

## DESCRIPCION DE LA TARJETA. PROPOSICIONES DE CONTROL. PROGRAMAS SIMPLES.

### INTRODUCCION.

Normalmente, las computadoras utilizan como un medio cintas de papel o tarjetas perforadas para recibir instrucciones, y presentan las respuestas en forma impresa en papel, junto con cualquier otra información que necesite el programador. En esta unidad vamos a ocuparnos de las tarjetas perforadas como medio de entrada, suponiendo que se obtendrá la salida de la computadora por la impresora de alta velocidad.

Cuando hablamos de diagramas de flujo (en la primera unidad), recordarán que teníamos dos tipos básicos de símbolos que se unen por medio de líneas. Los llamamos símbolos de decisión y de proceso. Los símbolos de proceso pueden contener varias instrucciones (proposiciones aritméticas). En esta unidad vamos a ocuparnos de las representaciones en FORTRAN del contenido de los símbolos que sirven para tomar decisiones. Esas decisiones controlan efectivamente el flujo de operaciones lógicas del programa.

Aquí también estudiaremos problemas cortos y trataremos ya de utilizar la computadora. La experiencia que se gane suministra una buena base para la comprensión de los programas que se presentarán en las demás unidades.

### OBJETIVOS.

Al terminar esta unidad deberás estar capacitado para:

1. Explicar con tus propias palabras el número de campos en que se divide la tarjeta y para que se usan.
2. Saber usar la perforadora.
3. Explicar para qué sirven las proposiciones de control IF ARITMETICO, IF lógico y GØ TØ.
4. Expresar la forma general de IF ARITMETICO y GØ TØ.
5. Indicar cómo se expresan estas instrucciones en la tarjeta perforada.

6. Dado un problema sencillo expresado en palabras, haciendo uso de las proposiciones estudiadas, hacer diagrama de flujo y programa para la computadora.
7. Aplicar todos los conocimientos adquiridos hasta ahora en la resolución de problemas sencillos.

#### PROCEDIMIENTO DE APRENDIZAJE.

1. Estudiar el anexo de la unidad.
2. Acudir al centro de cómputo para aprender el uso de la perforadora y el procedimiento para mandar correr un programa.
3. Si hay alguna duda, acudir a asesoría ó consultar algún libro de FORTRAN que contenga el tema.
4. Consultar el manual de FORTRAN 3000 de la H.P.

REQUISITO: Antes de presentar el examen de esta unidad, deberás entregar la autoevaluación resuelta.

#### AUTOEVALUACION.

1. Especifica las funciones que desarrolla la unidad de entrada.
2. Dado el siguiente conjunto de instrucciones, hacer el listado de acuerdo con los lugares que le corresponden en la tarjeta. En las hojas de codificación.

#### C PROGRAMA PARA RESOLVER UNA ECUACION DE SEGUNDO GRADO.

```

READ (5, *)A, B, C
WRITE (6, *) 'A=', A, 'B=', B, 'C=', C
IF (B**2-4.0*A*C) 10, 11, 12
10X 1R=-B/(2.0*A)
X1I=SQRT (4.0*A*C - B**2)/(2.0*A)
X2R=X1R
X2I=X1I
WRITE (6, *) 'X1R=', X1R, 'X1I=', X1I, 'X2R=', X2R, 'X2I=', X2I
'DISCRIMINANTE NEGATIVO'
GO TO 20
11X 1R=B/(2.0*A)
X2R=X1R
WRITE (6, *) 'X1R=', X1R, 'X2R=', X2R
GO TO 20

```

```

12 X 1R = (-B + SQRT (B**2 - 4.*A*C))/2.0*A)
X2R = (-B - SQRT (B**2 - 4.0*A*C))/(2.0*A)
WRITE (6, *) 'X1R=', X1R, 'X2R=', X2R
20 STOP
END

```

- \*3. Del listado anterior, desarrolla el diagrama de flujo y perfora las tarjetas.
- \*4. Forma tu lote de tarjetas (tarjetas de control, tarjetas del programa, tarjeta de control para la entrada de datos, tarjeta de datos) y presentalo en la ventanilla del centro de cómputo para que tu programa sea procesado.
5. Usando tus propias palabras escribe como se leen las instrucciones IF y GO TO que se encuentran en el listado 2.
6. En los siguientes casos indica si la instrucción es correcta ó no, en caso de no ser correcta indicar porque.

```

6
|
| .
| .
| .
| GO TO 80
| .
| .
| .
| .
| .
| .
80 | X = A + B

```

```

6
|
| 10 | .....
| .
| .
| IF (SIN (X) - X) 10, 20, 10
| .
| .
| .
| .
| .
| .
20 | .....

```

```

6
|
| 10 | .....
| .
| .
| IF (X.GT.0) GO TO 20
| X = X + 1
| GO TO 10
| .
| .
20 | .....

```

```

6
|
| .
| .
| .
| IF (A - B .EQ. X)
| Y = A + B
| .
| .
| .

```

\*Para desarrollar los puntos 3 y 4, consulta la GUIA DE USUARIO PARA EL SISTEMA BATCH/3000 editado por la Coordinación de Servicios de Computo, UAM, 1984. Todos los programas que se desarrollen en el curso, se efectuarán directamente desde una terminal. Leer la última hoja de la Unidad, página 102. y 101

## MATERIAL ANEXO.

Descripción de la Tarjeta :

¿Qué es una tarjeta perforada y cómo debe utilizarse? Esa tarjeta, tal y como lo implica su nombre, es simplemente una hoja delgada de cartón, que tiene por lo común un tamaño de 18.8 x 8.3 cm.; aún cuando sean posible otros tamaños. - Esas tarjetas pueden apilarse unas sobre otras para formar un "lote". Las tarjetas puede "leerlas" la computadora, que interpreta cada una de las proposiciones (cada proposición requiere una tarjeta separada) que se han perforado en cada tarjeta. Dichas tarjetas se leen por la parte superior del lote. Por ende, la primera tarjeta que se lee en un lote será la que se encuentra arriba y la última, la que esté al final de la pila.

