

PROPOSICIONES DE ENTRADA, SALIDA Y DE FORMATO.**INTRODUCCION.**

En esta unidad vamos a tomar en consideración los métodos por medio de los que pueden leerse los datos en la computadora bajo el control de nuestro programa, y, además, los modos en los que pueden imprimirse o mecanografiarse los resultados en la salida, después de que se hayan completado los cálculos.

Los resultados de nuestros cálculos pueden presentarse en numerosas formas distintas. El método más común, es el de imprimirlos utilizando una impresora de renglones de alta velocidad. Pueden perforarse en tarjetas, almacenarse en cintas o discos magnéticos, perforarse en cintas de papel o, incluso, trazarse automáticamente, por medio de trazadores especiales de gráficas.

La regla que se debe tener siempre en cuenta al utilizar proposiciones que implican la lectura o la impresión de datos es que :

"Para poder realizar una operación de lectura o escritura la computadora debe tener dos conjuntos diferentes de informaciones. El primero le indicará cuáles son las variables involucradas, mientras que el segundo le dará instrucciones sobre el modo en que deberá efectuar la operación"

OBJETIVOS .

Al terminar esta unidad deberás ser capaz de :

1. Dada una proposición de READ o WRITE, identificar cuál es la lista de variables, el número que indica la entrada ó salida de datos, el número que indica el formato.
2. Explicar con tus propias palabras el significado de las operaciones READ, WRITE, FORMAT.
3. Dada una proposición de READ, su FORMAT y una o varias tarjetas a leer, asociar las variables con sus valores correspondientes.
4. Dada una proposición WRITE, su FORMAT y el valor de las variables que se imprimen, asociar las variables con los datos cómo quedaron impresos los resultados.

5. Representar por tarjetas cómo deberán proporcionarse los datos para una proposición READ de acuerdo a un FORMAT.
6. Dada una lista de proposiciones READ y WRITE y su formato, poder identificar cuáles tienen error.
7. Dado el contenido de unas variables, y la forma en que se desea imprimir—las, construir las proposiciones WRITE y FORMAT necesarios para lograr—lo.
8. Definir el significado de las proposiciones READ, WRITE y los formatos F, I, X, H.
9. Practicar las proposiciones READ, WRITE, haciendo uso de los formatos F, I, X, H, etc.

PROCEDIMIENTO DE APRENDIZAJE.

1. Estudiar el material anexo a la unidad.
2. Lee las páginas 37 a 41 del RULE. Programación con FORTRAN IV, ó cualquier otro libro que contenga el tema.

EXAMEN DE AUTOEVALUACION.

Estarás listo para contestar preguntas de los objetivos cuando puedas resolver el siguiente cuestionario:

1. En la siguiente lista de proposiciones READ y WRITE identifica cuáles tienen error. ¿Por qué?
 - a. READ (5, 5) A B
 5 FORMAT (I2, I3, 15F5.0)
 - b. READ (5, 10) A, B, C
 10 FØRMAT (2F5.0)
 - c. WRITE (6, 8) 5, X(I)
 8 FØRMAT (I10, F10.0)
 - d. WRITE (6, 10) A, B, C, IJ, P, NA
 10 FØRMAT (F6.2, F8.3, F10.4, I5, F10.3, I5)
2. Escribe correctamente las proposiciones que tengan error del inciso anterior.

3. Dada la siguiente lista de variables y de valores correspondientes.
- Hacer la instrucción de lectura y el formato correspondiente. Las variables se deben leer en el orden indicado.
 - Indicar los datos en una tarjeta

A = 86.4
 B = - 91.98
 C = 555.98
 IJ = 75
 P = 86.88
 NA = 98

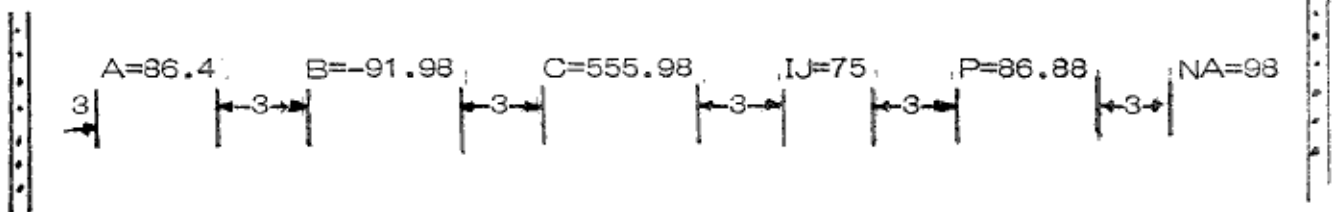
4. Si de los datos anteriores se tiene la siguiente instrucción

```
WRITE (6, 10)A, B, C, IJ, P, NA
10  FØRMAT (F6.2, F8.3, F10.4, I5, F10.3, I5)
```

```
WRITE (6, 11)A, B, C, IJ, P, NA
11  FØRMAT (3F10.4, I5, F10.4, I5)
```

En ambos casos mostrar lo que se obtendría de salida.

5. Hacer la instrucción o instrucciones para que nos imprima como salida en un solo renglón lo siguiente :



6. Hacer las instrucciones necesarias para escribir los siguientes rótulos :

LOS DATOS DEL PROBLEMA SON

LOS RESULTADOS OBTENIDOS SON

MATERIAL ANEXO.

PROPOSICION READ :

Este postulado se utiliza para leer datos al almacenamiento de la computadora, por lo común a partir de tarjetas perforadas. (Pueden utilizarse como dispositivo de entrada: máquina de escribir eléctrica o consola, lectora de tarjetas, lectora de cintas perforadas, lectora de cinta magnética, explorador óptico y tubos de rayos catódicos). Veamos un ejemplo

$$\text{READ}(5, 12) A(7), K(9)$$

Esto le indica realmente a la computadora que: "Lea de algún dispositivo de entrada —la lectora de tarjetas— al que se da el número 5 de código, dos números, y asignense a las variables A(7) y K(9), respectivamente. La información sobre cómo serán esos números se encontrará en la proposición 12".

En este caso, la proposición 12 es una instrucción no ejecutable, denominada —proposición FORMAT, que estudiaremos más adelante. La forma general de la proposición READ es:

$$\text{READ}(m, n) \text{ lista}$$

donde

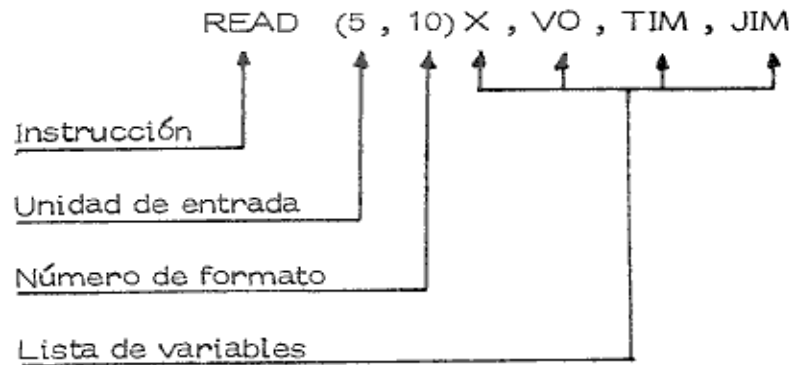
- m = es el número del dispositivo de entrada
- n = número de proposición correspondiente a una proposición FORMAT
- lista = es una secuencia ordenada de uno o más nombres de variables separadas por comas.

