01	32
LISTADO DEL PROGRAMA MATH02	34
LISTADO DEL PROGRAMA MATH03	36
LISTADO DEL PROGRAMA MATH04	38
LISTADO DEL PROGRAMA MATH05	40
LISTADO DEL PROGRAMA MATH06	42
LISTADO DEL PROGRAMA MATH07	44
LISTADO DEL PROGRAMA MATH08	49
TABLA RELACIONADA	53
DISEÑO DE PANTALLA	54

1. INTRODUCCION

El ESPOL-MATH es un sistema desarrollado para proporcionar al usuario la facilidad de ejecutar operaciones típicas de las calculadoras de bolsillo, desde un terminal del computador IBM 4341.

El ESPOL-MATH puede realizar operaciones estadísticas, trigonométricas, logarítmicas y exponenciales, además de tener la facilidad de manipular y alterar valores numéricos mantenidos en una escala de memoria automática (STACK) de cinco registros primarios (X, Y, Z, T y W) y en siete registros de almacenamiento secundario.

A través de este manual se podrá aprender la manera de implementar este sistema, cuyos programas se encuentran desarrollados en lenguaje FORTRAN y EXEC II.

El sistema operativo bajo el cual está desarrollado el ESPOL-MATH es el V.M. (Virtual Machine). Este sistema operativo es recomendado para el desarrollo de programas o computación personal vía uno de sus componentes, el C.M.S. (Conversational Monitor System).

El CMS provee un sistema de computación interactivo para la resolución de problemas generales y desarrollo de programas, y sirve de base para aplicaciones interactivas, de tal manera que el ESPOL-MATH puede ser utilizado por varios usuarios al mismo tiempo.

2. DESCRIPCION DE PROGRAMAS

El sistema que proporciona al usuario la facilidad de ejecutar el ESPOL-MATH esta compuesto de 9 programas:

MATH00
MATH01
MATH02
MATH03
MATH04
MATH05
MATH06
MATH07
MATH08

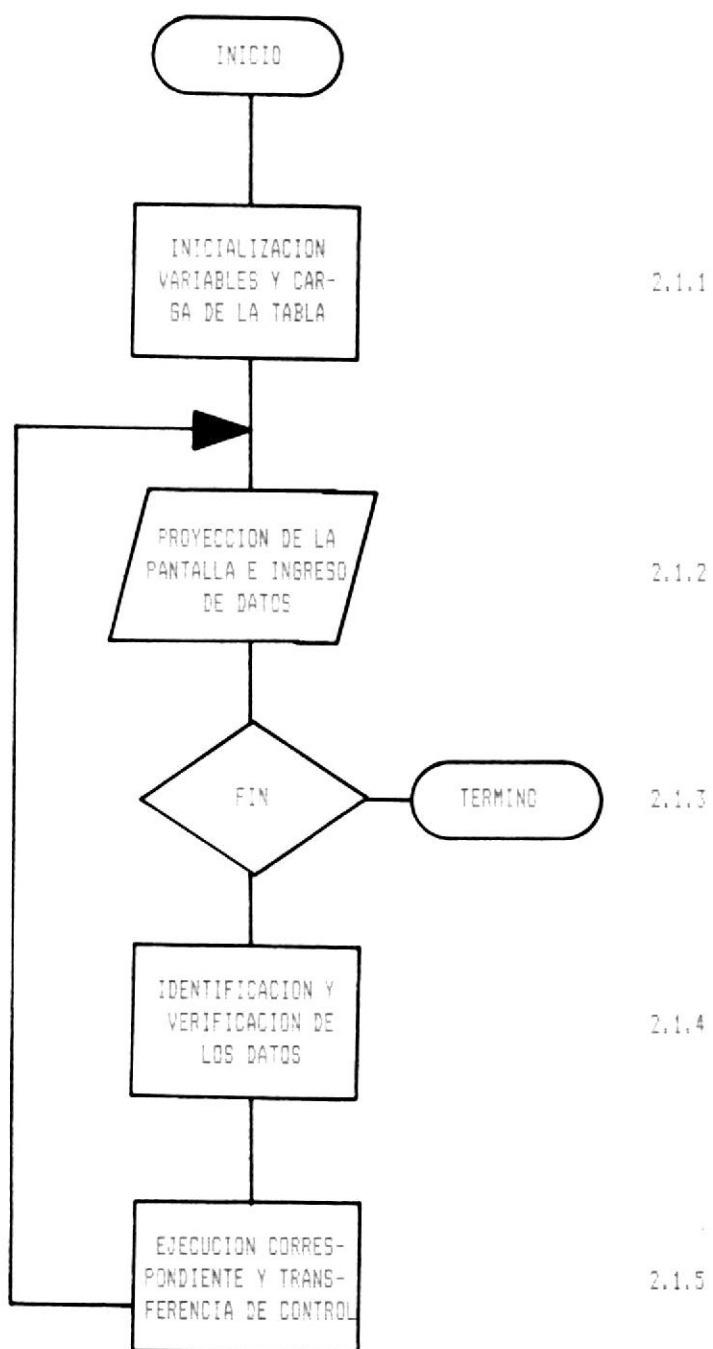
2.1 MATH00

El programa MATH00 esta desarrollado en lenguaje EXEC II. Basicamente este programa es el de mayor importancia para el ESPOL-MATH, ya que se encarga de identificar el dato ingresado por el usuario para ser procesado.

Si es un valor numerico, lo ubicara en el primer registro del STACK, que lo ira ascendiendo en la escala de acuerdo al ingreso sucesivo de numeros. Si es una funcion, el programa MATH00 transferira el control al programa que le correspondera ejecutarla.

Los registros primarios se encuentran "apilados" como estantes, uno encima del otro, y pueden almacenar hasta cinco resultados intermedios, hasta que se los necesita, insertandolos luego en los calculos.

El diagrama de flujo seguido por el programa MATH00 es el que se muestra en la siguiente pagina.



2.1.1 INICIALIZACION

El MATH00 comienza inicializando las variables del STACK, del registro LAST X, de los registros secundarios y de las variables auxiliares que utilizara el programa.

<u>VARIABLES</u>	<u>DESCRIPCION</u>
SAL1=0.00 SAL2=0.00 SAL3=0.00 SAL4=0.00 SAL5=0.00	Variables del STACK
STOR1=0.00 STOR2=0.00 STOR3=0.00 STOR4=0.00 STOR5=0.00 STOR6=0.00 STOR7=0.00	Variables de los registros secundarios
LSTX=0.00	Variable del registro LAST X
FIX=2	Variable que controla la cantidad de numeros despues del punto de- cimal
CHRIS=	Variable que almacena los datos ingresados

A continuacion carga en memoria una tabla relacionada llamada MATH\$TAB, que contendra todas las funciones reconocidas por el ESPOL-MATH, los programas correspondientes que las ejecutaran y un codigo de identificacion para el programa, ademas de indicar cuales son las funciones que actualizaran al registro LAST X.

La mencionada tabla tiene la siguiente forma:

<u>FUNCION</u>	<u>PROGRAMA</u>	<u>CODIGO</u>	<u>LAST X</u>
ABS	MATH01	1	SI
INT	MATH01	2	SI
.			
.			
.			
CLST	MATH02	5	NO
.			
.			

Cuando se ingresa una funcion, el programa MATH00 hara una busqueda en esta tabla, hasta encontrar la entrada de la funcion correspondiente. El nombre del programa que se encuentra adyacente a la funcion, es el que se debera ejecutar. Como cada programa realiza varias funciones, el CODIGO desvia el control dentro de este programa para identificar que parte del mismo debe ejecutar la funcion ingresada. El campo de la tabla correspondiente a LAST X indica si la funcion afecta el contenido del registro LAST X.

Por ejemplo, cuando se ingresa la funcion ABS, el programa MATH00 la buscara secuencialmente en la tabla hasta encontrarla, y la relacionara con el programa MATH01, proximo a ejecutarse. Como esta funcion debe actualizar el registro LAST X, el valor que contenia el ultimo registro del STACK (reg. X) es ubicado en el registro LAST X.

2.1.2 PROYECCION DE LA PANTALLA

Una vez cargada la tabla en memoria se proyectara la pantalla del ESPOL-MATH, que previamente ha sido disenada a traves del utilitario DTRPANEL que tiene el sistema operativo del computador IBM 4341, que facilita el diseno de pantalla. DTRPANEL.

En el diseno de la pantalla aparece el STACK, el registro LAST X, los registros secundarios y una zona de comandos.

Si es la primera corrida del MATH00, todas las variables del panel apareceran inicializadas a cero, de lo contrario se observaran valores numericos, ya sea ingresados anteriormente o en respuesta de un calculo efectuado. Tambien puede aparecer algun mensaje de error al intentar procesar algo equivocado.

Una vez proyectada la pantalla, el MATH00 esperara hasta que se le ingrese algun dato.

2.1.3 FINALIZACION

Despues de la proyeccion de la pantalla, el MATH00 espera hasta que algun dato sea ingresado. Lo primero que intentara detectar es el ingreso de la palabra FIN. El MATH00 hara una comparacion entre esta palabra y la variable que almacena el dato recién ingresado, si coinciden con sus contenidos, el control es desviado al final del programa en que dara por terminado el sistema, de lo contrario el dato ingresado se sometera a identificacion.

2.1.4 IDENTIFICACION Y VERIFICACION

Este proceso se encargara de detectar que tipo de dato el usuario ha ingresado, y luego de la verificacion respectiva realizara el proceso correspondiente, lo cual se indica en la siguiente tabla.

	<u>VERIFICACION</u>	<u>PROCESO CORRESPONDIENTE</u>
Si se ingresa:		
	Un blanco (se pulso solamente ENTER)	El registro X es duplicado en el registro Y, produciendo un ascenso de la escala.
	PI	El valor de 3.141592654 es asignado al registro X, produciendo un ascenso de la escala.
	LSTX	El valor que contiene el registro LAST X es llevado al registro X, produciendo un ascenso de la escala.
	Un numero	Se verifica si fue ingresado bajo el formato correcto, es decir, que los numeros deben contener solo digitos y/o con un punto decimal, y si se tratara de numeros de punto flotante con la letra E que representa el simbolo de la exponenciacion. De no ser asi se envia el mensaje : FUNCION NO DEFINIDA y se transfiere el control a la seccion 2.1.2 Si estuvo correcto el numero es colocado en el registro X produciendo un ascenso de la escala.
	FIX-	Se verifica que n este dentro del rango permitido (1 a 9). De no ser asi se emite el correspondiente mensaje de error: FIX INSUFICIENTE y se transfiere control al punto 2.1.2. Si esta correcto, se asigna n a una variable de control (&FIX) de formatos de presentacion.

Por cualquiera de las comparaciones anteriores que hayan tenido exito, se prepara la salida del panel de acuerdo al formato especificado, es decir, se proyectaran los numeros de acuerdo a la cantidad de decimales que contiene la variable de control &FIX. Luego se transfiere el control al punto 2.1.2, para proyeccion de la pantalla.

Si el dato ingresado ha pasado por las comparaciones anteriores sin tener éxito, significa que será una función que deberá ser ejecutada por algún programa.

Ciertas funciones requieren ser verificadas antes de rastrear la tabla, como por ejemplo:

<u>FUNCION INGRESADA</u>	<u>PROCESO CORRESPONDIENTE</u>
HMS o HR	Se verifica que el número almacenado en el registro X no sea punto flotante, si es así se envía el mensaje de error: ERROR DE DATOS y se transfiere el control al punto 2.1.2
STOn, STO-n, STO-n, STOxn o RCLn	Se verifica que la dirección del registro n se encuentre dentro del rango permitido (1 a 7). De no ser así se emite el respectivo mensaje: FUERA DE RANGO y se transfiere el control al punto 2.1.2

El programa MATH00 hará una búsqueda secuencial en la tabla cargada en memoria, procediendo a comparar la función ingresada con todas aquellas que se encuentran en la tabla. Si durante dicha búsqueda la función no ha sido localizada, significa que se ingresó una no reconocida por el ESPOL-MATH. Si es así se envía el mensaje: FUNCION NO DEFINIDA, y se transfiere el control a la sección 2.1.2.

2.1.5 EJECUCION

Una vez localizada la función en la tabla, se determina a cual programa se le transferirá el control, se prepara un archivo llamado ENTRADA que contendrá los parámetros dejados por el MATH00 para que el programa a ejecutarse los tome y los procese.

Los datos preparados en el archivo ENTRADA son:

- Los valores del STACK
- Los valores de los registros secundarios
- El código de identificación de la función dentro del programa

Antes de transferir el control al programa que se ejecutará, se actualiza el registro LAST X, solo en el caso que la función ingresada lo afecte.

Una vez ejecutado el programa que efectúa la función ingresada, el control retorna al MATH00 recogiendo los valores calculados, a través de un archivo SALIDA dejado por el programa, en el que contendrá los valores calculados en el STACK y en los registros secundarios.

Una vez cargado el STACK y los registros secundarios con los valores adquiridos por el archivo SALIDA, se prepara la salida del panel, de acuerdo al contenido de la variable de control de formato de presentacion (&FIX), y finalmente se transfiere el control al punto 2.1.2, para que sean proyectados los valores calculados.

2.2 MATH01

El MATH01 esta desarrollado en lenguaje Fortran, cuyo objetivo es el de ejecutar las siguientes funciones:

ABS	Valor absoluto
INT	Entero
FRAC	Fraccion
CHS	Cambio de signo
POR	Porciento
DPOR	Diferencia Porcensual
MOD	Modulo

El programa se inicia definiendo las variables que se utilizaran:

<u>NOMBRE VARIABLE</u>	<u>TIPO</u>	<u>CONTENIDO</u>
STACK1	real, doble precision	Arreglo de 5 elementos que almacena los valores del Stack, que vienen en el archivo ENTRADA
A1, A2, A3 y A4	real, doble	Contiene la capacidad maxima de los valores numericos que puede soportar el ESPOL MATH. A1 = 10**75 A2 = 10**(-75) A3 = -10**75 A4 = -10**(-75)
MSG	caracter	Se carga en esta variable mensajes emitidos por este programa
NUM	entero	Contiene el codigo de identificacion tomado en el archivo ENTRADA

A continuacion el programa lee el archivo ENTRADA creado por el MATH00, carga los valores del Stack en el arreglo llamado STACK1 y el codigo de identificacion en la variable NUM.

Luego se verifica que los valores recién tomados esten dentro del rango permitido, caso contrario se carga en la variable MSG el mensaje: FUERA DE RANGO, y se crea el archivo SALIDA (explicado mas adelante).

Como este programa realiza la operacion de varias funciones, se determina que parte de la misma se debe ejecutar para efectuar el calculo de la funcion recién ingresada. Esto se determina a traves del codigo de identificacion que se encuentra en la variable NUM, mediante el cual se desvia el control a la parte respectiva del programa.