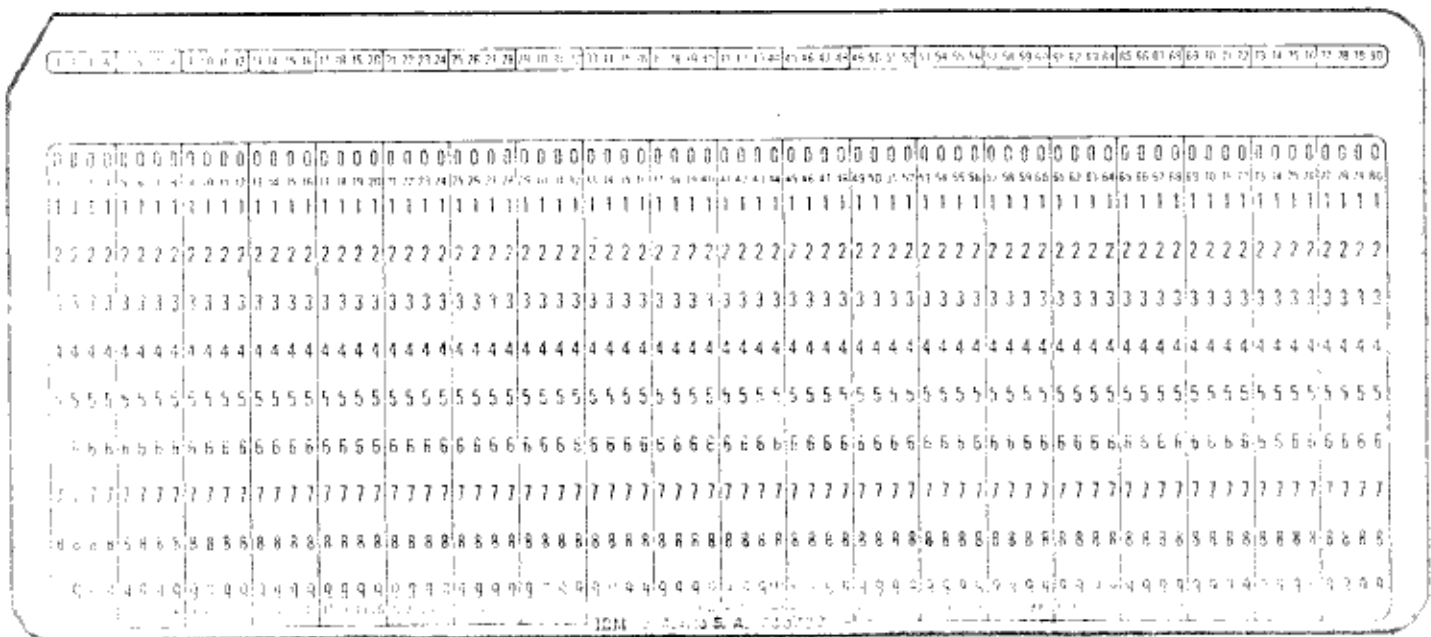


Fig. 1

En la fig. 1 se presenta una de esas tarjetas. Como podrán ver, está dividida, - en sentido longitudinal, en ochenta columnas numeradas; en cada columna se - puede poner un símbolo (letra, número, u otro caracter). Estas 80 columnas se dividen en campos (grupos de columnas).

Campo Principal. Está formado por el conjunto de columnas de la 7 a la 72 inclusive, y se reserva para la proposición FORTRAN.

Campo de identificación. Columnas de la 73 a la 80. La información contenida en este campo no es ejecutada por el computador, es decir, el computador no lee o procesa la información en este campo. Por lo tanto, el programador lo usa para ayudarse en la identificación de la tarjeta.

Campo para asignar dirección a proposiciones. Está formado por las columnas 1-5, son campos reservados para los números de proposiciones cuando se usan. Puesto que las tarjetas se procesan en orden de aparición, no todas las tarjetas necesitan números de proposición. Cuando se usan los números de proposición se pueden asignar arbitrariamente.

DOS TARJETAS NO PUEDEN TENER  
EL MISMO NUMERO DE PROPOSICION

El número de proposición debe ser un entero positivo entre 1 y algún límite (9999) que satisfice a todos los compiladores.

Columna de continuación. Columna 6. Se usa cuando una proposición es tan larga que necesita más de una tarjeta. Cada tarjeta adicional requiere de un carácter diferente de cero en la columna 6. Normalmente se emplean dígitos del 1 al 9; pero caracteres alfabéticos también se emplean.

Columna de comentarios. Es importante para un programador poder saber en un momento dado lo que hace un programa, y en los programas complicados que consisten de un número grande de instrucciones y variables, poner en claro, en el mismo listado, el significado de las variables. Lo anterior se puede conseguir con tarjetas de comentarios.

Una tarjeta de comentario se puede emplear colocando una "C" en la columna 1. Esta informa al compilador que la tarjeta es una tarjeta especial y no se procesará. En la columna 1 se escribe una "C", se deja un blanco y enseguida se escribe el comentario que se desea imprimir.

Además de las ochenta columnas dispuestas en su sentido longitudinal, las tarjetas estándar tienen doce hileras, de arriba abajo. Diez de esas hileras están numeradas de cero a nueve. Hay otras dos, denominadas, a veces, hileras 11 y 12, aún cuando por lo común, no van marcadas en esa forma. Se consideran renglones de zona los designados como 0, 11 y 12 y como renglones de dígitos los 10 restantes incluyendo el 0, que es común para ambos.

Cuando se desea perforar números del 0 al 9, se usan solamente los renglones de dígitos, perforando aquel que corresponda con el número deseado.

Los caracteres especiales tales como + ) ( . , = etc., no siguen un orden definido, sin embargo, la mayoría tiene 3 perforaciones por columna.