La lista que forma parte de la proposición READ puede contener cierto número de variables diferentes. Pueden ser enteras o reales con o sin subíndice y que carecerá de signo. Esto quiere decir que:

$$\text{READ}(5, 34) + N, -H(9) \quad \underline{\text{no será válido}}$$

Puesto que el signo "+" o "-" no están permitidos. Esto no quiere decir que no sea posible leer números negativos; los signos negativos son totalmente válidos en las tarjetas de datos; pero no en las listas del postulado READ. Acuerdate — que las variables empiezan con una letra.

Ejemplo :



Lo anterior se lee de la siguiente manera :

"Lee, la unidad de entrada número 5 de acuerdo con el formato número 10, los valores para las variables X, VO, TIM, JIM".

La proposición READ hace que el dispositivo de entrada especificado examine un archivo de datos y transmita la información que allí se encuentra, a las localidades del almacenamiento designadas por los nombres en la lista. El orden de transmisión de los números examinados va de acuerdo con el orden de las variables en la lista. El primer número examinado se almacenará en la localización reservada para la primera variable nombrada en la lista, el segundo número se transmitirá en el almacenamiento correspondiente a la segunda variable, etc.

El modo de los datos transmitidos debe corresponder con el modo denotado por el nombre de la variable. Es decir, un valor entero debe almacenarse en una localización reservada a una variable entera; un número real debe ir a una localización reservada para una variable real.

Ejemplo :

`READ (5, 10)XP, R70, R95, J1, J9, R1`

Los valores de cada una serían (escritos en una tarjeta de acuerdo a un formato)

-38.4		67.81		-75.86		28		780		98.65	
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9	9	9

IBM de México S. A. 75 4727

Es de donde

XP	=	- 38.4
R70	=	67.81
R95	=	- 75.86
J1	=	28
J9	=	780
R1	=	98.65

PROPOSICION WRITE :

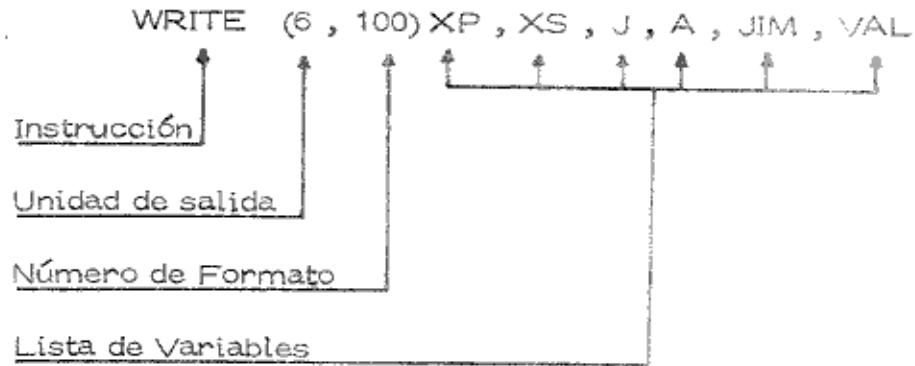
Ya vimos que una variable es una localidad de memoria en donde se guarda algún valor numérico y ya se vió también distintas formas de asignarle valores a las variables. Ahora nuestro problema consiste en pedir los valores que hay en ciertas variables que hemos empleado en algún programa determinado.

La proposición WRITE es complementaria de la proposición READ. En la misma forma en que la instrucción READ prevee la entrada de datos a la máquina, la proposición WRITE permite que los valores numéricos de la variable —los resultados— se incluyen en la salida, en una impresora de renglones o algún otro dispositivo periférico. Gran parte de lo que hemos dicho para la proposición READ se aplica exactamente en la misma forma a la instrucción WRITE

WRITE (m , n) lista

m = una constante entera o una variable que hace referencia a un dispositivo de salida

- n = número de proposición FORMAT
- lista = una secuencia ordenada de uno o más nombres de variables separadas por comas



se lee de la siguiente manera :

"Escribe por medio de la unidad 6 (en este caso la impresora) y de acuerdo con el formato número 100 lo que hay en las localidades de memoria (variables) XP, XS, J, A, JIM, VAL.

Es muy importante entender que lo que se escribe es el valor que tiene asignada la variable.

El código de unidad de salida especifica qué unidad de salida va a recibir estos valores. En nuestro ejemplo es la unidad 6 (impresora). Los nombres en la lista indican cuáles localidades de memoria (variables) van a proporcionar los valores de salida, y la proposición 100 FORMAT controla cómo estos valores van a ser escritos.

Los números asignados como número de referencia a los dispositivos de salida dependen del sistema de computación en uso. Los siguientes son dispositivos de salida que pueden ser disponibles :

1. Máquina de escribir eléctrica ó consola
2. Impresora
3. Perforadora de tarjetas
4. Perforadora de cintas
5. Cinta magnética
6. Disco magnético

7. Tubo de rayos catódicos

La mayor parte de las impresoras de renglones permiten que se impriman hasta 120 caracteres en una sola línea. Otras permiten 132, 136 etc. Si la lista de variables es muy grande se imprimirán automáticamente líneas adicionales y cada línea conforme a el formato referido.

Ejemplo :

Supóngase que en un problema dado se tienen las variables XA1, XA2 que adquirieron los valores 768.68 y -948.67 respectivamente, la instrucción de salida sería :

```
WRITE (6 , 109)XA1 , XA2
```

se lee : escribe por la impresora (6) de acuerdo con el formato 109 lo que hay en las localidades de memoria XA1 y XA2

```
768.680 ..... -948.67
```

Una de las diferencias que existen entre la proposición WRITE y READ es que en el primero se encuentra, a veces, una lista vacía, o sea, sin variables. Ejemplo :

```
WRITE (3 , 6)
```

Esto significa que, no hay variables impresas de salida, pero deseamos imprimir en la salida algún encabezado o alguna otra información pertinente para el programa.

PROPOSICION FORMAT :

Cuando se estudió las proposiciones de entrada y salida (READ y WRITE), se asoció con cada instrucción una proposición llamada FORMAT y que nos sirve para indicar los lugares que le corresponden a cada variable en la entrada o la salida.

Antes de seguir adelante debemos recordar que hay dos formas básicas de constantes : enteras y reales. En las enteras no se expresa el punto decimal, mientras que en las reales sí. Además descubrimos que, en FORTRAN, los números reales pueden expresarse en una de dos formas distintas. En la forma normal ó en forma exponencial.

Antes de seguir adelante, asegúrate de que estés familiarizado con esa forma de notación, de lo contrario repasa la unidad 3.

Puesto que hay tres modos posibles de representar los números, la máquina deberá recibir cierta información relativa al método que se utiliza en cualquier momento dado.

La proposición FORMAT es una proposición no ejecutable que se escribe

$$n \text{ FORMAT } (s_1, s_2, s_3, \dots, s_m)$$

donde

n = un número de proposición al que hace referencia una proposición de E/S (entrada/salida)

s_1, s_2, \dots, s_m = una lista ordenada de especificaciones de formato.