<u>FUNCION</u>	<u>CALCULO</u>	<u>RESULTADO</u>
ABS	Calcula el valor absoluto del contenido en el quinto elemento del arreglo STACK1 (representa al registro X.)	En el quinto elemento del arreglo STACK1
INT	Presenta la porcion entera del contenido en el quinto elemento del arreglo STACK1 (representa al registro X.)	En el quinto elemento del arreglo STACK1
FRAC	Presenta la porcion fraccional del contenido en el quinto elemento del arreglo STACK1 (representa al registro X.)	En el quinto elemento del arreglo STACK1
CHS	Cambia el signo del quinto elemento del arreglo STACK1 (representa al registro X.)	En el quinto elemento del arreglo STACK1
POR	Calcula el X por ciento de Y. Utiliza el cuarto y quinto elementos del STACK1 para la formula.	En el quinto elemento del arreglo STACK1

<u>FUNCION</u>	<u>CALCULO</u>	<u>RESULTADO</u>
DPDR	Calcula la diferencia porcentual del cuarto y quinto elemento del arreglo STACK1	En el quinto elemento del arreglo STACK1
MOD	Representa el residuo de la division del quinto elemento para el cuarto elemento del arreglo STACK1	En el quinto elemento del arreglo STACK1

Una vez calculada cualquiera de las funciones anteriores, se crea un archivo llamado SALIDA, que contendra los valores del arreglo STACK1 y de la variable MSG para que sean tomados por el programa MATH00. Finalmente retorna el control al programa MATH00.

10.3 MATH02

El MATH02 esta desarrollado en lenguaje Fortran, cuyo objetivo es el de ejecutar las siguientes funciones:

RDD	Descenso de la escala
RUP	Ascenso de la escala
XYX	Intercambio X con Y
CLX	Borrado del registro X
CLST	Borrado del Stack

El programa se inicia definiendo las variables que se utilizaran:

<u>NOMBRE VARIABLE</u>	<u>TIPO</u>	<u>CONTENIDO</u>
STACK1	real, doble precision	Arreglo de 5 elementos que almacena los valores del Stack, que vienen en el archivo ENTRADA
A1, A2, A3 y A4	real, doble precision	Contiene los maximos valores numericos que puede soportar el ESPOL MATH. A1 = 10**75 A2 = 10**(-75) A3 = -10**75 A4 = -10**(-75)

MSG	caracter	Se carga en esta variable algun mensaje emitido por este programa
NUM	entero	Contiene el codigo de identificacion tomado en el archivo ENTRADA

A continuacion el programa lee el archivo ENTRADA creado por el MATH00, carga los valores del Stack en el arreglo llamado STACK1 y el codigo de identificacion en la variable NUM.

Luego se verifica que los valores recién tomados esten dentro del rango permitido, caso contrario se carga en la variable MSG el mensaje: FUERA DE RANGO, y se crea el archivo SALIDA (explicado mas adelante).

Como este programa realiza la operacion de varias funciones, se determina que parte de la misma se debe ejecutar para efectuar el calculo de la funcion recién ingresada. Esto se determina a traves del codigo de identificacion que se encuentra en la variable NUM, mediante el cual se desvia el control a la parte respectiva del programa.

<u>FUNCION</u>	<u>CALCULO</u>	<u>RESULTADO</u>
RDD	Desplaza descendente en forma circular el arreglo STACK1	STACK1(1)<=STACK1(2) STACK1(2)<=STACK1(3) STACK1(3)<=STACK1(4) STACK1(4)<=STACK1(5) STACK1(5)<=STACK1(1)
RUP	Desplaza ascendente en forma circular el arreglo STACK1	STACK1(1)<=STACK1(5) STACK1(2)<=STACK1(1) STACK1(3)<=STACK1(2) STACK1(4)<=STACK1(3) STACK1(5)<=STACK1(4)
XYC	Intercambia el contenido del cuarto elemento con el quinto del STACK1	STACK1(5)<=STACK1(4) STACK1(4)<=STACK1(5)
CLX	Encera el quinto elemento del STACK1	STACK1(5)<=0.
CLST	Encera todo el arreglo STACK1	STACK1(1)<=0 STACK1(2)<=0 STACK1(3)<=0 STACK1(4)<=0 STACK1(5)<=0

Una vez calculado cualquiera de las funciones anteriores, se crea un archivo llamado SALIDA, que contendra los valores del arreglo STACK1 y de la variable MSG para que sean tomados por el programa MATH00. Finalmente retorna el control al programa MATH00.

2.4 MATH03

El MATH03 esta desarrollado en lenguaje Fortran, cuyo objetivo es el de ejecutar las siguientes funciones:

+	Adicion
-	Substraccion
x	Multiplicacion
/	Division
YX	Exponenciacion

El programa se inicia definiendo las variables que se utilizaran:

<u>NOMBRE VARIABLE</u>	<u>TIPO</u>	<u>CONTENIDO</u>
STACK1	real, doble precision	Arreglo de 5 elementos que almacena los valores del Stack, que vienen en el archivo ENTRADA
A1, A2, A3 y A4	real, doble precision	Contiene los maximos valores numericos que puede soportar el ESPOL MATH. A1 = 10**75 A2 = 10**(-75) A3 = -10**75 A4 = -10**(-75)
MSG	caracter	Se carga en esta variable algun mensaje emitido por este programa
NUM	entero	Contiene el codigo de identificacion tomado en el archivo ENTRADA

A continuacion el programa lee el archivo ENTRADA creado por el MATH00, carga los valores del Stack en el arreglo llamado STACK1 y el codigo de identificacion en la variable NUM.

Luego se verifica que los valores recién tomados esten dentro del rango permitido, caso contrario se carga en la variable MSG el mensaje: FUERA DE RANGO, y se crea el archivo SALIDA (explicado mas adelante).

Como este programa realiza la operacion de varias funciones, se determina que parte de la misma se debe ejecutar para efectuar el calculo de la funcion recién ingresada. Esto se determina a traves del codigo de identificacion que se encuentra en la variable NUM, mediante el cual se desvia el control a la parte respectiva del programa.

<u>FUNCION</u>	<u>CALCULO</u>	<u>RESULTADO</u>
+	Suma el contenido del quinto elemento del STACK1 con el cuarto	$STACK1(5) \leftarrow STACK1(4) + STACK1(5)$
-	Resta el contenido del quinto elemento del STACK1 con el cuarto	$STACK1(5) \leftarrow STACK1(4) - STACK1(5)$
x	Multiplica el contenido del quinto elemento del STACK1 con el cuarto	$STACK1(5) \leftarrow STACK1(4) * STACK1(5)$
/	Divide el contenido del quinto elemento del STACK1 con el cuarto	$STACK1(5) \leftarrow STACK1(4) / STACK1(5)$
YX	Permite elevar el cuarto elemento como una potencia del quinto elemento del STACK1	$STACK1(5) \leftarrow STACK1(4) ** STACK1(5)$

Una vez calculado cualquiera de las funciones anteriores, se crea un archivo llamado SALIDA, que contendra los valores del arreglo STACK1 y de la variable MSG para que sean tomados por el programa MATH00. Finalmente retorna el control al programa MATH00.

2.5 MATH04

El MATH04 esta desarrollado en lenguaje Fortran, cuyo objetivo es el de ejecutar las siguientes funciones:

INV	Inverso
BQ	Cuadrado
SQRT	Raiz cuadrada

El programa se inicia definiendo las variables que se utilizaran:

<u>NOMBRE VARIABLE</u>	<u>TIPO</u>	<u>CONTENIDO</u>
STACK1	real, doble precision	Arreglo de 5 elementos que almacena los valores del Stack, que vienen en el archivo ENTRADA
A1, A2, A3 y A4	real, doble precision	Contiene los maximos valores numericos que puede soportar el ESPOL MATH. A1 = 10**75 A2 = 10**(-75) A3 = -10**75 A4 = -10**(-75)
MSG	caracter	Se carga en esta variable algun mensaje emitido por este programa
NUM	entero	Contiene el codigo de identificacion tomado en el archivo ENTRADA

A continuacion el programa lee el archivo ENTRADA creado por el MATH00, carga los valores del Stack en el arreglo llamado STACK1 y el codigo de identificacion en la variable NUM.

Luego se verifica que los valores recién tomados esten dentro del rango permitido, caso contrario se carga en la variable MSG el mensaje: FUERA DE RANGO, y se crea el archivo SALIDA (explicado mas adelante).

Como este programa realiza la operacion de varias funciones, se determina que parte de la misma se debe ejecutar para efectuar el calculo de la funcion recién ingresada. Esto se determina a traves del codigo de identificacion que se encuentra en la variable NUM, mediante el cual se desvia el control a la parte respectiva del programa.

<u>FUNCION</u>	<u>CALCULO</u>	<u>RESULTADO</u>
INV	Calcula el reciproco del quinto elemento del STACK1	STACK1(5)<=1/STACK1(5)
SQ	Eleva al cuadrado el quinto elemento del STACK1	STACK1(5)<=STACK1(5)**2
SGRT	Calcula la raiz cuadrada del quinto elemento del STACK1	STACK1(5)<=STACK1(5)**1/2

Una vez calculado cualquiera de las funciones anteriores, se crea un archivo llamado SALIDA, que contendra los valores del arreglo STACK1 y de la variable MSG para que sean tomados por el programa MATH00. Finalmente retorna el control al programa MATH00.

2.6 MATH05

El MATH05 esta desarrollado en lenguaje Fortran, cuyo objetivo es el de ejecutar las siguientes funciones:

SIN	Seno
COS	Coseno
TAN	Tangente
ASIN	Arcoseno
ACOS	Arcocoseno
ATAN	Arcotangente

El programa se inicia definiendo las variables que se utilizaran:

<u>NOMBRE VARIABLE</u>	<u>TIPO</u>	<u>CONTENIDO</u>
STACK1	real, doble precision	Arreglo de 5 elementos que almacena los valores del Stack, que vienen en el archivo ENTRADA
A1, A2, A3 y A4	real, doble precision	Contiene los maximos valores numericos que puede soportar el ESPOL MATH. A1 = 10**75 A2 = 10**(-75) A3 = -10**75 A4 = -10**(-75)
PI	real, doble precision	Contiene el valor constante de PI, 3.141592654
MSG	caracter	Se carga en esta variable algun mensaje emitido por este programa
NUM	entero	Contiene el codigo de identificacion tomado en el archivo ENTRADA

A continuacion el programa lee el archivo ENTRADA creado por el MATH00, carga los valores del Stack en el arreglo llamado STACK1 y el codigo de identificacion en la variable NUM.

Luego se verifica que los valores recién tomados esten dentro del rango permitido, caso contrario se carga en la variable MSG el mensaje: FUERA DE RANGO, y se crea el archivo SALIDA (explicado mas adelante).

Como este programa realiza la operacion de varias funciones, se determina que parte de la misma se debe ejecutar para efectuar el calculo de la funcion recién ingresada. Esto se determina a traves del codigo de identificacion que se encuentra en la variable NUM, mediante el cual se desvia el control a la parte respectiva del programa.

<u>FUNCION</u>	<u>CALCULO</u>	<u>RESULTADO</u>
SIN	Calcula el seno del quinto elemento del STACK1	STACK1(5)<=SIN(STACK1(5))
COS	Calcula el coseno del quinto elemento del STACK1	STACK1(5)<=COS(STACK1(5))
TAN	Calcula la tangente del quinto elemento del STACK1	STACK1(5)<=TAN(STACK1(5))
ASIN	Calcula el arcoseno del quinto elemento del STACK1	STACK1(5)<=ASIN(STACK1(5))
ACOS	Calcula el arcocoseno del quinto elemento del STACK1	STACK1(5)<=ACOS(STACK1(5))
SIN	Calcula el arcotangente del quinto elemento del STACK1	STACK1(5)<=ATAN(STACK1(5))

Una vez calculada cualquiera de las funciones anteriores, se crea un archivo llamado SALIDA, que contendra los valores del arreglo STACK1 y de la variable MSG para que sean tomados por el programa MATH00. Finalmente retorna el control al programa MATH00.

2.7 MATH06

El MATH05 esta desarrollado en lenguaje Fortran, cuyo objetivo es el de ejecutar las siguientes funciones:

LN	Logaritmo natural
ALN	Antilogaritmo natural
LOG	Logaritmo comun
ALOG	Antilogaritmo comun

El programa se inicia definiendo las variables que se utilizaran:

<u>NOMBRE VARIABLE</u>	<u>TIPO</u>	<u>CONTENIDO</u>
STACK1	real, doble precision	Arreglo de 5 elementos que almacena los valores del Stack, que vienen en el archivo ENTRADA
A1, A2, A3 y A4	real, doble precision	Contiene los maximos valores numericos que puede soportar el ESPOL MATH. A1 = 10^{**75} A2 = $10^{**(-75)}$ A3 = -10^{**75} A4 = $-10^{**(-75)}$
MSG	caracter	Se carga en esta variable algun mensaje emitido por este programa
NUM	entero	Contiene el codigo de identificacion tomado en el archivo ENTRADA

A continuacion el programa lee el archivo ENTRADA creado por el MATH00, carga los valores del Stack en el arreglo llamado STACK1 y el codigo de identificacion en la variable NUM.

Luego se verifica que los valores recién tomados esten dentro del rango permitido, caso contrario se carga en la variable MSG el mensaje: FUERA DE RANGO, y se crea el archivo SALIDA (explicado mas adelante).

Como este programa realiza la operacion de varias funciones, se determina que parte de la misma se debe ejecutar para efectuar el calculo de la funcion recién ingresada. Esto se determina a traves del codigo de identificacion que se encuentra en la variable NUM, mediante el cual se desvia el control a la parte respectiva del programa.

<u>FUNCION</u>	<u>CALCULO</u>	<u>RESULTADO</u>
LN	Calcula el logaritmo natural del quinto elemento del STACK1	STACK1(5)<=LN(STACK1(5))
ALN	Calcula el antilogaritmo natural del quinto elemento del STACK1	STACK1(5)<=ALN(STACK1(5))

LOG	Calcula el logaritmo comun del quinto elemento del STACK1	STACK1(5)<=LOG(STACK1(5))
ALOG	Calcula el antilogaritmo comun del quinto elemento del STACK1	STACK1(5)<=ALOG(STACK1(5))

Una vez calculada cualquiera de las funciones anteriores, se crea un archivo llamado SALIDA, que contendra los valores del arreglo STACK1 y de la variable MSG para que sean tomados por el programa MATH00. Finalmente retorna el control al programa MATH00.

2.8 MATH07

El MATH07 esta desarrollado en lenguaje Fortran, cuyo objetivo es el de ejecutar las siguientes funciones:

STOn	Almacenamiento
RCLn	Recuperacion
A+	Acumulaciones
A-	Corrector acumulaciones
MD	Promedio aritmetico
CLR	Borrado de registros
STO+n	Suma involucrando el registro n
STO-n	Resta involucrando el registro n
STO*n	Multiplificacion involucrando el reg. n
STO/n	Division involucrando el reg. n

El programa se inicia definiendo las variables que se utilizaran:

<u>NOMBRE VARIABLE</u>	<u>TIPO</u>	<u>CONTENIDO</u>
STACK1	real, doble precision	Arreglo de 5 elementos que almacena los valores del Stack, que vienen en el archivo ENTRADA
REG1	real, doble precision	Arreglo de 7 elementos que almacena los valores de los registros secundarios que vienen en el archivo ENTRADA

A1, A2, A3 y A4	real, doble precision	Contiene los maximos valores numericos que puede soportar el ESPOL MATH. A1 = 10**75 A2 = 10**(-75) A3 = -10**75 A4 = -10**(-75)
MSG	caracter	Se carga en esta variable algun mensaje emitido por este programa
NUM	entero	Contiene el codigo de identificacion tomado en el archivo ENTRADA

A continuacion el programa lee el archivo ENTRADA creado por el MATH00, carga los valores del Stack en el arreglo llamado STACK1, los valores de los registros secundarios en el arreglo llamado REG1 y el codigo de identificacion en la variable NUM.