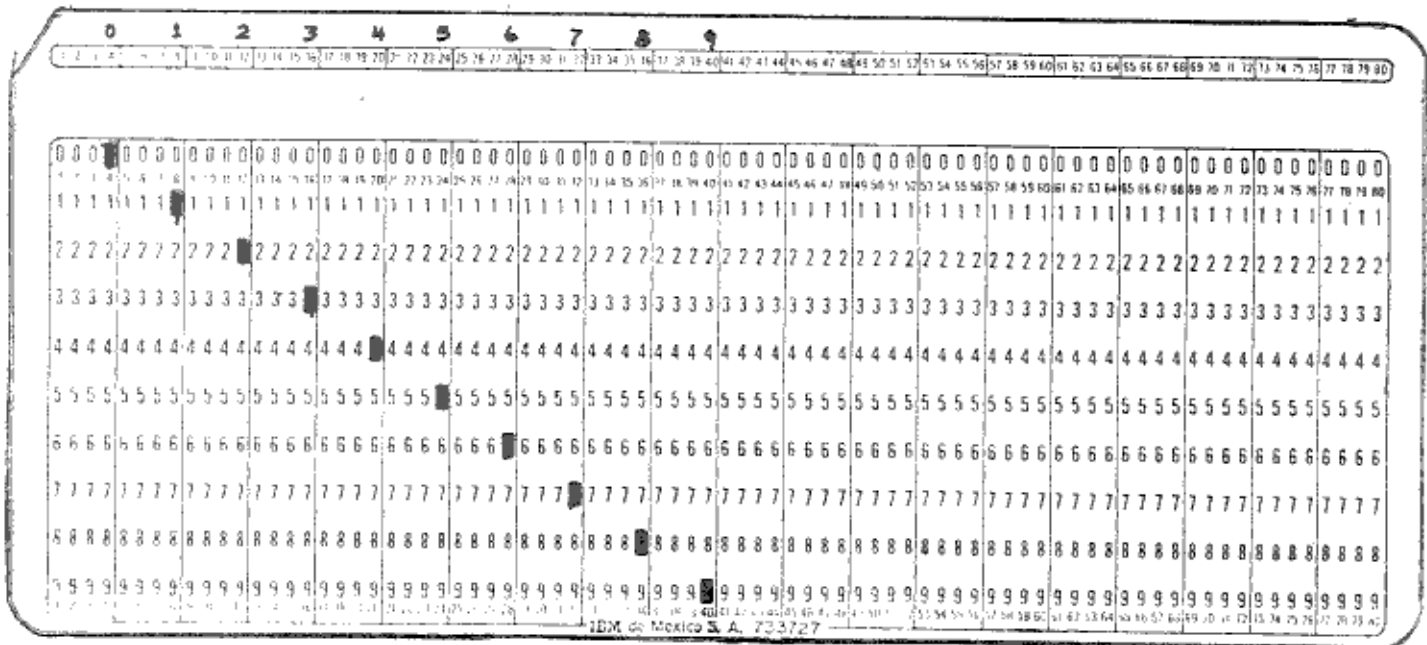
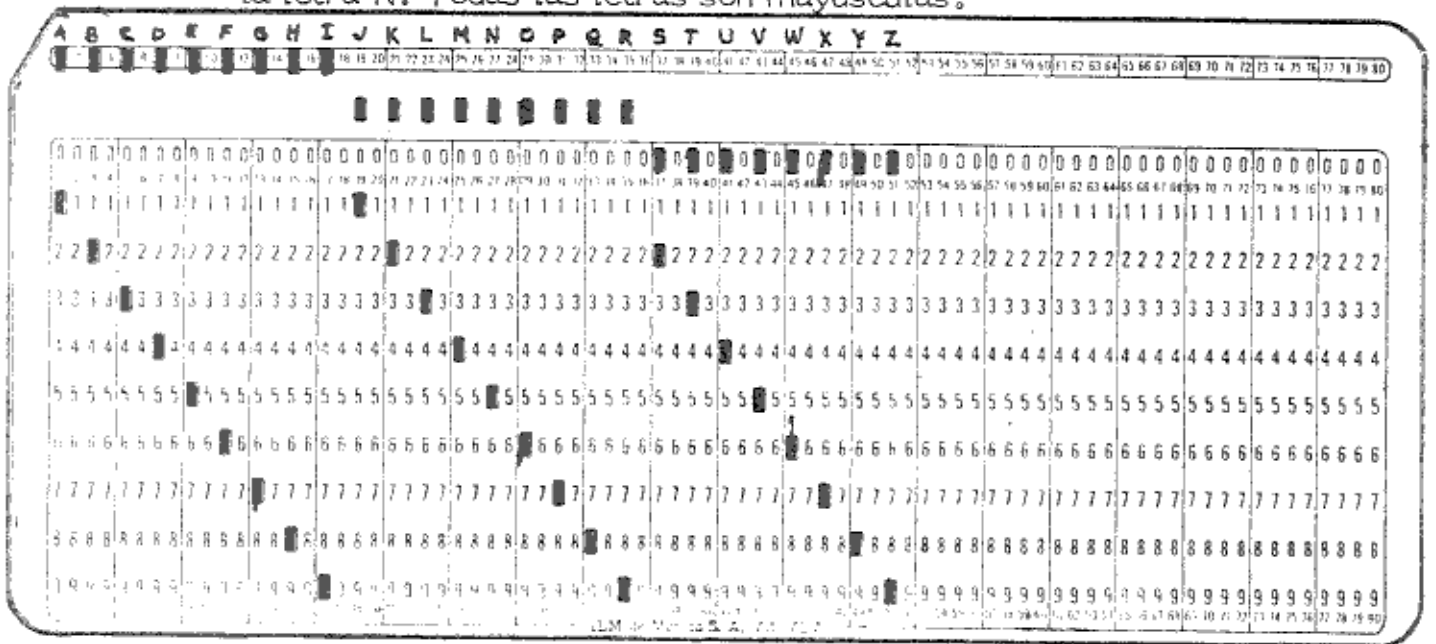


Fig. 2

Cuando se desea perforar letras se usan combinaciones entre los renglones de - ZONA y de DIGITOS siguiendo la siguiente regla :

- a. Zona 12 con los dígitos del 1 al 9 son las letras de la A a la I respectivamente.
- b. Zona 11 con los dígitos del 0 al 9 son las letras de la J a la R.
- c. Zona 0 con los dígitos del 2 al 9 son las letras de la S a la Z.

NOTA: No existen las letras dobles (CH, LL, RR) en una sola columna, ni la letra Ñ. Todas las letras son mayúsculas.



Es necesario aclarar que la máquina perforadora tiene un teclado como el de cualquier máquina de escribir, la cual perfora automáticamente los orificios en los renglones respectivos de acuerdo con la tecla seleccionada.

Lo impreso en la parte superior de la tarjeta solamente es información visual para las personas, lo importante son las perforaciones que estén en ella.

### ALGUNAS PROPOSICIONES QUE NOS PERMITEN ESCRIBIR PROGRAMAS Y PROCESARLOS EN LA COMPUTADORA.

El conjunto de proposiciones FORTRAN colocado en orden adecuado constituye un programa fuente.

Las proposiciones FORTRAN se dividen en dos categorías: proposiciones de ejecución y proposiciones de no-ejecución.

Las proposiciones de ejecución son aquellas que resultan en alguna instrucción que el computador debe llevar a cabo. Una vez que el programa esté procesado.