Las proposiciones FORMAT poseen la cualidad singular de poder localizarse en cualquier parte dentro del programa fuente, siempre y cuando se utilicen de manera apropiada. Como excepción a esta regla, existe el hecho de que una proposición FORMAT no puede colocarse en una localización del programa donde sólo pueden existir proposiciones ejecutables. Un procedimiento muy usado, es colocar los FORMAT enseguida de la proposición de READ o WRITE que hacen referencia a ella.

El lenguaje FORTRAN permite el uso de los siguientes tipos de especificaciones de formato:

FORMATO "I".

Este formato sirve tanto para entrada (READ) como para salida (WRITE). Se emplea para leer y escribir valores de variables enteras.

Su forma general es I_w . Donde I indica que se usa para variables enteras y w es un número entero que nos indica el número de lugares (ancho del campo) que puede usar la variable ya sea en la entrada o en la salida.

Al seleccionar el ancho del campo se debe considerar que el signo negativo ocupa un lugar.

Así por ejemplo si se desea asignar el valor -1985 a la variable M, las instrucciones serían:

```
      READ (5, 8) M
      8  FORMAT (I5)
```

I por que la variable (M) es entera.

5 porque se requieren cinco lugares (-1985).

En el caso de la lectura y escritura de datos no se tienen problemas al considerar el formato ya que sabemos los números que se manejan.

En el caso de escritura de resultado en donde no se sabe el número que se imprimirá se pueden tener problemas si el ancho del campo no es suficiente para que quepa el número en cuestión.

Por ejemplo, si se supone que una variable entera IP1 adquiere un valor de -589481 el valor que se tiene como salida dependerá del formato empleado.

NUMERO	FORMATO	SALIDA
-589481	I7	<u>- 5 8 9 4 8 1</u>
-589481	I10	<u> - 5 8 9 4 8 1</u>
-589481	I6	# # # # # #
no escribe el		
-589481	I5	# # # # #

Si el campo es menor que el número, escribe signos (#) que indican que el formato fue insuficiente.

Si un campo es mayor que el número, la parte sobrante de la izquierda contiene blancos.

Formato F.

Este formato sirve tanto para entrada como para salida. Se emplea para leer y escribir valores de variables reales.

Su forma general es Fw,d

En donde F indica que se refiere a variables reales.

w = ancho total del campo

d = número de cifras después del punto decimal.

Al seleccionar el ancho del campo se debe tener en cuenta que tanto el signo como el punto decimal ocupan lugar.

Así un formato como F8.3 servirá un número máximo de 9999.999 y un número mínimo de -999.999.

En el uso de lectura y escritura de datos no hay problema para seleccionar formato, sin embargo en la escritura de resultados es muy difícil predecir el tamaño del valor de salida por lo que se recomienda especificar un ancho de campo amplio.

En los casos en que el ancho del campo no es suficiente para el número se reporta lo anterior por medio de asteriscos que es una señal para modificar el formato empleado.

Formato E.

Sabemos que un número real se puede representar en forma exponencial, por ejemplo:

NUMERO	FORMA EXPONENCIAL
175	0.175×10^3
0.0016	0.16×10^{-2}
-631.48	-0.63148×10^3
151.94	0.1519×10^3
- 0.00148	-0.148×10^{-2}

El formato E tiene la forma general Ew,d, donde E indica que es exponencial.

w = ancho del campo

d = número de cifras decimales

Así los números mostrados si se usara el formato E se tendrían:

<u>NUMERO</u>	<u>FORMATO</u>	<u>FORTRAN</u>
175	E 10.3	<u>0.175E 03</u>
0.0016	E 9.2	<u>0.16E-04</u>
-631.48	E 12.5	<u>0.63148E 04</u>
151	E 10.3	<u>-0.148E-02</u>

La forma general de un número usando formato exponencial es :

$$\underline{+}0.\underline{XXXX}E\underline{+XX}$$

Podemos observar que se requieren

2 lugares para el exponente

1 lugar para el signo del exponente

1 lugar para la letra E

1 lugar para el punto decimal

1 lugar para el cero

1 lugar para el signo del número

Ø sea que independientemente el número de cifras decimales deseados en la mantisa se requieren de 7 lugares.

De lo anterior se concluye que el ancho w deberá cumplir con la siguiente ecuación :

$$\underline{w = d + 7}$$

Este formato es empleado principalmente en la salida de resultados ya que se pueden tener números muy pequeños y muy grandes (exponentes de -77 a 78).

Ejemplos :

En un problema determinado se tienen las siguientes variables XP1, XP2, RIP. Y se les desea asignar valores de acuerdo con la siguiente tabla :

$$\begin{aligned} \text{XP1} &= 75.86 \\ \text{XP2} &= -678.61 \\ \text{RIP} &= 9467.784 \end{aligned}$$

Hacer las instrucciones necesarias para leer los datos.

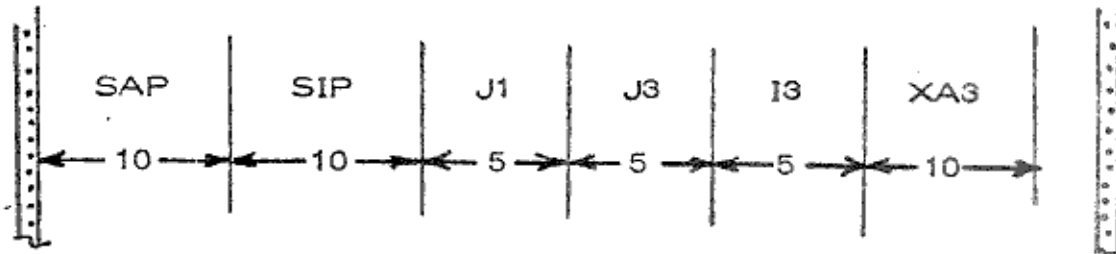
```

      6
      || READ (5, 8)XP1, XP2, RIP
      8 || FØRMAT (F5.2, F7.2, F8.3)
  
```

y la tarjeta de datos aparecería como :

y en la hoja de salida se tendría lo siguiente :

Lugares para imprimir el valor de



En la figura anterior se muestran los lugares que se les están asignando para imprimir los números que hay en las distintas localidades de memoria de la lista de la instrucción WRITE. Como en este caso no sabemos que números hay (supuestamente son resultados de un problema) se tienen 2 posibilidades y son :

1a. Que realmente los números en las localidades de memoria quepan en los lugares asignados, por ejemplo si

$$\text{SAP} = -156.75$$

$$\text{SIP} = 78.48$$

$$\text{J1} = 651$$

$$\text{J2} = -1581$$

$$\text{I3} = 98196$$

$$\text{XA3} = 758.91$$

y en este caso se tendría la siguiente salida :



Nuevamente se observa que lo que se imprime es una serie de números y en algunos casos se pueden confundir con otro como es en el caso del número 3 y del 4 que corresponden a las variables J1 y J2 respectivamente.

2a. Que el número de lugares asignados a alguna de las variables no sea suficiente para guardar el número, en este caso se marcan asteriscos y eso nos indica que debemos ampliar los lugares asignados a las variables.

Ejemplo 4.

Hacer el formato para escribir la siguiente lista de variables, asignándole 15 lugares a cada variable con 6 cifras decimales.

WRITE (6, 10)X1, X2, X3, X4, X5, X6, X7, X8, X9, X10, X11, X12, X13, X14
X15, X16, X17, X18, X19, X20.