Luego se verifica que los valores recién tomados esten dentro del rango permitido, caso contrario se carga en la variable MSG el mensaje: FUERA DE RANGO, y se crea el archivo SALIDA (explicado mas adelante).

Como este programa realiza la operacion de varias funciones, se determina que parte de la misma se debe ejecutar para efectuar el calculo de la funcion recién ingresada. Esto se determina a traves del codigo de identificacion que se encuentra en la variable NUM, mediante el cual se desvia el control a la parte respectiva del programa.

<u>FUNCION</u>	<u>CALCULO</u>	<u>RESULTADO</u>
STOn	Almacena el valor del quinto elemento del STACK1 en el elemento n del arreglo REG1	REG1(n)<=STACK1(5)
RECLn	Recupera el valor del elemento n del arreglo REG1 y lo coloca en el quinto elemento del arreglo STACK1	STACK1(5)<=REG1(n)
A+	Ofrece automaticamente diferentes sumas y productos a la vez, de los valores en el cuarto y quinto	REG1(1)<= STACK1(5) REG1(2)<= STACK1(5)^2 REG1(3)<= STACK1(4) REG1(4)<= STACK1(4)^2 REG1(5)<= STACK1(4)*STACK1(5)

	elemento del arreglo STACK1	REG1(6) <= n
<u>FUNCION</u>	<u>CALCULO</u>	<u>RESULTADO</u>
A-	Corrector del operador A+	
MD	Calcula el promedio aritmético de los valores del cuarto y quinto elemento del arreglo STACK1, acumulados en el primer y tercer elementos del arreglo REG1 respectivamente	STACK1(5) <= REG1(1)/REG1(6) STACK1(4) <= REG1(3)/REG1(6)
STD+n	Suma el quinto elemento del arreglo STACK1 con el elemento n del arreglo REG1	REG1(n) <= REG1(n) + STACK1(5)
STD-n	Resta el quinto elemento del arreglo STACK1 del elemento n del arreglo REG1	REG1(n) <= REG1(n) - STACK1(5)
STDxn	Multiplica el quinto elemento del arreglo STACK1 con el elemento n del arreglo REG1	REG1(n) <= REG1(n) * STACK1(5)
STD/n	Divide para el quinto elemento del arreglo STACK1 el elemento n del arreglo REG1	REG1(n) <= REG1(n) / STACK1(5)

Una vez calculada cualquiera de las funciones anteriores, se crea un archivo llamado SALIDA, que contendrá los valores del arreglo STACK1, los valores del arreglo REG1 y de la variable MSG para que sean tomados por el programa MATH00. Finalmente retorna el control al programa MATH00.

2.9 MATH08

El MATH08 está desarrollado en lenguaje Fortran, cuyo objetivo es el de ejecutar las siguientes funciones:

HR	Conversion a horas decimales
HMS	Conversion a HH.MM.SS.
HMS+	Suma HH.MM.SS.
HMS-	Resta HH.MM.SS.

El programa se inicia definiendo las variables que se utilizaran:

<u>NOMBRE VARIABLE</u>	<u>TIPO</u>	<u>CONTENIDO</u>
STACK1	real, doble precision	Arreglo de 5 elementos que almacena los valo- res del Stack, que vie- nen en el archivo EN- TRADA
A1, A2, A3 y A4	real, doble precision	Contiene los maximos valores numericos que puede soportar el ESPOL MATH. A1 = 10**75 A2 = 10**(-75) A3 = -10**75 A4 = -10**(-75)
MSG	caracter	Se carga en esta varia- ble algun mensaje emi- tido por este programa
NUM	entero	Contiene el codigo de identificacion tomado en el archivo ENTRADA
HR	real, doble precision	Contiene el valor con- vertido en horas deci- males, del quinto ele- mento del STACK1.
HMS	real, doble precision	Contiene el valor con- vertido en HH.MM.SS., del quinto elemento del STACK1.

A continuacion el programa lee el archivo ENTRADA creado por el MATH00, carga los valores del Stack en el arreglo llamado STACK1 y el codigo de identificacion en la variable NUM.

Luego se verifica que los valores recién tomados esten dentro del rango permitido, caso contrario se carga en la variable MSG el mensaje: FUERA DE RANGO, y se crea el archivo SALIDA (ver siguiente pagina).

Como este programa realiza la operacion de varias funciones, se determina que parte de la misma se debe ejecutar para efectuar el calculo de la funcion recién ingresada. Esto se determina a traves del codigo de identificacion que se encuentra en la variable NUM, mediante el cual se desvia el control a la parte respectiva del programa.

<u>FUNCION</u>	<u>CALCULO</u>	<u>RESULTADO</u>
HR	Convierte "HORAS MINUTOS Y SEGUNDOS" del quinto elemento del STACK1, en horas decimales	En el quinto elemento del STACK1
HMS	Convierte " HORAS DECIMALES" del quinto elemento del STACK1, en horas minutos y segundos	En el quinto elemento del STACK1
HMS+	Suma el valor del quinto elemento del STACK1 con el cuarto, en el formato de angulos u horas.	En el quinto elemento del STACK1
HMS-	Resta el quinto elemento del STACK1 del cuarto, en el formato de angulos u horas	En el quinto elemento del STACK1

Una vez calculada cualquiera de las funciones anteriores, se crea un archivo llamado SALIDA.

Este archivo es creado por el programa para que el MATH00 lo lea, y los resultados sean ubicados en las variables del STACK y REGISTROS SECUNDARIOS, para que sean proyectados en la pantalla.

En el archivo SALIDA se graban los valores del STACK1 y de la variable MSG, en caso de que en el programa se haya cargado. Finalmente retorna el control al programa MATH00.

&TRACE OFF

```
*****
*
*                                     M A T H 0 0
*
*   PROCEDIMIENTO QUE PROPORCIONA AL USUARIO LA FACILIDAD DE EJECU-
*   TAR EL 'ESPOL-MATH', CUYA FINALIDAD ES REALIZAR OPERACIONES TI-
*   PICAS DE LAS CALCULADORAS DE BOLSILLO.
*
*****
```

SET DOS OFF

DTRSERV INITIAL

&TRACE OFF

* CARGAMOS LA TABLA DE FUNCIONES EN MEMORIA

DTRSERV TEDIT MATH\$TAB

&IND = 0

&PUN = .

&FIX = 2

&CEROS = 00000000000000

* INICIALIZAMOS LAS VARIABLES INTERNAS QUE CONTENDRAN LOS VALORES
* INICIALES DEL STACK

&SAL1 = 0.000000000

&SAL2 = 0.000000000

&SAL3 = 0.000000000

&SAL4 = 0.000000000

&SAL5 = 0.000000000

*
&LSTX = 0.000000000

*
&STOR1 = 0.000000000

&STOR2 = 0.000000000

&STOR3 = 0.000000000

&STOR4 = 0.000000000

&STOR5 = 0.000000000

&STOR6 = 0.000000000

&STOR7 = 0.000000000

*
&GOTO -PREPARE

* DISPLAYA LA PANTALLA Y ACEPTA EL VALOR QUE ENTRA POR ELLA

-SAQUE

DTRSERV DISPLAY ENTRY MATH

DTRSERV SAVE ENTRY CHRIS

*
&MSG =

* VERIFICA EL CONTENIDO DE LA VARIABLE DE ENTRADA

&IF .&CHRIS = .FIN &EXIT

&IF .&CHRIS NE .PI &GOTO -SIGA

&SAL5 = 3.141592654

```

&CHRIS =
&GOTO -PREPARE
*.....
-SIGA
&IF .&CHRIS = .&BLANK &GOTO -ASCENDER
&IF .&CHRIS = .LSTX &GOTO -ASCENDER
&IF .&CHRIS NE .HMS &GOTO -PASE
&GOTO -PRUEBE
*.....
-PASE
&IF .&CHRIS NE .HR &GOTO -CONTINUE
*-----
*          SI LA FUNCION A EJECUTAR ES HR O HMS, COMPRUEBA QUE EL VALOR
*          DEL REGISTRO X ESTE DENTRO DE UN RANGO, SIN CONTENER EXPONEN-
*          CIACION
*-----
-PRUEBE
&E = &PIECE OF &SAL5 12 1
&IF .&E = .E &GOTO -RANGO
&E = &PIECE OF &SAL5 13 1
&IF .&E = .E &GOTO -RANGO
&GOTO -BUSQUE
*-----
*          IDENTIFICA SI EL DATO INGRESADO TIENE ALGUN PUNTO O SE LO HA
*          INGRESADO CON EXPONENCIACION
*-----
-CONTINUE
&PTO = &LOCATION OF &PUN &CHRIS
&LETE = &LOCATION OF E &CHRIS
*.....
&IF &LETE NE 0 &GOTO -CONE
&IF &PTO NE 0 &GOTO -CONPUNTO
*.....
&CHRIS1 = &CHRIS
&GOTO -TIPOCH
*-----
*          EXTRAER EL SIMBOLO DE EXPONENCIACION, EL SIGNO Y EL PUNTO DECI-
*          MAL PARA IDENTIFICAR AL DATO INGRESADO COMO NUMERO VALIDO
*-----
-CONE
&Z1 = &PIECE OF &CHRIS 1 1
&IF .&Z1 = .- &GOTO -NEGCONE
&IF .&PTO NE .2 &GOTO -MALEXPON
&CE1 = 3
&CE2 = 1
&CE3 = 11
&GOTO -POSCONE
*.....
-NEGCONE
&IF .&PTO NE .3 &GOTO -MALEXPON
&CE1 = 4
&CE2 = 2
&CE3 = 11
*.....
-POSCONE
&NSIG = &LETE + 1
&SIGN = &PIECE OF &CHRIS &NSIG 1
&IF .&SIGN NE .+ &GOTO -MENOS
&GOTO -CONSIGN
*.....

```

```

-MENDS
&IF .&SIGN NE .- &GOTO -NOHAY
*.....
-CONSIGN
&FIN = &LETE - &CE1
&INI = &LETE + 2
&LONG = &LENGTH OF &CHRIS
&CHRIS2 = &PIECE OF &CHRIS &CE1 &FIN
&CHRIS3 = &PIECE OF &CHRIS &INI &LONG
&CHRIS1 = &CONCAT OF &CHRIS2 &CHRIS3
&CHRIS4 = &PIECE OF &CHRIS &CE2 1
&CHRIS5 = &CONCAT OF &CHRIS4 &CHRIS1
&TIPI1 = &DATATYPE OF &CHRIS5
*.....
&IF &TIPI1 = CHAR &GOTO -NOHAY
&LNUM = &LETE - 1
&NUM1 = &PIECE OF &CHRIS 1 &LNUM
&NUM2 = &PIECE OF &CHRIS &LETE 4
*.....
&CER = &CE3 - &LNUM
&LOOP 1 &CER
&NUM1 = &CONCAT OF &NUM1 0
&CHRIS = &CONCAT OF &NUM1 &NUM2
&GOTO -ASCENDER
*-----
*          EXTRAER EL PUNTO DECIMAL DEL DATO INGRESADO PARA IDENTIFICARLO
*          COMO VALOR NUMERICO O COMO FUNCION
*-----
-CONPUNTO
&FIN = &PTO - 1
&INI = &PTO + 1
&LONG = &LENGTH OF &CHRIS
&CHRIS2 = &PIECE OF &CHRIS 1 &FIN
&CHRIS3 = &PIECE OF &CHRIS &INI &LONG
&CHRIS1 = &CONCAT OF &CHRIS2 &CHRIS3
*-----
*          VERIFICAMOS SI EL DATO ES NUMERICO O ALFABETICO
*-----
-TIPOCH
&TIPO = &DATATYPE OF &CHRIS1
&IF &TIPO = CHAR &GOTO -BUSQUE
*-----
*          ELEVA LOS NUMEROS EN LA ESCALA DE MEMORIA AUTOMATICA
*-----
-ASCENDER
&I = 1
&K = 2
&LOOP 3 4
&SAL&I = &SAL&K
&I = &I + 1
&K = &K + 1
*-----
*          SI EL DATO QUE ENTRA ES UN BLANCO EL NUMERO EN X NO VARIA,
*          SI ES LSTX, SU VALOR ES PUESTO EN X, O SI ES NUMERICO, ESTE
*          ULTIMO ES PUESTO TAMBIEN EN X
*-----
&IF .&CHRIS NE .&BLANK &GOTO -REEMPLACE
&GOTO -PREPARE
*.....
-REEMPLACE

```

```

&IF .&CHRIS NE .LSTX &GOTO -REEMPL
&SAL5 = &LSTX
&CHRIS =
&GOTO -PREPARE
*.....
-REEMPL
&SAL5 = &CHRIS
&CHRIS =
&GOTO -PREPARE
*****
*
*      CADA VEZ QUE EL DATO DE ENTRADA ES IDENTIFICADO COMO CARACTER,
*      COMPRUEBA QUE DICHA FUNCION SEA EJECUTABLE O NO
*
*      LO PRIMERO QUE HACE ES IDENTIFICAR SI LA FUNCION ES STO (STORA-
*      GE), RCL (RECALL) O FIX ANTES DE EMPEZAR A RASTREAR LA TABLA DE
*      DE FUNCIONES
*
*****
-BUSQUE
&ST = &PIECE OF &CHRIS 1 3
&IF .&ST NE .STO &GOTO -RECALL
&CODIGO = 1
&GOTO -NUMERO
*.....
-RECALL
&IF .&ST NE .RCL &GOTO -FIX
&CODIGO = 2
*-----
*      VERIFICA SI EL INDICE PARA ALMACENAR EN LOS REGISTROS SECUNDA-
*      RIOS ESTA DENTRO DEL RANGO (1-7)
*-----
-NUMERO
&IND = &PIECE OF &CHRIS 4 1
&TIPOIND = &DATATYPE OF &IND
&IF &TIPOIND = NUM &GOTO -NUMERO1
&IF .&IND NE .+ &SKIP 2
&CODIGO = 7
&GOTO -QUESERA
&IF .&IND NE .- &SKIP 2
&CODIGO = 8
&GOTO -QUESERA
&IF .&IND NE .X &SKIP 2
&CODIGO = 9
&GOTO -QUESERA
&IF .&IND NE ./ &GOTO -NOHAY
&CODIGO = 19
-QUESERA
&IND = &PIECE OF &CHRIS 5 1
&TIPOIND = &DATATYPE OF &IND
&IF &TIPOIND = CHAR &GOTO -NOHAY
-NUMERO1
&IF &IND GT 7 &GOTO -RANGO
&IF &IND = 0 &GOTO -RANGO
&PROGRAM = MATH07
&ULTIMOX = NO
&GOTO -GRABE
*-----
*      CARGA LA VARIABLE FIX PARA PREPARAR LA SALIDA DEL STACK, RE-
*      GISTROS SECUNDARIOS Y LAST X DE ACUERDO AL FORMATO ESPECIFI-