Las proposiciones de no-ejecución son aquellas que proveen información necesaria para la ejecución adecuada de un programa.

Entrada y salida de datos en forma libre. - son proposiciones ejecutables.

Para la entrada de datos, simplemente se requiere una instrucción READ (5, \*) y los nombres de las variables separados por comas.

Ejemplo:

```

| 67
| READ (5, *) ITEM, X, Y

```

los valores de las variables se escriben en una tarjeta (tarjeta de datos) separados por comas, si:

```

ITEM = -160
X = 11.38
Y = -.32 E 06

```

estas dos instrucciones se perforan de la siguiente manera:





Segunda forma :

```
ITEM = -160  X = 0.1138 E 02  Y = 0.3200 E 06
```

Proposición STOP : es una proposición de ejecución que se escribe

```
STOP
```

hace que la computadora pare, causa la terminación del programa que se está ejecutando. Puede aparecer varias veces en el programa, inmediatamente la computadora se va a la última proposición.

Proposición END : es una proposición de no ejecución que se escribe

```
END
```

define el final de un programa fuente y debe ser la última proposición en un programa.

Ejemplo :

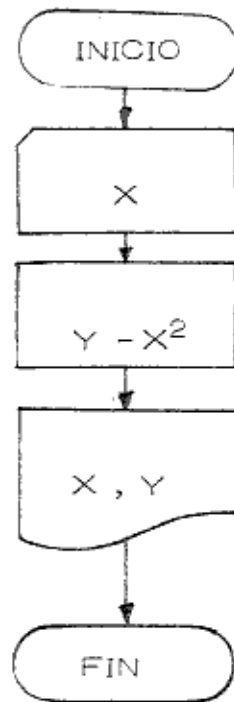
Todo lo aprendido hasta ahora debe permitirnos escribir programas cortos que podemos procesar en la computadora. Es necesario hacer notar ciertos aspectos.

- Las proposiciones que hemos estudiado deben saberse perfectamente.
- En la elaboración del algoritmo y diagramas de flujo es importante la creatividad, se debe diseñar la solución, ésta no se obtiene después de aplicar una serie de reglas.

- c. Para resolver un problema, se pueden seguir algunas recomendaciones útiles como: analizar detalladamente el enunciado del problema. No iniciar el algoritmo si no se ha entendido perfectamente el problema. Recuerda que la computadora no te resuelve el problema si no lo sabes resolver tú.

1. La ecuación de una parábola  $Y = X^2$

Diagrama de flujo:





Centro de Estudios

UNIVERSIDAD AUTÓNOMA METROPOLITANA

HOJA DE CALIFICACION FORTRAN

0072

LINEA	FECHA	COMENTARIOS
1		JOB SAI, PO442/J0CELYN.A0310P; TIME=40
2		:COMMENT UNIDAD SAI
3		:FORT60
4		\$CONTROL USLINIT
5		C CALCULO DE LA PARABOLA Y=X**2
6		C ESTE PROGRAMA EVALUA LA FUNCION Y TAMBIEN ILUSTR EL USO DE VARIAS
7		C METODOS DE IMPRESION
8		C READ(S,*)X
9		C CALCULAR EL VALOR DE Y
10		C Y=X**2
11		C ESCRIBIR LOS VALORES DE X, Y
12		C WRITE(6,*)X, Y
13		C ESCRIBIR LOS VALORES DE X, Y, IDENTIFICANDO CADA CIFRA
14		C WRITE(6,*)X', Y', Y
15		C STOP
16		C END
17		8E0D
18		2.5
19		8E0J

TARJETAS DE CONTROL

TARJETAS DE CONTROL

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15

87

```

SCUATROL USLE11
1 C CALCULO DE LA PARABOLA Y=AX**2
2 C ESTE PROGRAMA EVALUA LA FUNCION Y TAMBIEN ILUSTRAN EL USO DE VARIOS
3 C VALORES DE IMPRESION
4 C
5 C CALICULA EL VALOR DE Y
6 C Y=AX**2
7 C
8 C MOSTRA LOS VALORES DE X,Y
9 C
10 C MOSTRA LOS VALORES DE X,Y IDENTIFICANDO CADA CIFRA
11 C
12 C STOP
13 END

```

PROGRAM UNIT NAME: COMPILE

PAGE 0002 HEWLETT-PACKARD 301020.01.01 FONTANA/3000 THU, SEP 27, 1979, 11:17 AM

```

**** SEVERAL STATISTICS ****
**** NO ERRORS OR WARNINGS ****
TOTAL COMPILE TIME 000001
TOTAL TAPED TIME 000002

```

END OF COMPILE  
END OF PREPARE

```

2.51000 4.20000
XB 2.00000 YE 0.25000

```

```

END OF PROGRAM
END
CPU SEC. = 0. ELAPSED TIME = 1. THU, SEP 27, 1979, 11:17 AM

```

Observaciones :

Después de las tarjetas de control se tienen :

- 1º Tres tarjetas de comentarios que nos explican que hace el programa. Observa que en cada instrucción, en la primera columna va una C.
- 2º La instrucción READ (5, \*)X. Esta instrucción indica que se le debe asignar un valor a la variable X. El valor de X se encuentra en la tarjeta 14.
- 3º Una proposición aritmética  $Y = X**2$ . Significa que a la variable Y (real) se le asigne el valor que resulte de elevar al cuadrado la variable X.
- 4º Siguen una serie de instrucciones WRITE. Nos indican que se deben imprimir los valores de las localidades de memoria (variables) X y Y en el orden señalado.
- 5º STOP y END. Sirven para ordenar a la computadora que ha terminado el programa.

Ejercicio :

Corre un programa semejante a éste con la ecuación  $Y = \text{sen}(x)$  con el siguiente comentario :

ESTE PROGRAMA EVALUA EL SENO DE UN ARGUMENTO POR MEDIO DE LA FUNCION SENO. TAMBIEN ILUSTRA EL USO DE VARIOS METODOS DE IMPRESION.

—USO DE LA COMPUTADORA PARA TOMAR DECISIONES.—

En el ejemplo anterior usamos la computadora para llevar a cabo varios cálculos. Si ésta fuera su única función, entonces la computadora no sería otra cosa que una poderosa calculadora (o regla de cálculo). La utilidad de la computadora se hace más poderosa por su habilidad para tomar decisiones; habilidad disponible por medio del uso de proposiciones de transferencia condicional.