Puesto que en cada fila hay 132 lugares, solamente caben 8 variables en cada línea y el formato podría ser:

10 FØRMAT (8F15.6) y la hoja de salida sería:

Lugares para imprimir los valores de

X1	X2	X3	X4	X5	X6	X7	X8
X9	X10	X11	X12	X13	X14	X15	X16
X17	X18	X19	X20				

En el orden mostrado escribirá lo que hay en las localidades mostradas.

10 FØRMAT (8F15.6) indica que en cada fila se usan 8 lugares de 15 columnas, cada uno para cada variable.

Para la primera fila, las primeras 8 variables

Para la segunda fila, las segundas 8 variables y

Para la tercera fila, las últimas 4 variables.

Formatos H y X.

Si en un problema determinado, se tiene un conjunto de variables que adquieren valores después de ciertos cálculos y suponemos que sean como sigue:

$$X1 = -108.671$$

$$X2 = 16.987$$

$$I3 = 65$$

$$J1 = -189$$

XMAY = 1981.98
 XMEN = - 98.0

Si se desea imprimir los valores de las variables anteriores, usaríamos el postulado WRITE.

Así si tuvieramos las siguientes instrucciones :

```

6
  WRITE (6, 10)X1, X2, I3, J1, XMAY, XMEN
19  FØRMAT (2F10.4, 2I5, 2F10.4) se tendría como salida

```

```

-108.671      16.987      65      -189      1981.98      -98.0 .....

```

Es decir sólo aparecen los valores adquiridos por las variables. Desde luego - que la información presentada en esta forma es muy deficiente ya que sólo la persona que hizo el programa puede interpretar los resultados y en el caso de que - haya un número considerable de resultados aun el programador tendría dificultades para interpretarlos.

Supóngase que la salida que se desea es de la siguiente forma :

```

X1 = -108.671   X2 = 16.987   I3 = 65   J1 = -189   XMAY =
1981.98   XMEN = -98.0   .....

```

Es decir se dejan los mismos lugares para los números pero además se desea - dejar lugares en blanco para separación e imprimir rótulos como X1 =, X2 =, I3 =, J1 =, XMAY = y XMEN =

Lo anterior lo podemos obtener con las siguientes instrucciones :

$$10 \left[\begin{array}{l} 6 \\ \text{WRITE (6, 10)X1, X2, I3, J1, XMAY, XMEN} \\ \text{FORMAT (3X, 3HX1=, F10.4, 3X, 3HX2=, F10.4, 3X, 3HI3=, I5, 3X,} \\ \text{3HJ1=, I5, 3X, 5HXMAY=, F104, 3X, 5HXMEN=, F10.4)} \end{array} \right.$$

Observaciones:

La instrucción `WRITE (6, 10)X1, X2, I3, J1, XMAY, XMEN`

es exactamente igual a la instrucción cuando no se buscaba poner r tulos.

La instrucci n de formato comienza con 3X lo que significa que pase 3 lugares sin imprimir nada.

Contin a con 3HX1= lo que significa que imprima el r tulo X1= (El 3H es por que X1= est  formado de 3 caracteres).

En general el formato X va precedido de un n mero entero positivo y significa que se pase ese n mero de lugares sin imprimir nada.

Por ejemplo:

5X, significa que pase 5 lugares sin impresi n

10X, significa que se pasen 10 lugares sin impresi n

El formato H va precedido de un n mero entero y significa que se impriman los siguientes caracteres de H indicados por el n mero.

Por ejemplo:

$$5H _ _ XP = _$$

imprimir a un blanco, enseguida `XP=`, y finalmente un blanco

Si se quisiera por ejemplo imprimir solamente un r tulo como `LOS DATOS DEL PROBLEMA SON` y adem s que dicho r tulo estuviera en el centro de la hoja.

Soluci n:

Suponiendo que se pueden imprimir 132 caracteres por l nea, como el r tulo `LOS DATOS DEL PROBLEMA SON` ocupa 26 lugares incluyendo los espacios de separaci n, se deberan tener

$$\frac{132 - 26}{2} = 53$$

lugares en blanco a cada lado del rótulo y las instrucciones serían

```

6 |
9 | WRITE (6 , 9)
  | FORMAT (53X , 26H LØS DATØS DEL PRØBLEMA SØN)

```

Obsérva que en la instrucción de WRITE (6 , 9) no va seguida de ninguna variable y es que no se va a imprimir el valor de ninguna variable, sino un rótulo.

Por la misma razón en la instrucción del formato no aparece ni formato I, F o E sino solamente el formato X y el H.

El mismo resultado que se obtiene usando el formato H se puede obtener encerrando entre apóstrofes aquellos que se desea sean impresos.

El ejemplo visto podría quedar en la forma :

```

6 |
10 | WRITE (6 , 10)X1 , X2 , I3 , J1 , XMAY , XMEN
   | FORMAT (3X , 3HX1= , F10.4 , 3X , 3HX2= , F10.4 , 3X , 3HI3= , I5 , 3X ,
   |         3HJ1= , I5 , 3X , 5HXMAY= , F10.4 , 3X , 5HXMEN= , F10.4)

```

Se puede poner en la forma :

```

6 |
10 | WRITE (6 , 10)X1 , X2 , I3 , J1 , XMAY , XMEN
   | FORMAT (3X , 'X1=' , F10.4 , 3X , 'X2=' , F10.4 , 3X , 'I3=' , I5 , 3X , 'J1=' ,
   |         I5 , 3X , 'XMAY=' , F10.4 , 3X , 'XMEN=' , F10.4)

```

O sea que ' _____ ' significa que se imprima lo que hay entre los apóstrofes.

(En algunas computadoras en lugar de apóstrofe se usan comillas " _____ ").

Se usan diagonales (✓) para indicar cambio de renglón.

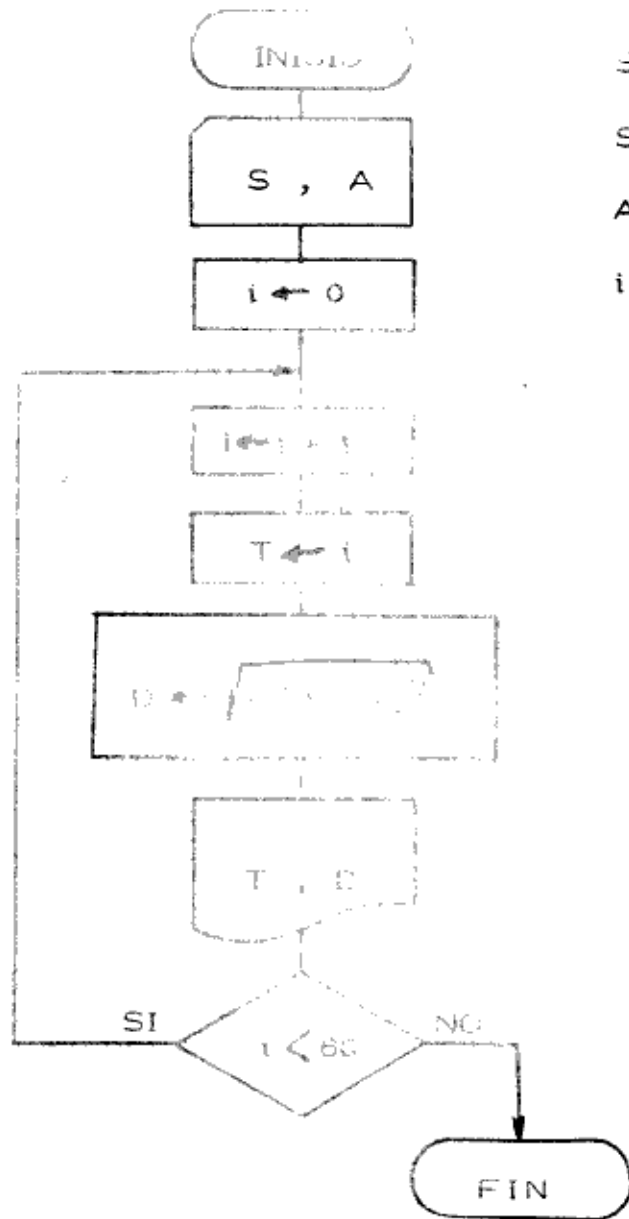
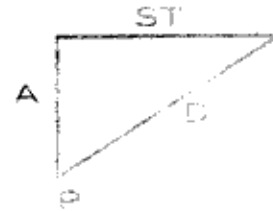
NOTA : La H.P. acepta las dos formas ' ___ ' y " ___ ".