```



```

*          CADO.
*-----
-FIX
&IF .&ST NE .FIX &GOTO -TABLA
&RR = &PIECE OF &CHRIS 4 1
&TT = &DATATYPE OF &RR
&IF &TT = CHAR &GOTO -NOHAY
&FIX = &RR
&CHRIS =
&GOTO -PREPARE
*-----
*          BUSCA LA FUNCION EN LA TABLA MATH$TAB QUE RELACIONA AL PROGRAMA
*          QUE LE CORRESPONDE Y LA MANDA A EJECUTAR
*-----
-TABLA
&CLAVE = &CHRIS
DTRSERV TFETCH MATH$TAB
&IF &RETCODE NE 0 &GOTO -NOHAY
&IF &PROGRAM = MATH07 &GOTO -GRABE
*****
*
*          GRABA EN EL STACK (MEMORIA) LOS VALORES DE LA ESCALA
*
*****
DESBUF
&IF .&ULTIMOX = .NO &SKIP 1
&LSTX = &SAL5
&STACK &SAL1
&STACK &SAL2
&STACK &SAL3
&STACK &SAL4
&STACK &SAL5
&STACK &CODIGO
&STACK
*-----
*          CREA EL ARCHIVO DE ENTRADA DONDE SE ENCUENTRAN LOS VALORES DE
*          LA ESCALA PARA QUE EL PROGRAMA CORRESPONDIENTE LOS LEA, LOS
*          PROCESE Y GRABE LOS RESULTADOS EN EL ARCHIVO DE SALIDA
*-----
-EJECUTE
FILESTCK MATH ENTRADA A F 80 * *
FI 1 DISK MATH ENTRADA
FI 8 DISK MATH SALIDA
FI 6 DISK NO VALE
&PROGRAM
FINDSTAK MATH SALIDA A ALL
ERASE MATH ENTRADA A
ERASE MATH SALIDA A
ERASE NO VALE A
*.....
&CHRIS =
*-----
*          TOMA LOS VALORES DEL ARCHIVO DE SALIDA Y LOS COLOCA EN EL
*          STACK (MEMORIA)
*-----
&I = 1
&LOOP 4 5
&READ VAR &A
&CALL -REDUC &A
&SAL&I = &GLSB

```

```

&I = &I + 1
&READ VAR &MENSA
&MSG = &MENSA
*-----
*           SI LA FUNCION ACABADA DE EJECUTAR A INVOLUCRADO EL USO DE LOS
*           REGISTROS SECUNDARIOS, LOS MANDA A LEER.
*-----
&IF .&TIPOIND = .NUM &GOTO -LEA
&GOTO -PREPARE
*-----
*           GRABA EN EL ARCHIVO DE ENTRADA LOS VALORES DEL STACK Y DE LOS
*           REGISTROS SECUNDARIOS CUANDO SE TRATA DE DE UNA FUNCION STO
*           (STORAGE) O DE UN RCL (RECALL)
*-----
-GRABE
DESBUF
&IF .&ULTIMOX = .NO &SKIP 1
&LSTX = &SAL5
&STACK &SAL1
&STACK &SAL2
&STACK &SAL3
&STACK &SAL4
&STACK &SAL5
&STACK &CODIGO
&STACK &STOR1
&STACK &STOR2
&STACK &STOR3
&STACK &STOR4
&STACK &STOR5
&STACK &STOR6
&STACK &STOR7
&STACK &IND
&STACK
&TIPOIND = NUM
&GOTO -EJECUTE
*-----
*           CUANDO EL ARCHIVO DE SALIDA ES CREADO POR EL PROGRAMA MATH07
*           DEBE CONTINUAR LEYENDO VALORES PARA LOS REGISTROS SECUNDARIOS
*-----
-LEA
&TIPOIND =
&I = 1
&LOOP 4 7
&READ VAR &A
&CALL -REDUC & A
&STOR&I = &GLSB
&I = &I + 1
*****
*
*           LOS VALORES INICIALES O RESULTANTES SON CARGADOS EN LAS VARIA-
*           BLES DEL PANEL, DE ACUERDO AL FORMATO DE SALIDA REQUERIDO
*
*****
-PREPARE
*=====
*           PREPARAMOS LA SALIDA DEL STACK
*=====
&M = 1
&LOOP 6 5
&NUMSIG =

```

```

&NS&M =
&CALL -FORMATED &SAL&M &FIX &ASAL&M
&ASAL&M = &FORMA
&NS&M = &NUMSIG
&M = &M + 1
*.....
*=====
*      PREPARAMOS LA SALIDA DE LOS REGISTROS SECUNDARIOS
*=====
&M = 1
&LOOP 6 7
&NUMSIG =
&NR&M =
&CALL -FORMATED &STOR&M &FIX &ASTOR&M
&ASTOR&M = &FORMA
&NR&M = &NUMSIG
&M = &M + 1
*.....
*=====
*      PREPARAMOS LA SALIDA DE LAST X
*=====
&NUMSIG =
&NLX =
&CALL -FORMATED &LSTX &FIX &ALSTX
&ALSTX = &FORMA
&NLX = &NUMSIG
&GOTO -SAQUE
*****
*      MENSAJES DE ERROR POR ALGUN DATO MAL INGRESADO
*****
-MALEXPON
&MSG = *-INVALIDO-PARA-EXPONENCIACION-*
&GOTO -PREPARE
*.....
-NOHAY
&MSG = *--FUNCION-NO-DEFINIDA--*
&GOTO -PREPARE
*.....
-RANGO
&MSG = *---FUERA--DE--RANGO---*
&GOTO -PREPARE
&EXIT
*=====
*      FORMATEAMOS LA SALIDA DEL PANEL
*=====
-FORMATED
&APTO = &LOCATION OF . &1
&IF &APTO = 0 &GOTO -SINPTO1
&AMEN = &PIECE OF &1 1 1
&IF .&AMEN = .- &GOTO -NEGATIVO1
&AE = &PIECE OF &1 12 1
&IF .&AE = .E &GOTO -EXPONENTE1
*.....
-NORMAL1
&XX = 15 - &APTO
&IF &FIX GT &XX &GOTO -FIN1
&Y1 = &APTO + 1
&DEC = &PIECE OF &1 &Y1 *
&LDEC = &LENGTH OF &DEC
&IF &LDEC GE &FIX &GOTO -NORM1

```

&NC = &FIX - &LDEC
&NCER = &PIECE OF &CEROS 1 &NC
&FORMA = &CONCAT OF &1 &NCER
&GOTO -FINAL1

*.....

-NORM1
&ZZ = &APTO + &FIX
&FORMA = &SUBSTR OF &1 1 &ZZ
&GOTO -FINAL1

*.....

-SINPTOL
&LONGI = &LENGTH OF &1
&TOT = &LONGI + 1
&CAP = 15 - &TOT
&IF &FIX GT &CAP &GOTO -FIN1
&CR = &PIECE OF &CEROS 1 &FIX
&CPO1 = &CONCAT OF &1 .
&FORMA = &CONCAT OF &CPO1 &CR
&AMEN1 = &PIECE OF &1 1 1
&IF .&AMEN1 NE .- &GOTO -FINAL1
&NUMSIG = N
&GOTO -FINAL1

*.....

-EXPONENTE1
&VER = &PIECE OF &1 14 2
&IF .&VER = .00 &GOTO -NORMAL1
&FIN = &FIX + 2
&C1 = &PIECE OF &1 1 &FIN
&C2 = &PIECE OF &1 14 2
&C3 = &CONCAT OF EN &C2
&FORMA = &CONCAT OF &C1 &C3
&GOTO -FINAL1

*.....

-NEGATIVO1
&NUMSIG = N
&ENEG = &PIECE OF &1 12 1
&IF .&ENEG NE .E &GOTO -NORMAL1
&NVER = &PIECE OF &1 14 2
&IF .&NVER = .00 &GOTO -NORMAL1
&IF &FIX NE 9 &SKIP 2
&NFIN = &FIX + 2
&SKIP 1
&NFIN = &FIX + 3
&NC1 = &PIECE OF &1 1 &NFIN
&NC2 = &PIECE OF &1 14 2
&NC3 = &CONCAT OF EN &NC2
&FORMA = &CONCAT OF &NC1 &NC3
&GOTO -FINAL1

*.....

-FIN1
&MSG = *--FIX-INSUFICIENTE--*
&FORMA = &SAL&M

*.....

-FINAL1
&RETURN

*=====

* SUBROUTINA QUE REDUCE A 15 POSICIONES PARA QUE LOS VALORES
* CALCULADOS, QUE VIENEN EN 41 O 42 POSICIONES PUEDAN SER
* PROYECTADOS EN EL PANEL

*=====

```

-REDUC
&SIGNO = &PIECE OF &A 1 1
&IF .&SIGNO = .- &GOTO -NEGNUM
-POSNUM
&SE = &PIECE OF &A 39 1
&SB1 = &PIECE OF &A 40 2
&IF .&SE = .- &GOTO -PNEGEX
-PPOSEX
&IF .&SB1 = . &SB1 = 0
&IF &SB1 GT 9 &GOTO -PPOSEX1
&IF &SB1 NE 00 &SKIP 2
&GLSB = &PIECE OF &A 1 10
&RETURN
&LB = 3 + &SB1
&LB1 = 11 - &SB1
&LB2 = &PIECE OF &A &LB &LB1
&LB3 = &PIECE OF &A 3 &SB1
&LB4 = &CONCAT OF &LB3 .
&GLSB = &CONCAT OF &LB4 &LB2
&RETURN
-PPOSEX1
&SB2 = &PIECE OF &A 1 11
&SB3 = &PIECE OF &A 38 4
&GLSB = &CONCAT OF &SB2 &SB3
&RETURN
-PNEGEX
&CER = 0.00000000000000
&IF &SB1 GT 4 &GOTO -PNEGEX1
&GL1 = &SB1 + 2
&GL2 = &PIECE OF &CER 1 &GL1
&GL3 = 14 - &GL1
&GL4 = &PIECE OF &A 3 &GL3
&GLSB = &CONCAT OF &GL2 &GL4
&RETURN
-PNEGEX1
&SB2 = &PIECE OF &A 1 11
&SB3 = &PIECE OF &A 38 4
&GLSB = &CONCAT OF &SB2 &SB3
&RETURN
*****
-NEGNUM
&SE = &PIECE OF &A 40 1
&SB1 = &PIECE OF &A 41 2
&IF .&SE = .- &GOTO -NNEGEX
-NPOSEX
&IF &SB1 GT 9 &GOTO -NPOSEX1
&IF &SB1 NE 00 &SKIP 2
&GLSB = &PIECE OF &A 1 10
&RETURN
&LB = 4 + &SB1
&LB1 = 11 - &SB1
&LB2 = &PIECE OF &A &LB &LB1
&LB3 = &PIECE OF &A 4 &SB1
&LB4 = &CONCAT OF &LB3 .
&LB5 = &CONCAT OF &LB4 &LB2
&GLSB = &CONCAT OF - &LB5
&RETURN
-NPOSEX1
&SB2 = &PIECE OF &A 1 11
&SB3 = &PIECE OF &A 39 4

```

```
&GLSB = &CONCAT OF &SB2 &SB3
&RETURN
-NNEGEX
&CER = -0.000000000000
&IF &SB1 GT 4 &GOTO -NNEGEX1
&GL1 = &SB1 + 3
&GL2 = &PIECE OF &CER 1 &GL1
&GL3 = 13 - &GL1
&GL4 = &PIECE OF &A 4 &GL3
&GLSB = &CONCAT OF &GL2 &GL4
&RETURN
-NNEGEX1
&SB2 = &PIECE OF &A 1 11
&SB3 = &PIECE OF &A 39 4
&GLSB = &CONCAT OF &SB2 &SB3
&RETURN
```

C*****	MAT00010
C*	MAT00020
C* M A T H O 1	MAT00030
C*	MAT00040
C* OPERACIONES ABS INT FRAC CHS POR CH MOD	MAT00050
C*****	MAT00060
REAL*16 STACK1(5),UNQ,A1,A2,A3,A4	MAT00070
CHARACTER*20 MSG	MAT00080
MSG=' '	MAT00090
UNQ=1.	MAT00100
A1=10.**75	MAT00110
A2=10.**(-75)	MAT00120
A3=-10.**75	MAT00130
A4=-10.**(-75)	MAT00140
C.....	MAT00150
C SE TOMAN LOS VALORES DEL STACK QUE VIENEN	MAT00160
C EN EL ARCHIVO DE ENTRADA	MAT00170
C.....	MAT00180
DO 14 I1=1,5	MAT00190
14 READ(1,*,END=999)STACK1(I1)	MAT00200
READ(1,I1,END=999)NUM	MAT00210
11 FORMAT(I1)	MAT00220
C.....	MAT00230
C SE VERIFICA QUE LOS VALORES ESTEN DENTRO	MAT00240
C DEL RANGO PERMITIDO	MAT00250
C.....	MAT00260
IF (STACK1(5).EQ. 0.) GO TO 77	MAT00270
IF (STACK1(5).GT.A1) GO TO 21	MAT00280
IF (STACK1(5).LT.A2 .AND. STACK1(5).GT. A4) GO TO 21	MAT00290
IF (STACK1(5).LT.A3) GO TO 21	MAT00300
C	MAT00310
IF (STACK1(4).EQ. 0.) GO TO 77	MAT00320
IF (STACK1(4).GT.A1) GO TO 21	MAT00330
IF (STACK1(4).LT.A2 .AND. STACK1(4).GT. A4) GO TO 21	MAT00340
IF (STACK1(4).LT.A3) GO TO 21	MAT00350
C.....	MAT00360
C CON EL CODIGO -NUM- SE IDENTIFICA LA OPERACION QUE	MAT00370
C SE DEBE EJECUTAR	MAT00380
C.....	MAT00390
77 GO TO (1,2,3,4,5,6,7)NUM	MAT00400
C.....	MAT00410
C ABS -- SACA VALOR ABSOLUTO DEL REG. X	MAT00420
C.....	MAT00430
1 STACK1(5)=QABS(STACK1(5))	MAT00440
GO TO 9	MAT00450
C.....	MAT00460
C INT -- EXTRAER EL VALOR ENTERO TRUNCANDO LA PARTE FRACCIONARIA	MAT00470
C.....	MAT00480
2 STACK1(5)=IQINT(STACK1(5))	MAT00490
GO TO 9	MAT00500
C.....	MAT00510
C FRAC -- EXTRAER EL VALOR FRACCIONARIO TRUNCANDO LA PARTE ENTERA	MAT00520
C.....	MAT00530
3 STACK1(5)=QMOD(STACK1(5),UNQ)	MAT00540
GO TO 9	MAT00550
C.....	MAT00560
C CHS -- CAMBIA EL SIGNO EL VALOR DEL REG. X	MAT00570
C.....	MAT00580
4 STACK1(5)=STACK1(5)*(-1.)	MAT00590
GO TO 9	MAT00600