Las proposiciones de transferencia se clasifican de dos maneras : transferencias incondicionales y transferencias condicionales.

Cada vez que se encuentre una proposición de transferencia incondicional, durante el flujo de control normal del programa, se produce una transferencia o salto que fue claramente especificado al momento de escribir el programa fuente.

Cada vez que se lleva a cabo una proposición de transferencia condicional se produce una transferencia que se escoge, dentro de varias alternativas, durante el proceso de ejecución del programa objeto.

Las proposiciones de transferencia incondicional son:

GØ TØ

CALL

RETURN

Las proposiciones de transferencia condicional son:

IF aritmético

IF lógico

GØ TØ computado

GØ TØ asignado

Para nuestros fines, sólo estudiaremos aquí el IF aritmético, IF lógico y el GØ - TØ computado, en otras unidades nos ocuparemos de estudiar los demás.

PROPOSICION GØ TØ es una proposición ejecutable que se escribe.

GØ TØ n

donde n es un número de proposición (una constante entera).

Esta proposición hace que el computador transfiera control (es decir, tome la siguiente instrucción) a la proposición n que se menciona. Una vez que la transferencia ha ocurrido, el computador continúa con la ejecución secuencia normal de instrucciones a partir del punto a donde fue transferido. Observa que la proposición a la cual se hace la transferencia puede localizarse en cualquier parte del programa.

Ejemplo: GØ TØ 98

Significa:

La siguiente instrucción que se va a ejecutar es la instrucción número 98. Que se puede encontrar antes o después de la proposición GØ TØ

98	6	.
		.
		.
		.
		.
		GØ TØ 98
		.
		.
		.
		.

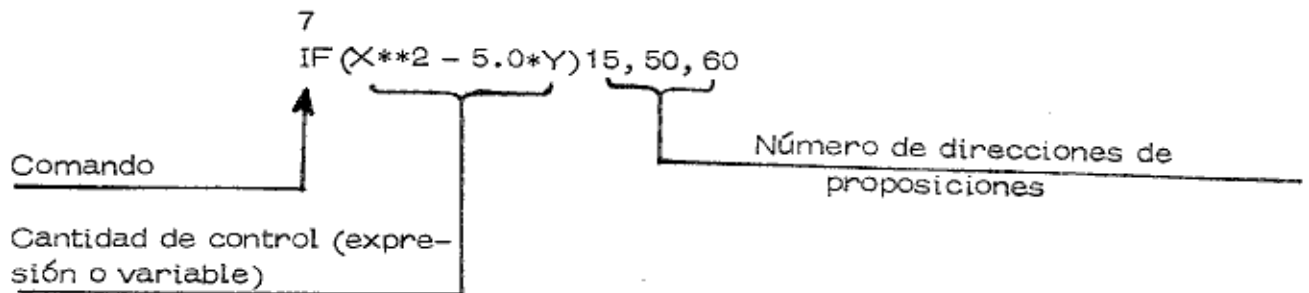
o bien

	6	.
		.
		.
		GØ TØ 98
		.
		.
98		.

Claramente cuando se tiene una proposición  $G\emptyset T\emptyset$  referido a alguna instrucción, dicha instrucción debe estar numerada. Si por ejemplo en un programa tenemos  $G\emptyset T\emptyset 150$  y no existe en el programa ninguna instrucción con el número 150, - se marcará error, o por el contrario si existen dos o más instrucciones con la dirección 150, también se marcará error.

PROPOSICION IF ARITMETICO. Es una proposición de control que nos permite transferir el control del programa a una de tres instrucciones. Esta transferencia depende del valor numérico de una cantidad (variable o expresión aritmética) contenida en paréntesis y que forma parte de la proposición.

Ejemplo :



- Si  $X**2 - 5.0*Y$  es menor que cero la siguiente instrucción que se ejecutará a la número 15.
- Si  $X**2 - 5.0*Y$  es igual a cero, la siguiente instrucción que se ejecutará es la número 50.
- Si  $X**2 - 5.0*Y$  es mayor que cero la siguiente instrucción que se ejecutará es la número 60.

Se pueden poner expresiones como la siguiente :

IF (A) 5, 5, 8                      significa :

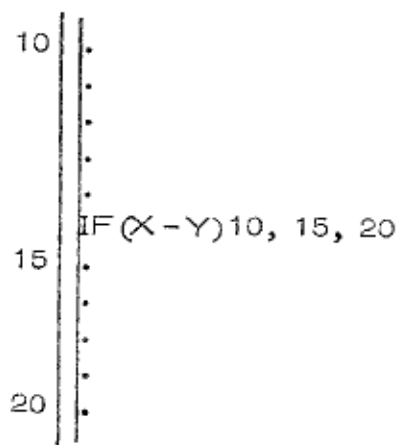
Si A es menor ó igual que cero, la siguiente instrucción que se ejecutará es la número 5.

IF (X\*\*2 - Y\*\*2) 3, 4, 4

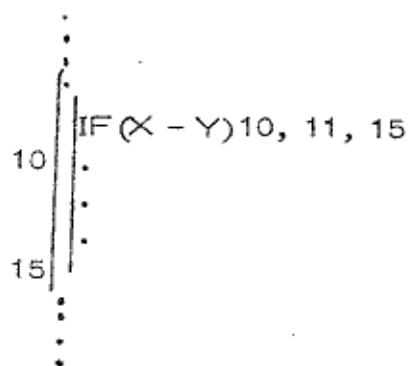
IF (S1) 6, 7, 8

IF (A \* SIN(X) - B \* COS(X)) 15, 20, 15

En seguida se muestran algunas formas que se pueden emplear para el IF aritmético.