Ejercicio :

Codificar con formato los ejercicios de la unidad anterior. Modificar y codificar los siguientes problemas en los que se da una solución que no es la óptima.

1. Un aeroplano que vuela a una altura A pasa directamente sobre un punto P. Si su velocidad es S, calcular su distancia requerida del punto P en los tiempos $T = 1, 2, 3, \dots, 60$ después del paso.

$$D = \sqrt{A^2 + (ST)^2}$$



St:

S = 0.10,

A = 3.0 ,

i = 0 , 1 , 2 , ...

1 , 2

3.0018 , 3.0018

C
C
C
C
C
C
C

COMENTACION I

RESUMTO OLFA MANUEL

*
* UNIDAD:VII PROGRAMA EXAMEL *
* GPO:SAI-25 *
*

**PROGRAMA PARA CALCULAR PUNTOS DE CONTROL
LA DESCRIPCION DEL PROBLEMA ES PARTE DE LA
SOLUCION DEL PROGRAMA **

```
WRITE(6,1)
WRITE(6,2)
WRITE(6,3)
WRITE(6,4)
WRITE(6,5)
WRITE(6,6)
WRITE(6,7)
WRITE(6,8)
WRITE(6,9)
WRITE(6,10)
WRITE(6,11)
WRITE(6,12)
WRITE(6,13)
WRITE(6,14)
WRITE(6,15)
WRITE(6,16)
WRITE(6,17)
WRITE(6,18)
WRITE(6,19)
WRITE(6,20)
WRITE(6,22)
WRITE(6,28)
WRITE(6,31)
WRITE(6,32)
WRITE(6,33)
WRITE(6,34)
WRITE(6,35)
WRITE(6,36)
WRITE(6,35)
WRITE(6,37)
WRITE(6,38)
WRITE(6,39)
WRITE(6,39)
WRITE(6,40)
WRITE(6,41)
WRITE(6,42)
WRITE(6,43)
WRITE(6,44)
WRITE(6,45)
DO 100 I=1,4
WRITE(6,44)
100 CONTINUE
WRITE(6,46)
DO 200 J=1,3
WRITE(6,51)
200 CONTINUE
WRITE(6,52)
DO 300 K=1,8
WRITE(6,47)
```

```
100 CONTINUE
WRITE(6,46)
DO 200 J=1,3
WRITE(6,51)
200 CONTINUE
WRITE(6,52)
DO 300 K=1,8
WRITE(6,47)
```


- 3.0 FORMULA
- 4.01 (6, 43)
- 5.01 (6, 47)
- 6.01 (6, 47)
- 7.01 (6, 47)
- 8.01 (6, 47)
- 9.01 (6, 47)
- 10.01 (6, 47)
- 11.01 (6, 47)
- 12.01 (6, 47)
- 13.01 (6, 47)
- 14.01 (6, 47)
- 15.01 (6, 47)
- 16.01 (6, 47)
- 17.01 (6, 47)
- 18.01 (6, 47)
- 19.01 (6, 47)
- 20.01 (6, 47)
- 21.01 (6, 47)
- 22.01 (6, 47)
- 23.01 (6, 47)
- 24.01 (6, 47)
- 25.01 (6, 47)
- 26.01 (6, 47)
- 27.01 (6, 47)
- 28.01 (6, 47)
- 29.01 (6, 47)
- 30.01 (6, 47)
- 31.01 (6, 47)
- 32.01 (6, 47)
- 33.01 (6, 47)
- 34.01 (6, 47)
- 35.01 (6, 47)
- 36.01 (6, 47)
- 37.01 (6, 47)
- 38.01 (6, 47)
- 39.01 (6, 47)
- 40.01 (6, 47)
- 41.01 (6, 47)
- 42.01 (6, 47)
- 43.01 (6, 47)
- 44.01 (6, 47)
- 45.01 (6, 47)
- 46.01 (6, 47)
- 47.01 (6, 47)
- 48.01 (6, 47)
- 49.01 (6, 47)
- 50.01 (6, 47)
- 51.01 (6, 47)
- 52.01 (6, 47)
- 53.01 (6, 47)
- 54.01 (6, 47)
- 55.01 (6, 47)
- 56.01 (6, 47)
- 57.01 (6, 47)
- 58.01 (6, 47)
- 59.01 (6, 47)
- 60.01 (6, 47)
- 61.01 (6, 47)
- 62.01 (6, 47)
- 63.01 (6, 47)
- 64.01 (6, 47)
- 65.01 (6, 47)
- 66.01 (6, 47)
- 67.01 (6, 47)
- 68.01 (6, 47)
- 69.01 (6, 47)
- 70.01 (6, 47)
- 71.01 (6, 47)
- 72.01 (6, 47)
- 73.01 (6, 47)
- 74.01 (6, 47)
- 75.01 (6, 47)
- 76.01 (6, 47)
- 77.01 (6, 47)
- 78.01 (6, 47)
- 79.01 (6, 47)
- 80.01 (6, 47)
- 81.01 (6, 47)
- 82.01 (6, 47)
- 83.01 (6, 47)
- 84.01 (6, 47)
- 85.01 (6, 47)
- 86.01 (6, 47)
- 87.01 (6, 47)
- 88.01 (6, 47)
- 89.01 (6, 47)
- 90.01 (6, 47)
- 91.01 (6, 47)
- 92.01 (6, 47)
- 93.01 (6, 47)
- 94.01 (6, 47)
- 95.01 (6, 47)
- 96.01 (6, 47)
- 97.01 (6, 47)
- 98.01 (6, 47)
- 99.01 (6, 47)
- 100.01 (6, 47)

```

D, 2X, 14H! <- - 11 - GATJ H.)
39 FORMAT(19X, 1H., 6Y, 1H., 22X, 1H., 6X, 1H., 35X, 1H., 6Y, 1H., 3X, 2H!! , 3X, 1H!
D, 2Y, 1H! , 3X, 2H!! )
40 FORMAT(19X, 1H., 6Y, 1H., 22X, 1H., 6X, 1H., 2X, 17HTAPON OF CONCRETO, 16X, 1
H. 6X, 1H., 1Y, 1H! , 16X, 1H!)
41 FORMAT(19X, 1H., 6Y, 1H., 22X, 1H., 6X, 1H., 11X, 1H! , 23X, 1H., 6X, 1H., 2X, 16H
#-----)
42 FORMAT(19X, 1H., 6Y, 1H., 22X, 1H., 6X, 1H., 11X, 1H., 23X, 1H., 6X, 1H., 3X, 2H!
D! , 1X, 6H***** , 1X, 2H!! )
43 FORMAT(13X, 7H..... , 6Y, 7H..... , 10X, 7H..... , 6X, 20H.....
D..... , 10X, 7H..... , 6X, 9H..... , * , 6X, 6H*..... )
44 FORMAT(15X, 1H., 18X, 1H., 10X, 1H., 18X, 10H.//////// , 3Y, 1H., 10Y, 1H., 18
9Y, 2H. * , 5X, 24* , 5Y, 1H.)
45 FORMAT(15X, 1H., 18X, 7H. <- L.C. , 4X, 1H., 18X, 10H.//////// , 3X, 7H. <- L.C.
D, 4X, 1H., 10Y, 2H. * , 6Y, 2H. * , 5Y, 7H. <- L.C.)
46 FORMAT(12X, 5H-/- , 15X, 5H-/- , 6Y, 5H-/- , 17X, 10H. * * * * * , 2X, 3H-/- , 8Y,
D34-/- , 17Y, 21. x, 6X, 2H. * , 2Y, 5H-/- )
52 FORMAT(13X, 20H..... * * * * * ..... , 10X, 21H..... * , 6X
D, 5H*..... , 10Y, 21H..... * , 6X, 6H*..... )
47 FORMAT(19X, 1H*, 6Y, 1H*, 36X, 1H*, 6X, 1H*, 35X, 1H*, 6Y, 1H*)
48 FORMAT(19X, 1H*, 6Y, 1H* <- P L I F , 20X, 1H*, 6X, 9H* <- P I I U T E , 27X, 1H*, 6Y, 9H
D* <- P L I F )
49 FORMAT(19X, 1H*, 3Y, 4H* * * * , 36X, 1H*, 3X, 4H* * * * , 35X, 1H*, 3Y, 4H* * * * )
50 FORMAT(19X, 1H*, 2X, 5H* * * * , 36X, 1H*, 2X, 5H* * * * , 35X, 1H*, 2X, 5H* * * * )
51 FORMAT(19X, 3H* * * * , 41X, 3H* * * * , 40X, 3H* * * * )
53 FORMAT(13X, 1H., 18X, 1H., 10X, 1H., 18X, 2H. * , 6X, 2H* , 3X, 1H., 10X, 1H., 18X
1, 2H. * , 6X, 2H* , 3X, 1H.)
STOP
END