C.....		MAT00610
C	SE VERIFICA QUE LA OPERACION SEA EJECUTABLE DENTRO DEL RANGO	MAT00620
C.....		MAT00630
5	IF (((STACK1(5)*STACK1(4))/100).GT.A1) GO TO 21	MAT00640
	IF (((STACK1(5)*STACK1(4))/100).LT.A2.AND.(((STACK1(5)*STACK1(4))/	MAT00650
	1100).GT.A4) GO TO 21	MAT00660
	IF (((STACK1(5)*STACK1(4))/100).LT.A3) GO TO 21	MAT00670
C.....		MAT00680
C	(POR -- UTILIZA LA FORMULA (X*Y/100)	MAT00690
C.....		MAT00700
	STACK1(5)=(STACK1(5)*STACK1(4))/100.	MAT00710
	GO TO 9	MAT00720
C.....		MAT00730
C	SE VERIFICA QUE LA OPERACION CH SEA EJECUTABLE DENTRO DEL RANGO	MAT00740
C.....		MAT00750
6	IF (STACK1(4).EQ.0.) GO TO 22	MAT00760
	IF (((STACK1(5)-STACK1(4))*100.)/STACK1(4).GT.A1) GO TO 21	MAT00770
	IF (((((STACK1(5)-STACK1(4))*100.)/STACK1(4).LT.A2).AND.(((STACK1(	MAT00780
	15)-STACK1(4))*100.)/STACK1(4).GT.A4)) GO TO 21	MAT00790
	IF (((STACK1(5)-STACK1(4))*100.)/STACK1(4).LT.A3) GO TO 21	MAT00800
C.....		MAT00810
C	(CH DPOR -- UTILIZA LA FORMULA ((X-Y)*100)/Y	MAT00820
C.....		MAT00830
	STACK1(5)=(((STACK1(5)-STACK1(4))*100.)/STACK1(4)	MAT00840
	GO TO 9	MAT00850
C.....		MAT00860
C	SE VERIFICA QUE LA OPERACION MOD SEA EJECUTABLE DENTRO DEL RANGO	MAT00870
C.....		MAT00880
7	IF (STACK1(5).EQ.0.) GO TO 22	MAT00890
	IF (QMOD(STACK1(4),STACK1(5)).EQ.0.) GO TO 71	MAT00900
	IF (QMOD(STACK1(4),STACK1(5)).GT.A1) GO TO 21	MAT00910
	IF ((QMOD(STACK1(4),STACK1(5)).LT.A2).AND.(QMOD(STACK1(4),STACK1(	MAT00920
	15)).GT.A4)) GO TO 21	MAT00930
	IF (QMOD(STACK1(4),STACK1(5)).LT.A3) GO TO 21	MAT00940
C.....		MAT00950
C	MOD -- EXTRAER LA PARTE FRACCIONARIA DE LA DIVISION DE Y PARA X	MAT00960
C.....		MAT00970
71	STACK1(5)=QMOD(STACK1(4),STACK1(5))	MAT00980
	GO TO 9	MAT00990
C.....		MAT01000
C	MENSAJES DE ERROR	MAT01010
C.....		MAT01020
21	MSG='*--FUERA-DE-RANGO--*'	MAT01030
	GO TO 9	MAT01040
22	MSG='*--ERROR-DE-DATOS--*'	MAT01050
C.....		MAT01060
C	SE GRABAN LOS RESULTADOS DEL STACK EN EL ARCHIVO DE SALIDA	MAT01070
C		MAT01080
9	DO 89 I1=1,5	MAT01090
89	WRITE(8,*)STACK1(I1)	MAT01100
	WRITE(8,33)MSG	MAT01110
33	FORMAT(A20)	MAT01120
999	STOP	MAT01130
	END	MAT01140

C*****	MAT00010
C*	MAT00020
C* M A T H O 2	MAT00030
C*	MAT00040
C* OPERACIONES RDO RUP XCY CLX CLST	MAT00050
C*****	MAT00060
REAL*16 STACK1(5),RAUX,A1,A2,A3,A4	MAT00070
CHARACTER*20 MSG	MAT00080
RAUX=0.	MAT00090
MSG=' '	MAT00100
A1=10.**75	MAT00110
A2=10.**(-75)	MAT00120
A3=-10.**75	MAT00130
A4=-10.**(-75)	MAT00140
C.....	MAT00150
C SE TOMAN LOS VALORES DEL STACK QUE VIENEN	MAT00160
C EN EL ARCHIVO DE ENTRADA	MAT00170
C.....	MAT00180
DO 14 I1=1,5	MAT00190
14 READ(1,*,END=999)STACK1(I1)	MAT00200
READ(1,I1,END=999)NUM	MAT00210
11 FORMAT(I1)	MAT00220
C.....	MAT00230
C SE VERIFICAN QUE LOS VALORES DEL STACK ESTEN DENTRO DEL RANGO	MAT00240
C.....	MAT00250
DO 66 K=1,5	MAT00260
IF (STACK1(K).EQ.RAUX) GO TO 66	MAT00270
IF (STACK1(K).GT.A1) GO TO 21	MAT00280
IF (STACK1(K).LT.A2 .AND. STACK1(K).GT.A4) GO TO 21	MAT00290
IF (STACK1(K).LT.A3) GO TO 21	MAT00300
66 CONTINUE	MAT00310
C.....	MAT00320
C CON EL CODIGO -NUM- SE IDENTIFICA LA OPERACION QUE	MAT00330
C SE DEBE EJECUTAR	MAT00340
C.....	MAT00350
GO TO (1,2,3,4,5)NUM	MAT00360
C.....	MAT00370
C RDO -- DESPLAZAMIENTO ASCENDENTE EN FORMA CIRCULAR DE LOS VALORES	MAT00380
C DEL STACK	MAT00390
C.....	MAT00400
1 RAUX=STACK1(5)	MAT00410
DO 40 I2=5,2,-1	MAT00420
40 STACK1(I2)=STACK1(I2-1)	MAT00430
STACK1(1)=RAUX	MAT00440
GO TO 9	MAT00450
C.....	MAT00460
C RUP -- DESPLAZAMIENTO DESCENDENTE EN FORMA CIRCULAR DE LOS VALORES	MAT00470
C DEL STACK	MAT00480
C.....	MAT00490
2 RAUX=STACK1(1)	MAT00500
DO 50 I3=1,4	MAT00510
50 STACK1(I3)=STACK1(I3+1)	MAT00520
STACK1(5)=RAUX	MAT00530
GO TO 9	MAT00540
C.....	MAT00550
C XCY -- INTERCAMBIO DE VALORES ENTRE LOS REG. X Y REG. Y	MAT00560
C.....	MAT00570
3 RAUX=STACK1(5)	MAT00580
STACK1(5)=STACK1(4)	MAT00590
STACK1(4)=RAUX	MAT00600

GO TO 9	MAT00610
C.....	MAT00620
C CLX -- ENCERA REGISTRO X	MAT00630
C.....	MAT00640
4 STACK1(5)=0.	MAT00650
GO TO 9	MAT00660
C.....	MAT00670
C CLST -- ENCERA EL STACK	MAT00680
C.....	MAT00690
5 DO 60 I=1,5	MAT00700
60 STACK1(I)=0.	MAT00710
GO TO 9	MAT00720
C.....	MAT00730
C MENSAJE DE ERROR	MAT00740
C.....	MAT00750
21 MSG='*--FUERA-DE-RANGO--*'	MAT00760
C.....	MAT00770
C SE GRABAN LOS RESULTADOS DEL STACK EN EL ARCHIVO DE SALIDA	MAT00780
C.....	MAT00790
9 DO 89 I1=1,5	MAT00800
89 WRITE(8,*)STACK1(I1)	MAT00810
WRITE(8,33)MSG	MAT00820
33 FORMAT(A20)	MAT00830
999 STOP	MAT00840
END	MAT00850

C*****	MAT00010
C*	MAT00020
C* M A T H 0 3	MAT00030
C*	MAT00040
C* OPERACIONES + - X / YX	MAT00050
C*****	MAT00060
REAL*16 STACK1(5),A1,A2,A3,A4,SB,UNO	MAT00070
CHARACTER*20 MSG	MAT00080
MSG=' '	MAT00090
UNO=1.	MAT00100
A1=10.**75	MAT00110
A2=10.**(-75)	MAT00120
A3=-10.**75	MAT00130
A4=-10.**(-75)	MAT00140
C.....	MAT00150
C SE TOMAN LOS VALORES DEL STACK QUE VIENEN	MAT00160
C EN EL ARCHIVO DE ENTRADA	MAT00170
C.....	MAT00180
DO 14 I1=1,5	MAT00190
14 READ(1,*,END=999)STACK1(I1)	MAT00200
READ(1,11,END=999)NUM	MAT00210
11 FORMAT(I1)	MAT00220
C.....	MAT00230
C CON EL CODIGO -NUM- SE IDENTIFICA LA OPERACION QUE	MAT00240
C SE DEBE EJECUTAR	MAT00250
C.....	MAT00260
GO TO (10,20,30,40,45)NUM	MAT00270
C.....	MAT00280
C SE VERIFICA QUE LA OPERACION + SE EJECUTE DENTRO DEL RANGO	MAT00290
C.....	MAT00300
10 IF (STACK1(5)+STACK1(4).EQ.0.) GO TO 71	MAT00310
IF (STACK1(5)+STACK1(4).GT.A1) GO TO 21	MAT00320
IF (STACK1(5)+STACK1(4).LT.A2.AND.STACK1(5)+STACK1(4).GT.A4) GO T	MAT00330
10 21	MAT00340
IF (STACK1(5)+STACK1(4).LT.A3) GO TO 21	MAT00350
C.....	MAT00360
C + ADD -- SUMA LOS REGISTROS X - Y	MAT00370
C.....	MAT00380
71 STACK1(5)=STACK1(5)+STACK1(4)	MAT00390
GO TO 50	MAT00400
C.....	MAT00410
C SE VERIFICA QUE LA OPERACION - SE EJECUTE DENTRO DEL RANGO	MAT00420
C.....	MAT00430
20 IF (STACK1(4)-STACK1(5).EQ.0.) GO TO 28	MAT00440
IF (STACK1(4)-STACK1(5).GT.A1) GO TO 21	MAT00450
IF (STACK1(4)-STACK1(5).LT.A2.AND.STACK1(4)-STACK1(5).GT.A4) GO T	MAT00460
10 21	MAT00470
IF (STACK1(4)-STACK1(5).LT.A3) GO TO 21	MAT00480
C.....	MAT00490
C - SUB -- RESTA EL REGISTRO Y DE X	MAT00500
C.....	MAT00510
28 STACK1(5)=STACK1(4)-STACK1(5)	MAT00520
GO TO 50	MAT00530
C.....	MAT00540
C SE VERIFICA QUE LA OPERACION X SE EJECUTE DENTRO DEL RANGO	MAT00550
C.....	MAT00560
30 IF (STACK1(4)*STACK1(5).EQ.0.) GO TO 31	MAT00570
IF (STACK1(4)*STACK1(5).GT.A1) GO TO 21	MAT00580
IF (STACK1(4)*STACK1(5).LT.A2.AND.STACK1(4)*STACK1(5).GT.A4) GO T	MAT00590
10 21	MAT00600

	IF (STACK1(4)*STACK1(5).LT.A3) GO TO 21	MAT00610
C.....		MAT00620
C	X MULT -- MULTIPLICA LOS REGISTROS X - Y	MAT00630
C.....		MAT00640
31	STACK1(5)=STACK1(5)*STACK1(4)	MAT00650
	GO TO 50	MAT00660
C.....		MAT00670
C	SE VERIFICA QUE LA OPERACION / SE EJECUTE DENTRO DEL RANGO	MAT00680
C.....		MAT00690
40	IF (STACK1(5).EQ.0.) GO TO 22	MAT00700
	IF (STACK1(4)/STACK1(5).EQ.0.) GO TO 41	MAT00710
	IF (STACK1(4)/STACK1(5).GT.A1) GO TO 21	MAT00720
	IF (STACK1(4)/STACK1(5).LT.A2.AND.STACK1(4)/STACK1(5).GT.A4) GO T	MAT00730
	10 21	MAT00740
	IF (STACK1(4)/STACK1(5).LT.A3) GO TO 21	MAT00750
C.....		MAT00760
C	/ DIV -- DIVIDE EL REGISTRO X PARA Y	MAT00770
C.....		MAT00780
41	STACK1(5)=STACK1(4)/STACK1(5)	MAT00790
	GO TO 50	MAT00800
C.....		MAT00810
C	SE VERIFICA QUE LA OPERACION 'YX' SE EJECUTE DENTRO DEL RANGO	MAT00820
C.....		MAT00830
45	IF (STACK1(4).EQ.0. .AND. STACK1(5).LE.0.) GO TO 22	MAT00840
	SB=QMOD(STACK1(5),UNO)	MAT00850
	IF (STACK1(4).EQ.0. .AND. SB.NE.0.) GO TO 22	MAT00860
	IF (STACK1(4)**STACK1(5).EQ.0.) GO TO 46	MAT00870
	IF (STACK1(4)**STACK1(5).GT.A1) GO TO 21	MAT00880
	IF (STACK1(4)**STACK1(5).LT.A2.AND.STACK1(4)**STACK1(5).GT.A4) GO	MAT00890
	1 TO 21	MAT00900
	IF (STACK1(4)**STACK1(5).LT.A3) GO TO 21	MAT00910
C.....		MAT00920
C	YX -- ELEVA Y A LA POTENCIA DE X	MAT00930
C.....		MAT00940
46	STACK1(5)=STACK1(4)**STACK1(5)	MAT00950
C.....		MAT00960
C	SE REALIZA LA OPERACION DE DESCENSO DEL STACK	MAT00970
C.....		MAT00980
50	DO 60 I=4,2,-1	MAT00990
60	STACK1(I)=STACK1(I-1)	MAT01000
	GO TO 90	MAT01010
C.....		MAT01020
C	MENSAJES DE ERROR	MAT01030
C.....		MAT01040
21	MSG='**-FUERA-DE-RANGO-**'	MAT01050
	GO TO 90	MAT01060
22	MSG='**-ERROR-DE-DATOS-**'	MAT01070
C.....		MAT01080
C	SE GRABAN LOS RESULTADOS DEL STACK EN EL ARCHIVO DE SALIDA	MAT01090
C.....		MAT01100
90	DO 89 I1=1,5	MAT01110
89	WRITE(8,*)STACK1(I1)	MAT01120
	WRITE(8,33)MSG	MAT01130
33	FORMAT(A20)	MAT01140
999	STOP	MAT01150
	END	MAT01160