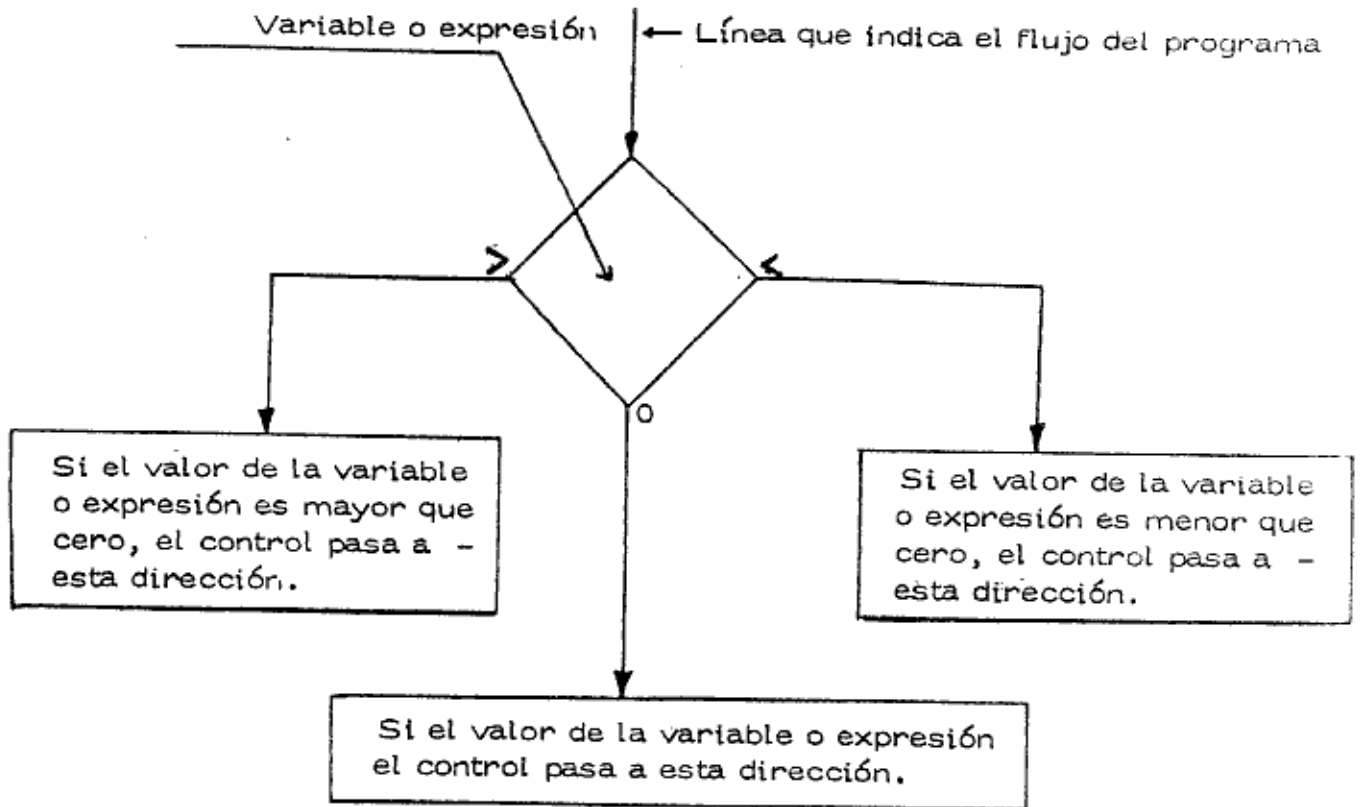


Una de las 3 direcciones debe seguir a la instrucción IF. Las 2 restantes pueden estar en cualquier parte del programa.





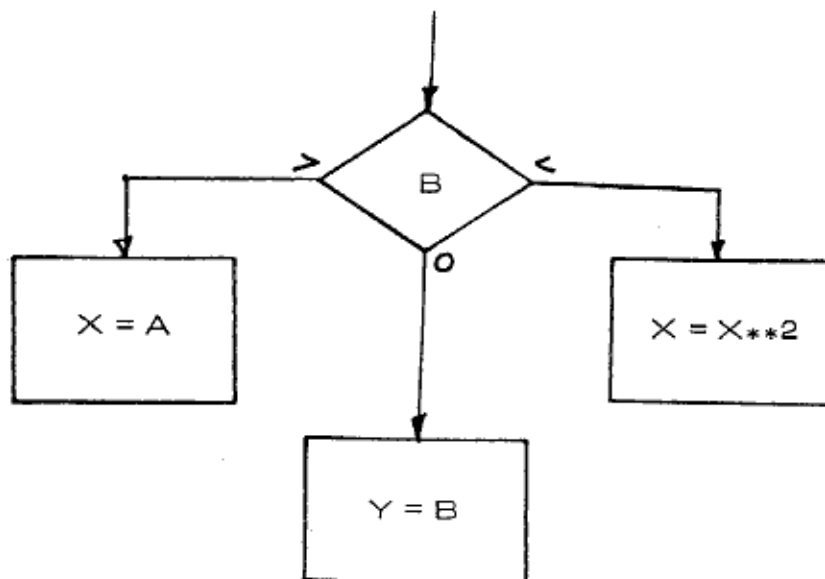
La representación en el diagrama de flujo es como sigue :



Ejemplo :

Dados los siguientes diagramas de flujo, hacer la instrucción FØRTRAN equivalente.

a)



Dicha expresión indica:

Si  $B > 0$  entonces la siguiente instrucción es  $X = A$ .

Si  $B = 0$  entonces la siguiente instrucción es  $Y = B$ .

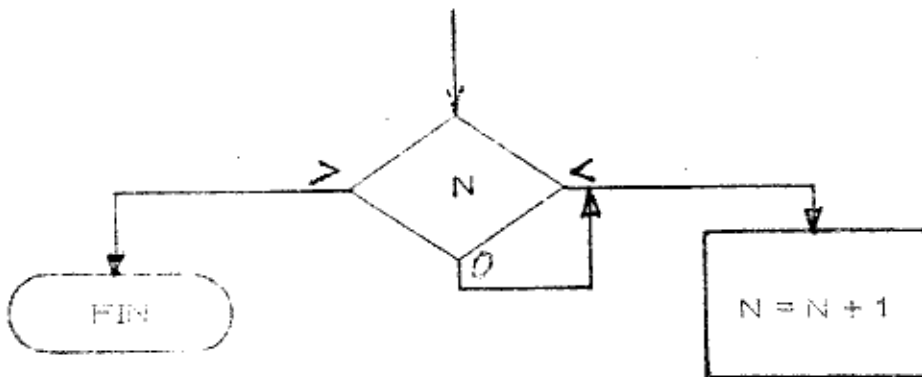
Si  $B < 0$  entonces la siguiente instrucción es  $X = X**2$ .

de donde la instrucción en FØRTRAN será:

```

30 | IF(B)10, 20, 30
    | X = P
    | .
    | .
    | .
20 | Y = B
    | .
    | .
    | .
10 | X = X**2
  
```

b)



La expresión se lee: Si  $N$  es menor ó igual que cero, la siguiente instrucción es  $N = N + 1$ .