```

DATA UNIT 1414 20 PILLD

PILOTES DE CONTROL

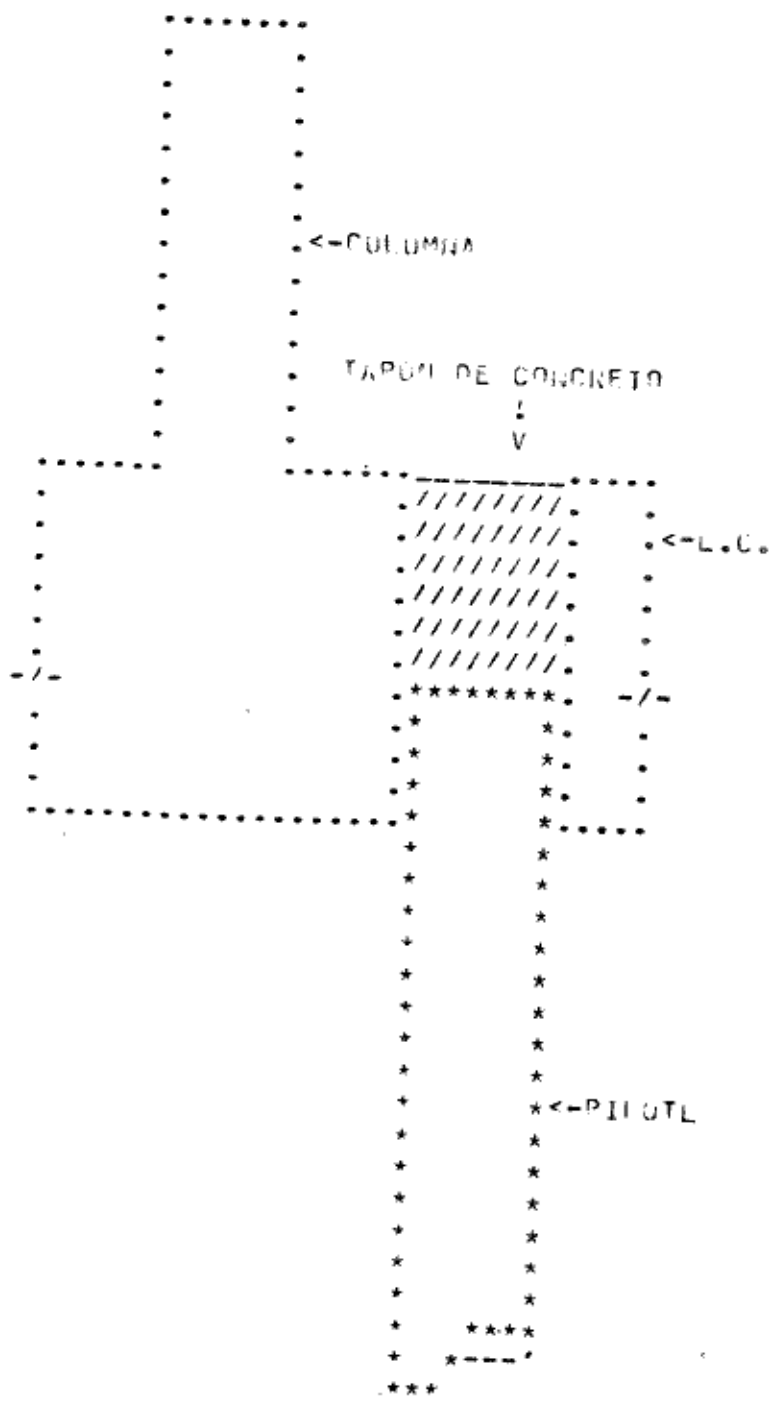
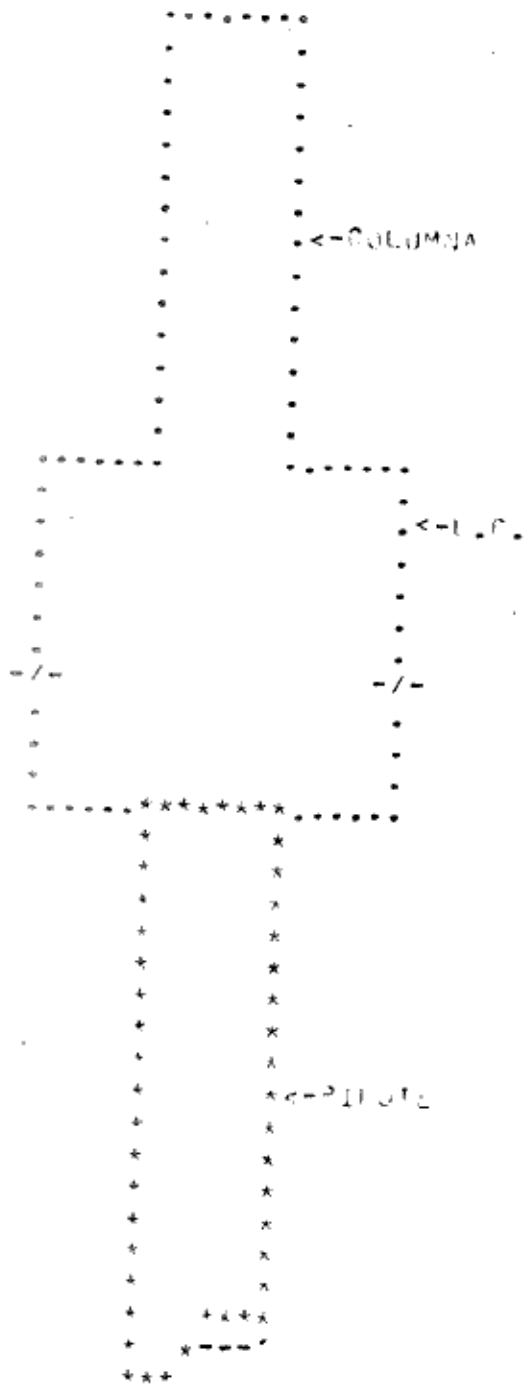
LOS PROBLEMAS DERIVADOS DE UN SUBSUELO DEBIL EN CONSTANTE PROCESO DE CONSOLIDACION, HA DADO LUGAR A LA CREACION DEL LLAMADO *PILOTE DE CONTROL* QUE SE PROYECTO PARA EVITAR QUE LOS EDIFICIOS SE QUEDARAN MAS ALTOS QUE LAS BANQUETAS Y JARDINES AL IR DESCENDIENDO LA CIUDAD DE MEXICO.