C*****	MAT00010
C*	MAT00020
C* M A T H O 4	MAT00030
C*	MAT00040
C* OPERACIONES INV SQ SQRT	MAT00050
C*****	MAT00060
REAL*16 STACK1(5),R1,A1,A2,A3,A4	MAT00070
CHARACTER*20 MSG	MAT00080
MSG=' '	MAT00090
R1=(10.**37)*3.1623	MAT00100
A1=10.**75	MAT00110
A2=10.**(-75)	MAT00120
A3=-10.**75	MAT00130
A4=-10.**(-75)	MAT00140
C.....	MAT00150
C SE TOMAN LOS VALORES DEL STACK QUE VIENEN	MAT00160
C EN EL ARCHIVO DE ENTRADA	MAT00170
C.....	MAT00180
DO 14 I1=1,5	MAT00190
14 READ(1,*,END=999)STACK1(I1)	MAT00200
READ(1,I1,END=999)NUM	MAT00210
11 FORMAT(I1)	MAT00220
C.....	MAT00230
C SE VERIFICA QUE LOS DATOS ESTEN DENTRO DEL RANGO	MAT00240
C.....	MAT00250
IF (STACK1(5).EQ.0.) GO TO 777	MAT00260
IF (STACK1(5).GT.A1) GO TO 21	MAT00270
IF (STACK1(5).LT.A2.AND.STACK1(5).GT.A4) GO TO 21	MAT00280
IF (STACK1(5).LT.A3) GO TO 21	MAT00290
C.....	MAT00300
C CON EL CODIGO -NUM- SE IDENTIFICA LA OPERACION QUE	MAT00310
C SE DEBE EJECUTAR	MAT00320
C.....	MAT00330
777 GO TO (10,20,30)NUM	MAT00340
C.....	MAT00350
C INV 1/X -- CALCULA EL INVERSO DEL REGISTRO X	MAT00360
C.....	MAT00370
10 IF (STACK1(5).EQ.0.) GO TO 22	MAT00380
STACK1(5)=1./STACK1(5)	MAT00390
GO TO 50	MAT00400
C.....	MAT00410
C SQ -- CALCULA LA POTENCIA CUADRADA DEL REGISTRO X	MAT00420
C.....	MAT00430
20 IF(STACK1(5).GE.R1) GO TO 21	MAT00440
STACK1(5)=STACK1(5)**2.	MAT00450
GO TO 50	MAT00460
C.....	MAT00470
C SQRT -- CALCULA LA RAIS CUADRADA DEL REGISTRO X	MAT00480
C.....	MAT00490
30 IF(STACK1(5).LT.0.) GO TO 22	MAT00500
STACK1(5)=QSQRT(STACK1(5))	MAT00510
GO TO 50	MAT00520
C.....	MAT00530
C MENSAJES DE ERROR	MAT00540
C.....	MAT00550
21 MSG='*--FUERA-DE-RANGO--*'	MAT00560
GO TO 50	MAT00570
22 MSG='*--ERROR-DE-DATOS--*'	MAT00580
C.....	MAT00590
C SE GRABAN LOS RESULTADOS DEL STACK EN EL ARCHIVO DE SALIDA	MAT00600

C.....		MAT00610
50	DO 89 I1=1,5	MAT00620
89	WRITE(8,*)STACK1(I1)	MAT00630
	WRITE(8,33)MSG	MAT00640
33	FORMAT(A20)	MAT00650
999	STOP	MAT00660
	END	MAT00670

C*****		MAT00010
C*		MAT00020
C*	M A T H O 5	MAT00030
C*		MAT00040
C* OPERACIONES SIN COS TAN ASIN ACOS ATAN		MAT00050
C*****		MAT00060
REAL*16 STACK1(5),CONST1,PI,R1,R3,ABSOL,FIN,A1,A2,A3,A4		MAT00070
CHARACTER*20 MSG		MAT00080
CONST1=180.		MAT00090
PI=3.141592654		MAT00100
MSG=' '		MAT00110
R1=2.**100		MAT00120
FIN=7.*(10.**27)		MAT00130
A1=10.**75		MAT00140
A2=10.**(-75)		MAT00150
A3=-10.**75		MAT00160
A4=-10.**(-75)		MAT00170
C.....		MAT00180
C SE TOMAN LOS VALORES DEL STACK QUE VIENEN		MAT00190
C EN EL ARCHIVO DE ENTRADA		MAT00200
C.....		MAT00210
DO 14 I1=1,5		MAT00220
14 READ(1,*,END=999)STACK1(I1)		MAT00230
READ(1,I1,END=999)NUM		MAT00240
11 FORMAT(I1)		MAT00250
C.....		MAT00260
C SE VERIFICA QUE LOS DATOS ESTEN DENTRO DEL RANGO		MAT00270
C.....		MAT00280
IF (STACK1(5).EQ.0.) GO TO 777		MAT00290
IF (STACK1(5)**1.GT.A1) GO TO 21		MAT00300
IF (STACK1(5)**1.LT.A2.AND.STACK1(5)**1.GT.A4) GO TO 21		MAT00310
IF (STACK1(5)**1.LT.A3) GO TO 21		MAT00320
ABSOL=QABS(STACK1(5))		MAT00330
C.....		MAT00340
C CON EL CODIGO -NUM- SE IDENTIFICA LA OPERACION QUE		MAT00350
C SE DEBE EJECUTAR		MAT00360
C.....		MAT00370
777 GO TO (10,20,30,40,50,60)NUM		MAT00380
C.....		MAT00390
C SIN -- CALCULA EL SENO DEL REGISTRO X		MAT00400
C.....		MAT00410
10 IF(ABSOL.GT.R1) GO TO 22		MAT00420
STACK1(5)=QSIN(STACK1(5)*PI/CONST1)		MAT00430
GO TO 90		MAT00440
C.....		MAT00450
C COS -- CALCULA EL COSENO DEL REGISTRO X		MAT00460
C.....		MAT00470
20 IF(ABSOL.GT.R1) GO TO 22		MAT00480
STACK1(5)=QCOS(STACK1(5)*PI/CONST1)		MAT00490
GO TO 90		MAT00500
C.....		MAT00510
C TAN -- CALCULA LA TANGENTE DEL REGISTRO X		MAT00520
C.....		MAT00530
30 IF(ABSOL.GT.R1) GO TO 22		MAT00540
DO 34 J=0,FIN		MAT00550
R3=((2*J+1)*90)		MAT00560
34 IF (R3.EQ.STACK1(5)) GO TO 22		MAT00570
STACK1(5)=QTAN(STACK1(5)*PI/CONST1)		MAT00580
GO TO 90		MAT00590
C.....		MAT00600

C	ASIN --	CALCULA EL ARCOSENO DEL REGISTRO X	MAT00610
C			MAT00620
40	IF(ABSOL.GT.1.)	GO TO 22	MAT00630
	STACK1(5)=(QARSIN(STACK1(5)))*CONST1/PI		MAT00640
	GO TO 90		MAT00650
C			MAT00660
C	ACOS --	CALCULA EL ARCOSENO DEL REGISTRO X	MAT00670
C			MAT00680
50	IF(ABSOL.GT.1.)	GO TO 22	MAT00690
	STACK1(5)=(QARCOS(STACK1(5)))*CONST1/PI		MAT00700
	GO TO 90		MAT00710
C			MAT00720
C	ATAN --	CALCULA EL ARCOTANGENTE DEL REGISTRO X	MAT00730
C			MAT00740
60	STACK1(5)=(QATAN(STACK1(5)))*CONST1/PI		MAT00750
	GO TO 90		MAT00760
C			MAT00770
C	MENSAJES DE ERROR		MAT00780
C			MAT00790
21	MSG='*--FUERA-DE-RANGO--*'		MAT00800
	GO TO 90		MAT00810
22	MSG='*--ERROR-DE-DATOS--*'		MAT00820
C			MAT00830
C	SE GRABAN LOS RESULTADOS DEL STACK EN EL ARCHIVO DE SALIDA		MAT00840
C			MAT00850
90	DO 89 I1=1,5		MAT00860
89	WRITE(8,*)STACK1(I1)		MAT00870
	WRITE(8,33)MSG		MAT00880
33	FORMAT(A20)		MAT00890
999	STOP		MAT00900
	END		MAT00910

C*****		MAT00010
C*		MAT00020
C*	M A T H O 6	MAT00030
C*		MAT00040
C* OPERACIONES LN NX LOG DX		MAT00050
C*****		MAT00060
REAL*16 STACK1(5),R1,R2,A1,A2,A3,A4		MAT00070
CHARACTER*20 MSG		MAT00080
MSG=' '		MAT00090
R1=-180.218		MAT00100
R2=174.673		MAT00110
A1=10.**75		MAT00120
A2=10.**(-75)		MAT00130
A3=-10.**75		MAT00140
A4=-10.**(-75)		MAT00150
C.....		MAT00160
C SE TOMAN LOS VALORES DEL STACK QUE VIENEN		MAT00170
C EN EL ARCHIVO DE ENTRADA		MAT00180
C.....		MAT00190
DO 14 I1=1,5		MAT00200
14 READ(1,*,END=999)STACK1(I1)		MAT00210
READ(1,I1,END=999)NUM		MAT00220
11 FORMAT(I1)		MAT00230
C.....		MAT00240
C SE VERIFICA QUE LOS DATOS ESTEN DENTRO DEL RANGO		MAT00250
C.....		MAT00260
IF (STACK1(5).EQ.0.) GO TO 71		MAT00270
IF (STACK1(5).GT.A1) GO TO 21		MAT00280
IF (STACK1(5).LT.A2.AND.STACK1(5).GT.A4) GO TO 21		MAT00290
IF (STACK1(5).LT.A3) GO TO 21		MAT00300
C.....		MAT00310
C CON EL CODIGO -NUM- SE IDENTIFICA LA OPERACION QUE		MAT00320
C SE DEBE EJECUTAR		MAT00330
C.....		MAT00340
71 GO TO (10,20,30,40)NUM		MAT00350
C.....		MAT00360
C LN -- CALCULA EL LOGARITMO NATURAL DEL REGISTRO X		MAT00370
C.....		MAT00380
10 IF (STACK1(5).LE.0.) GO TO 22		MAT00390
STACK1(5)=QLOG(STACK1(5))		MAT00400
GO TO 90		MAT00410
C.....		MAT00420
C NX -- CALCULA EL ANTILOGARITMO NATURAL DEL REGISTRO X		MAT00430
C.....		MAT00440
20 IF ((STACK1(5).GT.R2).OR.(STACK1(5).LT.R1)) GO TO 22		MAT00450
STACK1(5)=QEXP(STACK1(5))		MAT00460
GO TO 90		MAT00470
C.....		MAT00480
C LOG -- CALCULA EL LOGARITMO COMUN DEL REGISTRO X		MAT00490
C.....		MAT00500
30 IF (STACK1(5).LE.0.) GO TO 22		MAT00510
STACK1(5)=QLOG10(STACK1(5))		MAT00520
GO TO 90		MAT00530
C.....		MAT00540
C DX -- CALCULA EL ANTILOGARITMO COMUN DEL REGISTRO X		MAT00550
C.....		MAT00560
40 IF (STACK1(5).GT.75.) GO TO 21		MAT00570
STACK1(5)=10.**(STACK1(5))		MAT00580
GO TO 90		MAT00590
C.....		MAT00600

C	MENSAJES DE ERROR	MAT00610
C.....		MAT00620
21	MSG='*--FUERA-DE-RANGO--*'	MAT00630
	GO TO 90	MAT00640
22	MSG='*--ERROR-DE-DATOS--*'	MAT00650
C.....		MAT00660
C	SE GRABAN LOS RESULTADOS DEL STACK EN EL ARCHIVO DE SALIDA	MAT00670
C.....		MAT00680
90	DO 89 I1=1,5	MAT00690
89	WRITE(8,*)STACK1(I1)	MAT00700
	WRITE(8,33)MSG	MAT00710
33	FORMAT(A20)	MAT00720
999	STOP	MAT00730
	END	MAT00740

C*****		MAT00010
C*		MAT00020
C*	M A T H O 7	MAT00030
C*		MAT00040
C* OPERACIONES	STO RCL A+ A- MD CLR	MAT00050
C*	STO+ STO- STOX STO/	MAT00060
C*****		MAT00070
	REAL*16 STACK1(5),REG1(7),A1,A2,A3,A4	MAT00080
	CHARACTER*20 MSG	MAT00090
	MSG=' '	MAT00100
	A1=10.**75	MAT00110
	A2=10.**(-75)	MAT00120
	A3=-10.**75	MAT00130
	A4=-10.**(-75)	MAT00140
C.....		MAT00150
C	SE TOMAN LOS VALORES DEL STACK QUE VIENEN	MAT00160
C	EN EL ARCHIVO DE ENTRADA	MAT00170
C.....		MAT00180
	DO 14 I1=1,5	MAT00190
14	READ(1,*,END=999)STACK1(I1)	MAT00200
	READ(1,I1,END=999)NUM,NU1	MAT00210
11	FORMAT(I1,I1)	MAT00220
	DO 15 I1=1,7	MAT00230
15	READ(1,*,END=999)REG1(I1)	MAT00240
	READ(1,I1,END=999)IND	MAT00250
C.....		MAT00260
C	CON EL CODIGO -NUM- SE IDENTIFICA LA OPERACION QUE	MAT00270
C	SE DEBE EJECUTAR	MAT00280
C.....		MAT00290
	IF (NU1.EQ.9.) GO TO 10	MAT00300
	GO TO (1,2,3,4,5,6,7,8,9)NUM	MAT00310
C.....		MAT00320
C	SE VERIFICA QUE LOS DATOS ESTEN DENTRO DEL RANGO PERMITIDO	MAT00330
C	ANTES DE ALMACENARLOS EN LOS REGISTROS SECUNDARIOS	MAT00340
C.....		MAT00350
1	IF (STACK1(5).EQ.0.) GO TO 111	MAT00360
	IF (STACK1(5).GT.A1) GO TO 21	MAT00370
	IF (STACK1(5).LT.A2.AND.STACK1(5).GT.A4) GO TO 21	MAT00380
	IF (STACK1(5).LT.A3) GO TO 21	MAT00390
C.....		MAT00400
C STO ---	SE ALMACENA EL VALOR DEL REGISTRO X EN ALGUN REGISTRO	MAT00410
C	SECUNDARIO DEPENDIENDO DEL INDICE	MAT00420
C.....		MAT00430
111	REG1(IND)=STACK1(5)	MAT00440
	GO TO 90	MAT00450
C.....		MAT00460
C RCL ---	RECUPERA EN EL REGISTRO X ALGUN VALOR DEL REGISTRO SECUN-	MAT00470
C	DARIO DEPENDIENDO DEL INDICE	MAT00480
C.....		MAT00490
2	DO 25 I=1,4	MAT00500
25	STACK1(I)=STACK1(I+1)	MAT00510
	STACK1(5)=REG1(IND)	MAT00520
	GO TO 90	MAT00530
C.....		MAT00540
C	SE VERIFICA QUE LOS VALORES ESTEN DENTRO DEL RANGO AL EJECUTAR	MAT00550
C	LA OPERACION A+	MAT00560
C.....		MAT00570
3	IF (REG1(1)+STACK1(5).EQ.01) GO TO 333	MAT00580
	IF (REG1(1)+STACK1(5).GT.A1) GO TO 21	MAT00590
	IF (REG1(1)+STACK1(5).LT.A2.AND.REG1(1)+STACK1(5).GT.A4) GO TO 21	MAT00600