Si  $N$  es mayor que cero, la siguiente instrucción es  $FIN$ .

El postulado FØRTRAN será:

```

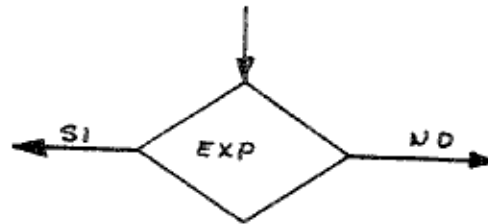
10 | IF(N)10, 10, 20
    | N = N + 1
    | .
    | .
    | .
20 | STOP
  
```

**PROPOSICION IF LOGICO.** Es una proposición ejecutable que utiliza una expresión lógica (en cambio de una expresión aritmética) para tomar decisiones correspondientes a la transferencia de control. Sólo tiene dos salidas, su forma general es :

```

| IF (exp) prop 1
| prop 2

```



Si es verdadero, se ejecuta la prop 1 si es falso, se ejecuta a prop 2

donde *exp* es una expresión lógica y *prop 1* y *prop 2* son proposiciones ejecutables, en FORTRAN. Existen restricciones en la proposición 1, éstas son que : - *prop 1* no puede ser un IF lógico, ni un DØ.

Para poder utilizar el IF lógico es necesario conocer los operadores de relación; se usan para hacer comparaciones entre los valores numéricos de dos expresiones aritméticas, la tabla siguiente describe estos operadores :

<u>Descripción</u>	<u>Símbolo Algebraico</u>	<u>Símbolo en FORTRAN</u>
igual a	=	.EQ.
no igual a	≠	.NE.
menor a	<	.LT.
menor o igual a	≤	.LE.
mayor a	>	.GT.
mayor o igual a	≥	.GE.

Ejemplo :

```

| IF (X.LT.Y) Z = SQRT (Y - X)
| P = A + Z
|
| prop 2          prop 1

```

Ejemplo :

Si se tienen las siguientes instrucciones :

```

1 | IF (X - 6) 1, 4, 4
  | Y = 1.05 * X
  | GO TO 10
4 | IF (X - 12.) 2, 3, 3
2 | Y = X + 1.36
  | GO TO 10
3 | Y = -1.67 * X + 33.4
10 | WRITE (6, *) 'X=', X, 'Y=', Y

```

Estas proposiciones pueden reemplazarse por las siguientes :

```

  | IF (X.LT.6.) GO TO 1
  | IF (X.LT.12) GO TO 2
  | Y = -1.67 * X + 33.4
  | GO TO 10
1 | Y = 1.05 * X
  | GO TO 10
2 | Y = X + 1.36
10 | WRITE (6, *) 'X=', X, 'Y=', Y

```

Ejercicio :

Pasar las dos formas a diagramas de flujo. Recuerda que el programa sigue una secuencia a menos que se le indique otra dirección.

Ejemplo de un programa.

Problema : Temperatura del agua para mezclas de concreto.

Durante el invierno (por debajo de  $40^{\circ}\text{F}$ ), el concreto se debe vaciar a una temperatura entre  $70 - 100^{\circ}\text{F}$  para evitar que la mezcla se congele antes de que el concreto se frague. El método más sencillo que usa el ingeniero para controlar la temperatura de la mezcla, es seleccionar la temperatura del agua que se añade. Un balance calorífico sencillo de todos los materiales que se añaden al concreto, hace posible obtener la siguiente expresión para la temperatura del agua.

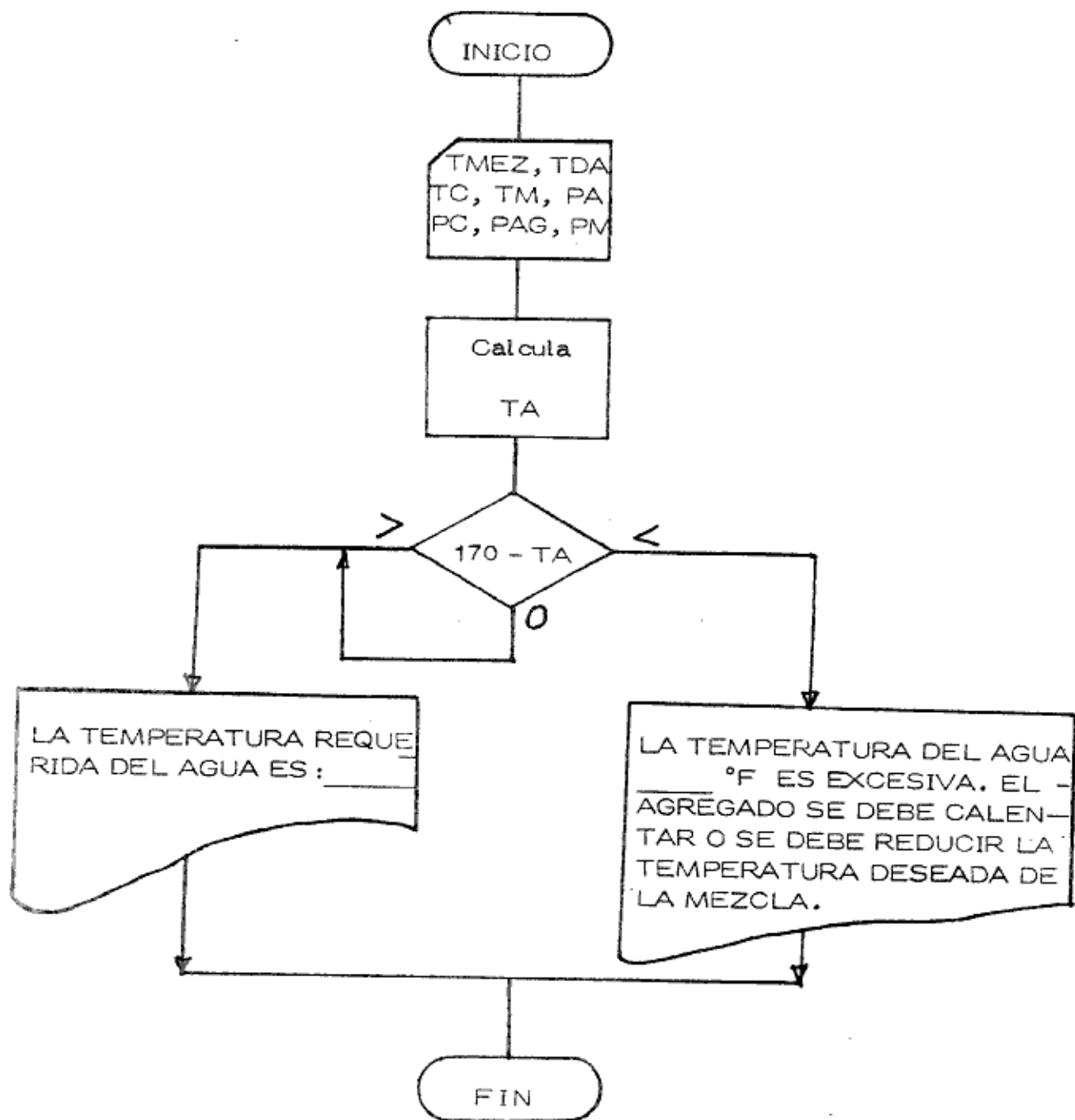
$$T_w = \frac{T_{mez.} (W_a + W_c)/5 + W_m + W_w) - (T_a W_a + T_c W_c)/5 - T_m W_m}{W_w}$$