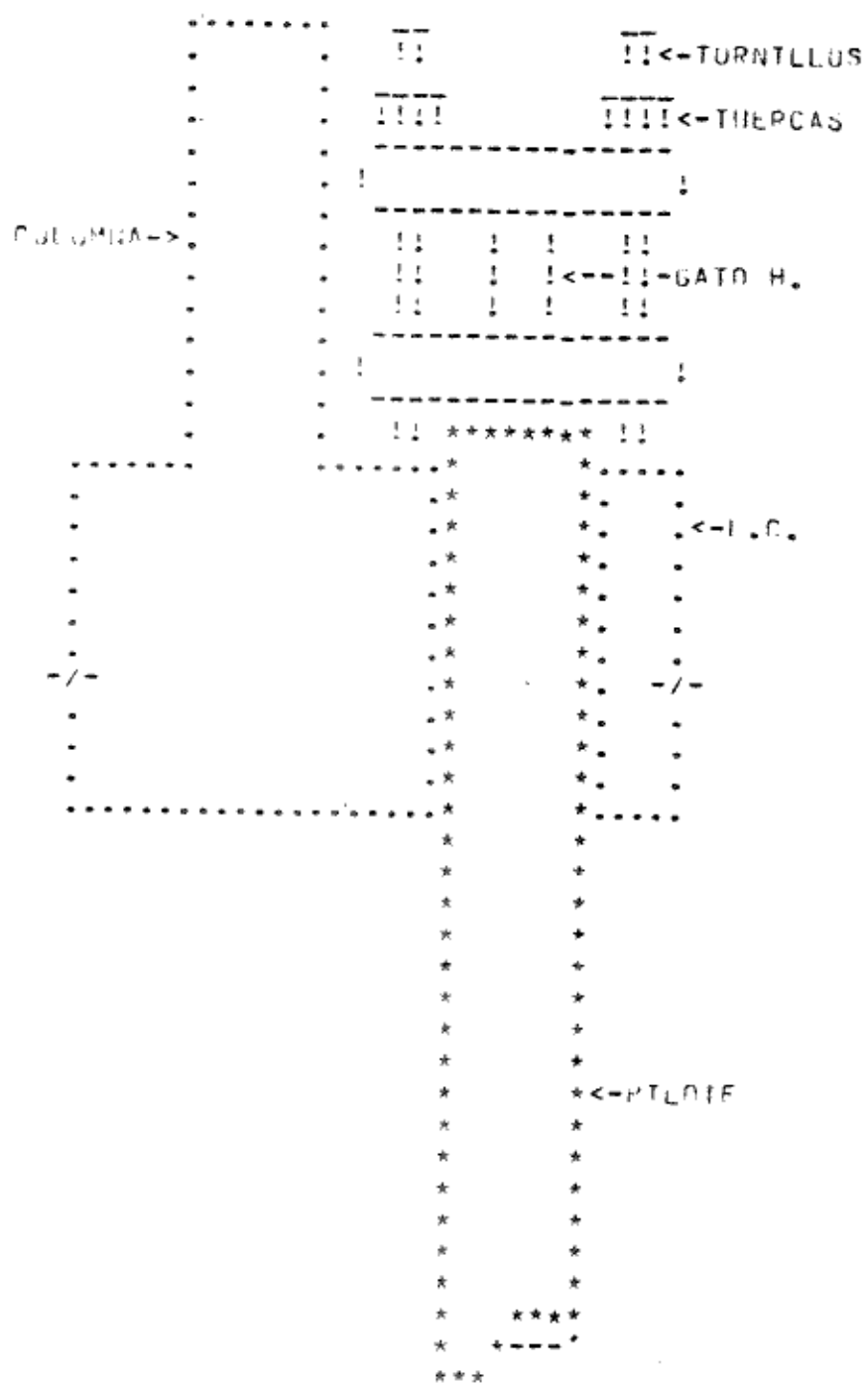
LOS PILOTES DE CONTROL NO VAN ARABO DE LAS COLUMNAS O LAS CONTRAPARES, SIÑO QUE ATRAVIEZAN LIBREMENTE LA LOZA DE CIMENTACION, QUEDANDO ACCESIBLES LAS CABEZAS DE LOS MISMOS PARA AUMENTARLES O DISMINUIRLES CARGAS, PARA SUBIR O BAJAR EDIFICIOS, NUEVOS O YA CONSTRUIDOS.

LOS DISPOSITIVOS DE CONTROL HAN SIDO CONCEBIDOS PARA APLICARSE EN LA CAREZA DE LOS PILOTES, SOMETIDOS A LA COMPRESION DE UN DETERMINADO MATERIAL O UN CATU HI - DRAULICO O BIEN TRANSFORMANDO ESA COMPRESION EN TENSION PARA HACER FLUIR VARILLAS. TAMBIEN SE HA PLANLADO USAR LA RESISTENCIA A LA FLEXION DE SOLERAS ADOSADAS AL PILOTE.