	IF (REG1(1)+STACK1(5).LT.A3) GO TO 21	MAT00610
C		MAT00620
333	IF (REG1(2)+(STACK1(5)**2.).EQ.0.) GO TO 444	MAT00630
	IF (REG1(2)+(STACK1(5)**2.).GT.A1) GO TO 21	MAT00640
	IF (REG1(2)+(STACK1(5)**2.).LT.A2.AND.REG1(2)+(STACK1(5)**2.).GT.	MAT00650
	1A4) GO TO 21	MAT00660
	IF (REG1(2)+(STACK1(5)**2.).LT.A3) GO TO 21	MAT00670
C		MAT00680
444	IF (REG1(3)+STACK1(4).EQ.0.) GO TO 555	MAT00690
	IF (REG1(3)+STACK1(4).GT.A1) GO TO 21	MAT00700
	IF (REG1(3)+STACK1(4).LT.A2.AND.REG1(3)+STACK1(4).GT.A4) GO TO 21	MAT00710
	IF (REG1(3)+STACK1(4).LT.A3) GO TO 21	MAT00720
C		MAT00730
555	IF (REG1(4)+(STACK1(4)**2.).EQ.0.) GO TO 666	MAT00740
	IF (REG1(4)+(STACK1(4)**2.).GT.A1) GO TO 21	MAT00750
	IF (REG1(4)+(STACK1(4)**2.).LT.A2.AND.REG1(4)+(STACK1(4)**2.).GT.	MAT00760
	1A4) GO TO 21	MAT00770
	IF (REG1(4)+(STACK1(4)**2.).LT.A3) GO TO 21	MAT00780
C		MAT00790
666	IF (REG1(5)+(STACK1(5)*STACK1(4)).EQ.0.) GO TO 777	MAT00800
	IF (REG1(5)+(STACK1(5)*STACK1(4)).GT.A1) GO TO 21	MAT00810
	IF (REG1(5)+(STACK1(5)*STACK1(4)).LT.A2.AND.REG1(5)+(STACK1(5)*ST	MAT00820
	ACK1(4)).GT.A4) GO TO 21	MAT00830
	IF (REG1(5)+(STACK1(5)*STACK1(4)).LT.A3) GO TO 21	MAT00840
C		MAT00850
777	IF (REG1(6)+1..EQ.0.) GO TO 888	MAT00860
	IF (REG1(6)+1..GT.A1) GO TO 21	MAT00870
	IF (REG1(6)+1..LT.A2.AND.REG1(6)+1..GT.A4) GO TO 21	MAT00880
	IF (REG1(6)+1..LT.A3) GO TO 21	MAT00890
C		MAT00900
C	A+ -- ACUMULA SUMAS Y PRODUCTOS DE LOS VALORES EN LOS REGISTROS	MAT00910
C	X E Y.	MAT00920
C		MAT00930
C	R1 = E X	MAT00940
C	R2 = E X**2	MAT00950
C	R3 = E Y	MAT00960
C	R4 = E Y**2	MAT00970
C	R5 = E X*Y	MAT00980
C	R6 = N	MAT00990
C		MAT01000
888	REG1(1)=REG1(1)+STACK1(5)	MAT01010
	REG1(2)=REG1(2)+(STACK1(5)**2.)	MAT01020
	REG1(3)=REG1(3)+STACK1(4)	MAT01030
	REG1(4)=REG1(4)+(STACK1(4)**2.)	MAT01040
	REG1(5)=REG1(5)+(STACK1(5)*STACK1(4))	MAT01050
	REG1(6)=REG1(6)+1	MAT01060
	STACK1(5)=REG1(6)	MAT01070
	GO TO 90	MAT01080
C		MAT01090
C	SE VERIFICA QUE LOS VALORES ESTEN DENTRO DEL RANGO AL EJECUTAR	MAT01100
C	LA OPERACION A-	MAT01110
C		MAT01120
4	IF (REG1(1)-STACK1(5).EQ.0.) GO TO 212	MAT01130
	IF (REG1(1)-STACK1(5).GT.A1) GO TO 21	MAT01140
	IF (REG1(1)-STACK1(5).LT.A2.AND.REG1(1)-STACK1(5).GT.A4) GO TO 21	MAT01150
	IF (REG1(1)-STACK1(5).LT.A3) GO TO 21	MAT01160
C		MAT01170
212	IF (REG1(2)-(STACK1(5)**2.).EQ.0.) GO TO 213	MAT01180
	IF (REG1(2)-(STACK1(5)**2.).GT.A1) GO TO 21	MAT01190
	IF (REG1(2)-(STACK1(5)**2.).LT.A2.AND.REG1(2)-(STACK1(5)**2.).GT.	MAT01200

1A4)	GO TO 21	MAT01210
	IF (REG1(2)-(STACK1(5)**2.).LT.A3) GO TO 21	MAT01220
C		MAT01230
213	IF (REG1(3)-STACK1(4).EQ.0.) GO TO 214	MAT01240
	IF (REG1(3)-STACK1(4).GT.A1) GO TO 21	MAT01250
	IF (REG1(3)-STACK1(4).LT.A2.AND.REG1(3)-STACK1(4).GT.A4) GO TO 21	MAT01260
	IF (REG1(3)-STACK1(4).LT.A3) GO TO 21	MAT01270
C		MAT01280
214	IF (REG1(4)-(STACK1(4)**2.).EQ.0.) GO TO 215	MAT01290
	IF (REG1(4)-(STACK1(4)**2.).GT.A1) GO TO 21	MAT01300
	IF (REG1(4)-(STACK1(4)**2.).LT.A2.AND.REG1(4)-(STACK1(4)**2.).GT.	MAT01310
1A4)	GO TO 21	MAT01320
	IF (REG1(4)-(STACK1(4)**2.).LT.A3) GO TO 21	MAT01330
C		MAT01340
215	IF (REG1(5)-(STACK1(5)*STACK1(4)).EQ.0.) GO TO 216	MAT01350
	IF (REG1(5)-(STACK1(5)*STACK1(4)).GT.A1) GO TO 21	MAT01360
	IF (REG1(5)-(STACK1(5)*STACK1(4)).LT.A2.AND.REG1(5)-(STACK1(5)*ST	MAT01370
ACK1(4)).GT.A4) GO TO 21		MAT01380
	IF (REG1(5)-(STACK1(5)*STACK1(4)).LT.A3) GO TO 21	MAT01390
C		MAT01400
216	IF (REG1(6)-1..EQ.0.) GO TO 217	MAT01410
	IF (REG1(6)-1..GT.A1) GO TO 21	MAT01420
	IF (REG1(6)-1..LT.A2.AND.REG1(6)-1..GT.A4) GO TO 21	MAT01430
	IF (REG1(6)-1..LT.A3) GO TO 21	MAT01440
C		MAT01450
C	A- -- RESTA SUMAS Y PRODUCTOS DE LOS VALORES EN LOS REGISTROS	MAT01460
C	X E Y.	MAT01470
C		MAT01480
217	REG1(1)=REG1(1)-STACK1(5)	MAT01490
	REG1(2)=REG1(2)-(STACK1(5)**2.)	MAT01500
	REG1(3)=REG1(3)-STACK1(4)	MAT01510
	REG1(4)=REG1(4)-(STACK1(4)**2.)	MAT01520
	REG1(5)=REG1(5)-(STACK1(5)*STACK1(4))	MAT01530
	REG1(6)=REG1(6)-1.	MAT01540
	STACK1(5)=REG1(6)	MAT01550
	GO TO 90	MAT01560
C		MAT01570
C	VERIFICA QUE LOS DATOS ESTEN DENTRO DEL RANGO ANTES	MAT01580
C	DE EJECUTAR LA OPERACION MD	MAT01590
C		MAT01600
5	IF (REG1(6).EQ.0.) GO TO 22	MAT01610
C		MAT01620
	IF (REG1(1)/REG1(6).EQ.0.) GO TO 987	MAT01630
	IF (REG1(1)/REG1(6).GT.A1) GO TO 21	MAT01640
	IF (REG1(1)/REG1(6).LT.A2.AND.REG1(1)/REG1(6).GT.A4) GO TO 21	MAT01650
	IF (REG1(1)/REG1(6).LT.A3) GO TO 21	MAT01660
C		MAT01670
987	IF (REG1(3)/REG1(6).EQ.0.) GO TO 229	MAT01680
	IF (REG1(3)/REG1(6).GT.A1) GO TO 21	MAT01690
	IF (REG1(3)/REG1(6).LT.A2.AND.REG1(3)/REG1(6).GT.A4) GO TO 21	MAT01700
	IF (REG1(3)/REG1(6).LT.A3) GO TO 21	MAT01710
C		MAT01720
C	MD -- CALCULA LA MEDIA (PROMEDIO ARITMETICO) DE LOS VALORES	MAT01730
C	X E Y ACUMULADOS EN LOS REG. ESTADISTICOS	MAT01740
C		MAT01750
229	STACK1(5)=REG1(1)/REG1(6)	MAT01760
	STACK1(4)=REG1(3)/REG1(6)	MAT01770
	GO TO 90	MAT01780
C		MAT01790
C	CLR -- ENCERA TODOS LOS REGISTROS SECUNDARIOS	MAT01800

C		MAT01810
6	DO 65 I=1,7	MAT01820
65	REG1(I)=0.	MAT01830
	GO TO 90	MAT01840
C		MAT01850
C	SE VERIFICA QUE LOS VALORES ESTEN DENTRO DEL RANGO AL EJECUTAR	MAT01860
C	LA OPERACION STO+	MAT01870
C		MAT01880
7	IF (REG1(IND)+STACK1(5).EQ.0.) GO TO 79	MAT01890
	IF (REG1(IND)+STACK1(5).GT.A1) GO TO 21	MAT01900
	IF (REG1(IND)+STACK1(5).LT.A2.AND.REG1(IND)+STACK1(5).GT.A4) GO T	MAT01910
	10 21	MAT01920
	IF (REG1(IND)+STACK1(5).LT.A3) GO TO 21	MAT01930
C		MAT01940
C	STO+ -- EL NUMERO EN EL REG. X SE SUMA AL CONTENIDO DEL REG. N	MAT01950
C		MAT01960
79	REG1(IND)=REG1(IND)+STACK1(5)	MAT01970
	GO TO 90	MAT01980
C		MAT01990
C	SE VERIFICA QUE LOS VALORES ESTEN DENTRO DEL RANGO AL EJECUTAR	MAT02000
C	LA OPERACION STO-	MAT02010
C		MAT02020
8	IF (REG1(IND)-STACK1(5).EQ.0.) GO TO 82	MAT02030
	IF (REG1(IND)-STACK1(5).GT.A1) GO TO 21	MAT02040
	IF (REG1(IND)-STACK1(5).LT.A2.AND.REG1(IND)-STACK1(5).GT.A4) GO T	MAT02050
	10 21	MAT02060
	IF (REG1(IND)-STACK1(5).LT.A3) GO TO 21	MAT02070
C		MAT02080
C	STO- -- EL NUMERO EN EL REG. X SE RESTA DEL CONTENIDO DEL REG. N	MAT02090
C		MAT02100
82	REG1(IND)=REG1(IND)-STACK1(5)	MAT02110
	GO TO 90	MAT02120
C		MAT02130
C	SE VERIFICA QUE LOS VALORES ESTEN DENTRO DEL RANGO AL EJECUTAR	MAT02140
C	LA OPERACION STOX	MAT02150
C		MAT02160
9	IF (REG1(IND)*STACK1(5).EQ.0.) GO TO 96	MAT02170
	IF (REG1(IND)*STACK1(5).GT.A1) GO TO 21	MAT02180
	IF (REG1(IND)*STACK1(5).LT.A2.AND.REG1(IND)*STACK1(5).GT.A4) GO T	MAT02190
	10 21	MAT02200
	IF (REG1(IND)*STACK1(5).LT.A3) GO TO 21	MAT02210
C		MAT02220
C	STOX -- EL NUMERO EN EL REG. X SE MULTIPLICA POR EL CONTENIDO	MAT02230
C	DEL REG. N	MAT02240
C		MAT02250
96	REG1(IND)=REG1(IND)*STACK1(5)	MAT02260
	GO TO 90	MAT02270
C		MAT02280
C	SE VERIFICA QUE LOS VALORES ESTEN DENTRO DEL RANGO AL EJECUTAR	MAT02290
C	LA OPERACION STO/	MAT02300
C		MAT02310
10	IF (STACK1(5).EQ.0.) GO TO 22	MAT02320
	IF (REG1(IND)/STACK1(5).EQ.0.) GO TO 101	MAT02330
	IF (REG1(IND)/STACK1(5).GT.A1) GO TO 21	MAT02340
	IF (REG1(IND)/STACK1(5).LT.A2.AND.REG1(IND)/STACK1(5).GT.A4) GO T	MAT02350
	10 21	MAT02360
	IF (REG1(IND)/STACK1(5).LT.A3) GO TO 21	MAT02370
C		MAT02380
C	STO/ -- EL NUMERO EN EL REG. X SE DIVIDE PARA EL CONTENIDO	MAT02390
C	DEL REG. N	MAT02400

C.....		MAT02410
101	REG1(IND)=REG1(IND)/STACK1(5)	MAT02420
	GO TO 90	MAT02430
C.....		MAT02440
C	MENSAJES DE ERROR	MAT02450
C.....		MAT02460
21	MSG='*--FUERA-DE-RANGO--*'	MAT02470
	GO TO 90	MAT02480
22	MSG='*--ERROR-DE-DATOS--*'	MAT02490
C.....		MAT02500
C	SE GRABAN LOS RESULTADOS DEL STACK EN EL ARCHIVO DE SALIDA	MAT02510
C.....		MAT02520
90	DO 89 I1=1,5	MAT02530
89	WRITE(8,*)STACK1(I1)	MAT02540
	WRITE(8,33)MSG	MAT02550
33	FORMAT(A20)	MAT02560
	DO 98 I1=1,7	MAT02570
98	WRITE(8,*)REG1(I1)	MAT02580
999	STOP	MAT02590
	END	MAT02600

C*****	MAT00010
C*	MAT00020
C* M A T H 0 8	MAT00030
C* OPERACIONES HR HMS HMS+ HMS-	MAT00040
C*****	MAT00050
REAL*16 STACK1(5),HR,HMS,A1,A2,A3,A4	MAT00060
CHARACTER*20 MSG	MAT00070
MSG=' '	MAT00080
MCOD=0	MAT00090
A1=10.**75	MAT00100
A2=10.**(-75)	MAT00110
A3=-10.**75	MAT00120
A4=-10.**(-75)	MAT00130
C.....	MAT00140
C SE TOMAN LOS VALORES DEL STACK QUE VIENEN	MAT00150
C EN EL ARCHIVO DE ENTRADA	MAT00160
C.....	MAT00170
DO 14 I1=1,5	MAT00180
14 READ(1,*,END=999)STACK1(I1)	MAT00190
READ(1,I1,END=999)NUM	MAT00200
11 FORMAT(I1)	MAT00210
C.....	MAT00220
C SE VERIFICA QUE LOS DATOS ESTEN DENTRO DEL RANGO	MAT00230
C.....	MAT00240
IF (STACK1(5).EQ.0.) GO TO 111	MAT00250
IF (STACK1(5).GT.A1) GO TO 21	MAT00260
IF (STACK1(5).LT.A2.AND.STACK1(5).GT.A4) GO TO 21	MAT00270
IF (STACK1(5).LT.A3) GO TO 21	MAT00280
C.....	MAT00290
C CON EL CODIGO -NUM- SE IDENTIFICA LA OPERACION QUE	MAT00300
C SE DEBE EJECUTAR	MAT00310
C.....	MAT00320
111 GO TO (1,2,3,4)NUM	MAT00330
C.....	MAT00340
C HR -- EJECUTA LA SUBROUTINA QUE TRANSFORMA HH-MM-SS A HORAS DECIMALES	MAT00350
C.....	MAT00360
1 CALL HORAS (STACK1(5),HR,A1,A2,A3,A4,MCOD)	MAT00370
IF (MCOD.EQ.1) GO TO 21	MAT00380
STACK1(5)=HR	MAT00390
GO TO 9	MAT00400
C.....	MAT00410
C HMS - EJECUTA LA SUBROUTINA QUE TRANSFORMA HORAS DECIMALES A HH-MM-SS	MAT00420
C.....	MAT00430
2 CALL HHMMSS (STACK1(5),HMS,A1,A2,A3,A4,MCOD)	MAT00440
IF (MCOD.EQ.1) GO TO 21	MAT00450
STACK1(5)=HMS	MAT00460
GO TO 9	MAT00470
C.....	MAT00480
C SE VERIFICA QUE LOS DATOS ESTEN DENTRO DEL RANGO	MAT00490
C ANTES DE EJECUTAR HMS+	MAT00500
C.....	MAT00510
3 IF (STACK1(4).EQ.0.) GO TO 222	MAT00520
IF (STACK1(4).GT.A1) GO TO 21	MAT00530
IF (STACK1(4).LT.A2.AND.STACK1(4).GT.A4) GO TO 21	MAT00540
IF (STACK1(4).LT.A3) GO TO 21	MAT00550
C.....	MAT00560
C HMS+ - SUMA EN HH-MM-SS LOS VALORES DE LOS REGISTROS X Y Y	MAT00570
C.....	MAT00580
222 CALL HORAS (STACK1(5),HR,A1,A2,A3,A4,MCOD)	MAT00590
IF (MCOD.EQ.1) GO TO 21	MAT00600