LA CARGA DEL EDIFICIO ES TRANSMITIDA A LOS FUSTES DE LOS PILOTES Y ES REGULADA POR FRICCION, EVENTUALMENTE SE PUEDE USAR LA CAREZA DEL PILOTE

* COLOCACION DE LOS PILOTES *





```

A
C ESTE PROGRAMA EXAMINA UNA LISTA DE ALUMNOS CON SUS RESPECTIVAS CALIFICACIONES. OBTIENE EL PROMEDIO DE LOS DOS PRIMEROS EXAMENES PARCIALES Y ESTE LO PROMEDIA CON LA CALIFICACION FINAL ESTABLECIENDO SI APRUEBA O NO. CALP1=CALIFICACION PARCIAL 1 CALP2 CALIFICACION PARCIAL 2 CALF=CALIFICACION FINAL CALD=CALIFICACION DEFINITIVA
C
C DOUBLE PRECISION NBOL
WRITE(6,3)
3 FORMAT(20X,'ALUMNO',10X,'NBOL',10X,'CALP1',10X,'CALP2',10X,'CALF',
-10X,'CALD',10X,'COMENTARIO'//)
DO 40 I=1,10
READ(5,*)NBOL,CALP1,CALP2,CALF
PRM1=(CALP1+CALP2)/2
CALD=(PRM1+CALF)/2
IF(CALD-7.)30,40,40
30 WRITE(6,7)I,NBOL,CALP1,CALP2,CALF,CALD
7 FORMAT(22X,I2,11X,I6,10X,F5.2,10X,F5.2,10X,F5.2,10X,'N
-ACREDITO'//)
GO TO 40
40 WRITE(6,9)I,NBOL,CALP1,CALP2,CALF,CALD
9 FORMAT(22X,I2,11X,I6,10X,F5.2,10X,F5.2,10X,F5.2,10X,'ACRE
*DITO'//)
40 CONTINUE
STOP
END

```

```

C ESTE PROGRAMA CALCULA EL VALOR DEL PERIODO CONOCIENDO LOS TIEMPOS
C T1 Y T2 QUE OSCILA (10 OSCILACIONES) UN SISTEMA DE DOS RESORTES
C SUJETOS EN UN EXTREMO A UNA MASA CONOCIDA, Y EN EL OTRO EXTREMO
C SUJETOS FIRMEAMENTE.
C ESTE PROGRAMA ES PARA MINIMOS CUADRADOS, DETERMINANDO LA PENDIEN-
C TE(TG), Y LA ORDENADA AL ORIGEN(R).
N= 10
S = 0
T = 0
U = 0
V = 0
WRITE (6,11)
11 FORMAT(7X,'MASA',4X,'T1',6X,'T2',8X,'TP',7X,'TCUAD')
DO 1 I=1,10
READ(5,15) X, T1, T2
15 FORMAT(5F10.5)
TP= (T1+T2)/2/10.
Y= TP**2
WRITE(6,16) X, T1, T2, TP, Y
16 FORMAT(5F10.5)
S= S+ X*Y
T= T+X
U= U+Y
V= V+X**2
1 CONTINUE
TG= (N*S -T*U)/(N*V - T**2)
R= (V*U - T*S)/(N*V - T**2)
WRITE(6,12) TG, R, S, T, U, V
12 FORMAT(/X,'EL VALOR DE LA PENDIENTE ES',F15.5, /
*,7X,'EL VALOR DE LA ORDENADA ES',F15.5, //, 'S=',F15.5, /,7X,'TV=',
*F15.5, /,7X,'U=',F15.5, /,7X,'V=',F15.5)
STOP
END

```

MASA	T1	T2	TP	TCUAD
100.00000	5.90000	6.20000	.60500	.36603
150.00000	7.40000	7.10000	.72500	.52562
200.00000	8.40000	8.00000	.82000	.67240
250.00000	9.00000	9.90000	.84500	.80103
300.00000	9.90000	10.00000	.99500	.99003
350.00000	10.70000	10.60000	1.06500	1.13423
400.00000	11.40000	11.50000	1.14500	1.31102
450.00000	12.00000	11.80000	1.19000	1.41610
500.00000	12.60000	12.50000	1.25500	1.57503
550.00000	13.30000	13.20000	1.32500	1.75563

EL VALOR DE LA PENDIENTE ES .00506

EL VALOR DE LA ORDENADA ES .06021

S= 4058.97018
TV= 3250.00000
U= 10.54710
V= 126250.00000

ALUMNO	GRUPO	CALP1	CALP2	CALF	CALD	COMENTARIO
1	76001	6.50	7.20	5.10	6.23	NO ACREDITO
2	76002	7.40	6.50	6.80	6.68	NO ACREDITO
3	76003	6.80	6.50	7.50	6.28	NO ACREDITO
4	76004	7.40	10.00	6.80	7.75	ACREDITO
5	76005	9.60	6.50	7.50	8.15	ACREDITO
6	76006	10.00	7.60	5.00	6.90	NO ACREDITO
7	76007	8.60	6.60	6.60	7.25	ACREDITO
8	76008	7.50	10.00	9.00	9.08	ACREDITO
9	76009	6.40	6.60	5.50	6.35	NO ACREDITO
10	76010	7.00	6.80	7.20	6.65	ACREDITO


```

00001000 C   PROGRAMA PARA MOSTRAR UN FORMATO
00002000 C   ORDONEZ MACIAS LUIS ANTONIO
00003000 C   COMPUTACION I
00004000 C   PROGRAMA DE EXAMEN
00005000 C
00006000 C LUIS8
00007000 C234567
00008000     WRITE(6,1)
00009000     1 FORMAT(X,"GUIA DE DICTADO",/,70(H_),/,13X,"CASSETTE Y LADO",8X,"
00010000     *FECHA",11X,"NUMERO",/,70(H_),/,X,"TIPO",45X,"OTRO")
00011000     WRITE(6,2)
00012000     2 FORMAT(X,"PAPEL",7X,"MEMBRETADO",4X,"MEMO",4X,"PERSONAL",/
00013000     *,70(H_),/,X,"DIRIGIDO",/,4X,"A:",/,70(H_))
00014000     WRITE(6,3)
00015000     3 FORMAT(X,"COPIAS",45X,"COPIA",/,X,"PARA:",45X,"RECORDATORIO
00016000     *",/,70(H_),/,X,"ASUNTO",50X,"NTA.REF.",/,70(H_),/,X,"EXTENSION")
00017000     WRITE(6,4)
00018000     4 FORMAT(2X,"APROX",4X,"1/2 HOJA",4X,"1 HOJA",4X,"2 HOJAS"
00019000     *,4X,"MAS DE DOS HOJAS",/,70(H_),/,X,"CONCEPTOS E IDEAS A TRATAR EN
00020000     * CADA PARRAFO",/,70(H_))
00021000     DO 5 J=1,4
00022000     5 WRITE(6,6)J
00023000     6 FORMAT(70(H_),/,70(H_),/,X,11,69(H_),/,70(H_),/)
00024000     WRITE(6,7)
00025000     7 FORMAT(70(H_),/,X,"CONCLUSION",/,70(H_),/,X,"INSTRUCCIONES ESPE
00026000     *CIALES Y ANEXOS",/,70(H_),/,70(H_))
00027000     STOP
00028000     END

```

81300112

PROGRAM UNIT MAIN' COMPILED

GUIA DE DICTADO

CASSETTE Y LADO		FECHA	NUMERO
-----------------	--	-------	--------

TIPO PAPEL	MEMBRETADO	MEMO	PERSONAL	OTRO
------------	------------	------	----------	------

DIRIGIDO A:

COPIAS PARA: COPIA RECORDATORIO

ASUNTO NTA. REF.

EXTENSION APROX 1/2 HOJA 1 HOJA 2 HOJAS MAS DE DOS HOJAS

CONCEPTOS E IDEAS A TRATAR EN CADA PARRAFO

- 1
- 2
- 3
- 4

.....
CONCLUSION

.....
INSTRUCCIONES ESPECIALES Y ANEXOS