STACK1(5)=HR	MAT00610
CALL HORAS (STACK1(4),HR,A1,A2,A3,A4,MCOD)	MAT00620
IF (MCOD.EQ.1) GO TO 21	MAT00630
C.....	MAT00640
C SE VERIFICA LOS RANGOS PERMITIDOS	MAT00650
C.....	MAT00660
IF (STACK1(5)+HR.EQ.0.) GO TO 601	MAT00670
IF (STACK1(5)+HR.GT.A1) GO TO 21	MAT00680
IF (STACK1(5)+HR.LT.A2.AND.STACK1(5)+HR.GT.A4) GO TO 21	MAT00690
IF (STACK1(5)+HR.LT.A3) GO TO 21	MAT00700
C.....	MAT00710
601 STACK1(5)=STACK1(5)+HR	MAT00720
CALL HHMMSS (STACK1(5),HMS,A1,A2,A3,A4,MCOD)	MAT00730
IF (MCOD.EQ.1) GO TO 21	MAT00740
STACK1(5)=HMS	MAT00750
GO TO 90	MAT00760
C.....	MAT00770
C SE VERIFICA QUE LOS DATOS ESTEN DENTRO DEL RANGO	MAT00780
C ANTES DE EJECUTAR HMS-	MAT00790
C.....	MAT00800
4 IF (STACK1(4).EQ.0.) GO TO 602	MAT00810
IF (STACK1(4).GT.A1) GO TO 21	MAT00820
IF (STACK1(4).LT.A2.AND.STACK1(4).GT.A4) GO TO 21	MAT00830
IF (STACK1(4).LT.A3) GO TO 21	MAT00840
C.....	MAT00850
C HMS- - RESTA EN HH-MM-SS LOS VALORES DE LOS REGISTROS Y Y X	MAT00860
C.....	MAT00870
602 CALL HORAS (STACK1(5),HR,A1,A2,A3,A4,MCOD)	MAT00880
IF (MCOD.EQ.1) GO TO 21	MAT00890
STACK1(5)=HR	MAT00900
CALL HORAS (STACK1(4),HR,A1,A2,A3,A4,MCOD)	MAT00910
IF (MCOD.EQ.1) GO TO 21	MAT00920
C.....	MAT00930
C SE VERIFICA LOS RANGOS PERMITIDOS	MAT00940
C.....	MAT00950
IF (HR-STACK1(5).EQ.0.) GO TO 604	MAT00960
IF (HR-STACK1(5).GT.A1) GO TO 21	MAT00970
IF (HR-STACK1(5).LT.A2.AND.HR-STACK1(5).GT.A4) GO TO 21	MAT00980
IF (HR-STACK1(5).LT.A3) GO TO 21	MAT00990
C.....	MAT01000
604 STACK1(5)=HR-STACK1(5)	MAT01010
CALL HHMMSS (STACK1(5),HMS,A1,A2,A3,A4,MCOD)	MAT01020
IF (MCOD.EQ.1) GO TO 21	MAT01030
STACK1(5)=HMS	MAT01040
C.....	MAT01050
C SE REALIZA LA OPERACION DE DESCENSO DEL STACK	MAT01060
C.....	MAT01070
90 DO 95 I=4,2,-1	MAT01080
95 STACK1(I)=STACK1(I-1)	MAT01090
GO TO 9	MAT01100
C.....	MAT01110
C MENSAJE DE ERROR	MAT01120
C.....	MAT01130
21 MSG='*--FUERA-DE-RANGO--*'	MAT01140
C.....	MAT01150
C SE GRABAN LOS RESULTADOS DEL STACK EN EL ARCHIVO DE SALIDA	MAT01160
C.....	MAT01170
9 DO 89 I1=1,5	MAT01180
89 WRITE(8,*)STACK1(I1)	MAT01190
WRITE(8,33)MSG	MAT01200

33	FORMAT(A20)	MAT01210
999	STOP	MAT01220
	END	MAT01230
C	*****	MAT01240
C	SUBROUTINA QUE CONVIERTE HH-MM-SS A HORAS DECIMALES	MAT01250
C	*****	MAT01260
	SUBROUTINE HORAS (RNUM,HR1,A1,A2,A3,A4,MCOD)	MAT01270
	REAL*16 MM,RNUM,HR1,HH,FR,SS,UNO,A1,A2,A3,A4	MAT01280
	UNO=1.	MAT01290
	HH=IQINT(RNUM)	MAT01300
C		MAT01310
C	SE VERIFICA RANGOS	MAT01320
C		MAT01330
	IF ((QMOD(RNUM,UNO))*100.EQ.0.) GO TO 604	MAT01340
	IF ((QMOD(RNUM,UNO))*100.GT.A1) GO TO 211	MAT01350
	IF ((QMOD(RNUM,UNO))*100.LT.A2.AND.(QMOD(RNUM,UNO))*100.GT.A4) GO	MAT01360
	1 TO 211	MAT01370
	IF ((QMOD(RNUM,UNO))*100.LT.A3) GO TO 211	MAT01380
C		MAT01390
604	FR=(QMOD(RNUM,UNO))*100.	MAT01400
C		MAT01410
C	SE VERIFICA RANGOS	MAT01420
C		MAT01430
	IF ((IQINT(FR))/60.EQ.0.) GO TO 605	MAT01440
	IF ((IQINT(FR))/60.GT.A1) GO TO 211	MAT01450
	IF ((IQINT(FR))/60.LT.A2.AND.(IQINT(FR))/60.GT.A4) GO TO 211	MAT01460
	IF ((IQINT(FR))/60.LT.A3) GO TO 211	MAT01470
C		MAT01480
605	MM=(IQINT(FR))/60.	MAT01490
C		MAT01500
C	SE VERIFICA RANGOS	MAT01510
C		MAT01520
	IF ((QMOD(FR,UNO)*100.)/3600.EQ.0.) GO TO 606	MAT01530
	IF ((QMOD(FR,UNO)*100.)/3600.GT.A1) GO TO 211	MAT01540
	IF ((QMOD(FR,UNO)*100.)/3600.LT.A2.AND.(QMOD(FR,UNO)*100.)/3600.G	MAT01550
	1T.A4) GO TO 211	MAT01560
	IF ((QMOD(FR,UNO)*100.)/3600.LT.A3) GO TO 211	MAT01570
C		MAT01580
606	SS=(QMOD(FR,UNO)*100.)/3600.	MAT01590
C		MAT01600
C	SE VERIFICA RANGOS	MAT01610
C		MAT01620
	IF (HH+MM+SS.EQ.0.) GO TO 607	MAT01630
	IF (HH+MM+SS.GT.A1) GO TO 211	MAT01640
	IF (HH+MM+SS.LT.A2.AND.HH+MM+SS.GT.A4) GO TO 211	MAT01650
	IF (HH+MM+SS.LT.A3) GO TO 211	MAT01660
C		MAT01670
607	HR1=HH+MM+SS	MAT01680
	GO TO 212	MAT01690
211	MCOD=1	MAT01700
212	RETURN	MAT01710
	END	MAT01720
C	*****	MAT01730
C	SUBROUTINA QUE CONVIERTE HORAS DECIMALES A HH-MM-SS	MAT01740
C	*****	MAT01750
	SUBROUTINE HHMMSS (RNUM1,HMS1,A1,A2,A3,A4,MCOD)	MAT01760
	REAL*16 MM1,RNUM1,HMS1,HH1,FR1,SS1,UNO,A1,A2,A3,A4	MAT01770
	UNO=1.	MAT01780
	HH1=IQINT(RNUM1)	MAT01790
C		MAT01800

C	SE VERIFICA RANGOS	MAT01810
C		MAT01820
	IF ((QMOD(RNUM1,UNO))*60.EQ.0.) GO TO 608	MAT01830
	IF ((QMOD(RNUM1,UNO))*60.GT.A1) GO TO 221	MAT01840
	IF ((QMOD(RNUM1,UNO))*60.LT.A2.AND.(QMOD(RNUM1,UNO))*60.GT.A4) GO	MAT01850
	1 TO 221	MAT01860
	IF ((QMOD(RNUM1,UNO))*60.LT.A3) GO TO 221	MAT01870
C		MAT01880
608	FR1=(QMOD(RNUM1,UNO))*60.	MAT01890
C		MAT01900
C	SE VERIFICA RANGOS	MAT01910
C		MAT01920
	IF ((IQINT(FR1))/100.EQ.0.) GO TO 609	MAT01930
	IF ((IQINT(FR1))/100.GT.A1) GO TO 221	MAT01940
	IF ((IQINT(FR1))/100.LT.A2.AND.(IQINT(FR1))/100.GT.A4) GO TO 221	MAT01950
	IF ((IQINT(FR1))/100.LT.A3) GO TO 221	MAT01960
C		MAT01970
609	MM1=(IQINT(FR1))/100.	MAT01980
C		MAT01990
C	SE VERIFICA RANGOS	MAT02000
C		MAT02010
	IF ((QMOD(FR1,UNO)*60.)/10000.EQ.0.) GO TO 619	MAT02020
	IF ((QMOD(FR1,UNO)*60.)/10000.GT.A1) GO TO 221	MAT02030
	IF ((QMOD(FR1,UNO)*60.)/10000.LT.A2.AND.(QMOD(FR1,UNO)*60.)/10000	MAT02040
	1.GT.A4) GO TO 221	MAT02050
	IF ((QMOD(FR1,UNO)*60.)/10000.LT.A3) GO TO 221	MAT02060
C		MAT02070
619	SS1=(QMOD(FR1,UNO)*60.)/10000.	MAT02080
C		MAT02090
C	SE VERIFICA RANGOS	MAT02100
C		MAT02110
	IF (HH1+MM1+SS1.EQ.0.) GO TO 610	MAT02120
	IF (HH1+MM1+SS1.GT.A1) GO TO 221	MAT02130
	IF (HH1+MM1+SS1.LT.A2.AND.HH1+MM1+SS1.GT.A4) GO TO 221	MAT02140
	IF (HH1+MM1+SS1.LT.A3) GO TO 221	MAT02150
C		MAT02160
610	HMS1=HH1+MM1+SS1	MAT02170
	GO TO 222	MAT02180
221	MCOD=1	MAT02190
222	RETURN	MAT02200
	END	MAT02210

\$\$\$004**CLAVE	PROGRAM	CODIGO	ULTIMOX
\$\$\$004**K	X	X	X
\$\$\$004**ABS	MATH01	1	SI
\$\$\$004**INT	MATH01	2	SI
\$\$\$004**FRAC	MATH01	3	SI
\$\$\$004**CHS	MATH01	4	NO
\$\$\$004**POR	MATH01	5	SI
\$\$\$004**{	MATH01	5	SI
\$\$\$004**DPOR	MATH01	6	SI
\$\$\$004**{CH	MATH01	6	SI
\$\$\$004**MOD	MATH01	7	SI
\$\$\$004**RDO	MATH02	1	NO
\$\$\$004**RUP	MATH02	2	NO
\$\$\$004**XCY	MATH02	3	NO
\$\$\$004**CLX	MATH02	4	NO
\$\$\$004**CLST	MATH02	5	NO
\$\$\$004**ADD	MATH03	1	SI
\$\$\$004**+	MATH03	1	SI
\$\$\$004**SUB	MATH03	2	SI
\$\$\$004**-	MATH03	2	SI
\$\$\$004**MULT	MATH03	3	SI
\$\$\$004**X	MATH03	3	SI
\$\$\$004**DIV	MATH03	4	SI
\$\$\$004**/	MATH03	4	SI
\$\$\$004**YX	MATH03	5	SI
\$\$\$004**INV	MATH04	1	SI
\$\$\$004**1/X	MATH04	1	SI
\$\$\$004**SQ	MATH04	2	SI
\$\$\$004**SQRT	MATH04	3	SI
\$\$\$004**SIN	MATH05	1	SI
\$\$\$004**COS	MATH05	2	SI
\$\$\$004**TAN	MATH05	3	SI
\$\$\$004**ASIN	MATH05	4	SI
\$\$\$004**ACOS	MATH05	5	SI
\$\$\$004**ATAN	MATH05	6	SI
\$\$\$004**LN	MATH06	1	SI
\$\$\$004**NX	MATH06	2	SI
\$\$\$004**LOG	MATH06	3	SI
\$\$\$004**DX	MATH06	4	SI
\$\$\$004**A+	MATH07	3	SI
\$\$\$004**A-	MATH07	4	SI
\$\$\$004**MD	MATH07	5	SI
\$\$\$004**CLR	MATH07	6	NO
\$\$\$004**HR	MATH08	1	SI
\$\$\$004**HMS	MATH08	2	SI
\$\$\$004**HMS+	MATH08	3	SI
\$\$\$004**HMS-	MATH08	4	SI
\$\$\$EOF**			

=ENTRY

DISE=0 DE PANTALLA

* ESCUELA SUPERIOR POLITECNICA DEL LITORAL *

STACK

REGISTROS SECUNDARIOS

```

=====
W  (&ASAL1          &NS1          LAST X          R1  (&ASTOR1          &NR1
=====
T  (&ASAL2          &NS2          (&ALSTX          &NLX R2  (&ASTOR2          &NR2
=====
Z  (&ASAL3          &NS3                                     R3  (&ASTOR3          &NR3
=====
Y  (&ASAL4          &NS4                                     R4  (&ASTOR4          &NR4
(&=====
IX (&ASAL5          &NS5                                     R5  (&ASTOR5          &NR5
(&=====
(&MSG
--  =15&CHRIS
    ZONA DE COMANDOS

```