Los símbolos usados en la fórmula se describen a continuación

Símbolo	Nombre	Unidades	
$T_w$	TA	Temperatura del agua usada en la mezcla	°F
	TMEZ	Temperatura deseada de la mezcla	°F
$T_a$	TDA	Temperatura del agregado seco	°F
$T_c$	TC	Temperatura del cemento seco	°F
$T_m$	TM	Temperatura del remojo agregado	°F
$W_a$	PA	Peso del agregado seco	lb
$W_c$	PC	Peso del cemento seco	lb
$W_w$	PAG	Peso del agua usada en la mezcla	lb
$W_m$	PM	Peso del remojo agregado	lb

La temperatura del agua calculada, se compara con 170°F que es la temperatura máxima permitida para el agua que se añade a la mezcla. Por encima de esta temperatura, la reacción química que produce la fraguación del cemento se lleva a cabo rápidamente produciendo el fenómeno de asentamiento rápido.

## a) Desarrollo del diagrama de flujo



El diagrama de flujo muestra que si la temperatura del agua sobrepasa el límite, entonces se escribe "un mensaje" descriptivo de la situación. Si la temperatura se encuentra en el intervalo permitido, entonces se imprime el valor calculado.

PROGRAMA PARA CALCULAR LA TEMPERATURA REQUERIDA DEL AGUA PARA MEZCLAR CONCRETO EN TIEMPO FRIO

SE LEEN TODAS LAS DATOS

READ(S,\*)TMEZ, TDA, TC, TM, PA, PC, PAG, PM

SE CALCULA LA TEMPERATURA DEL AGUA DE ACUERDO A LA FORMULA:

TA = (TMEZ \* ((PA + PC) / 5) + PM + PAG) - (TDA \* PA + TC \* PC) / 5 - TM \* PM / PAG

PRECONTAR SI LA MEZCLA ES MENOR O IGUAL A 170 OF

IF (170. - TA) 2, 3, 3

WRITE(6, \*) LA TEMP REQUERIDA DEL AGUA TA, ES EXESIVA. EL AGREGADO DEBE REDUCIR LA TEMP DE LA MEZCLA.

WRITE(6, \*) LA TEMP REQUERIDA DEL AGUA ES TA, OF

STOP

END

C PROGRAMA PARA CALCULAR LA TEMPERATURA REQUERIDA DEL AGUA PARA MEZCLAR

C CONCRETO EN TIEMPO FRIO

C SE LEEN TODAS LAS DATOS

C READ(S,\*)TMEZ, TDA, TC, TM, PA, PC, PAG, PM

C SE CALCULA LA TEMPERATURA DEL AGUA DE ACUERDO A LA FORMULA:

C TA = (TMEZ \* ((PA + PC) / 5) + PM + PAG) - (TDA \* PA + TC \* PC) / 5 - TM \* PM / PAG

C PRECONTAR SI LA MEZCLA ES MENOR O IGUAL A 170 OF

C IF (170. - TA) 2, 3, 3

C WRITE(6, \*) LA TEMP REQUERIDA DEL AGUA TA, ES EXESIVA. EL AGREGADO DEBE REDUCIR LA TEMP DE LA MEZCLA.

C WRITE(6, \*) LA TEMP REQUERIDA DEL AGUA ES TA, OF

C STOP

C END

85. , 40.0 , 55. , 40.1410 , 282. , 125. , 25.

SCOUTING SLUITS

C PREPASA PARA CALCULAR LA TEMPERATURA RESULTANTE DEL AGUA PARA MEZCLAR

E CUBIERTO EN HORAS FRIO

C SE USA TODOS LOS DATOS

20. (S+T) 102, 105, 10, 10, PA, PC, PAB, PB

C SE CALCULA LA TEMPERATURA DEL AGUA SE ADECUA A LA FORMULA:

$$T = ((T1 * V1) + (T2 * V2) + (T3 * V3)) / (V1 + V2 + V3)$$

C DESPUES DE LA MEZCLA SE REDUCE A 170 DE

1000. (S+T) 102, 105, 10, 10

2. (S+T) LA TEMP RESULTANTE DEL AGUA ES, TA, ES EXESIVA, EL AGREGADO

DEBE CALENTARLO SE DEBE REDUCIR LA TEMP DE LA MEZCLA

SI SE

3. (S+T) LA TEMP RESULTANTE DEL AGUA ES, TA, DE

SI SE

END

LA TEMP RESULTANTE DEL AGUA 216.520 ES EXESIVA, EL AGREGADO

DEBE CALENTARLO SE DEBE REDUCIR LA TEMP DE LA MEZCLA



### Ejercicios :

Desarrolla el diagrama de flujo y programa para computadora, perfóralo y córrelo, para los siguientes problemas :

1. A, B y C son valores numéricos. Escribir los valores en orden ascendente.
  
3. A, B son los coeficientes de la ecuación  $AX + B = 0$ .
  - a. Escribe los valores de A y B.
  - b. Calcula el valor de X.
  
3.  $A_1X_1 + B_1Y + C_1 = 0$  y  $A_2X + B_2Y + C_2 = 0$ , son las ecuaciones de dos rectas en el plano.
  - a. Leer  $A_1$ ,  $B_1$ ,  $C_1$ ,  $A_2$ ,  $B_2$ , y  $C_2$ .
  - b. Determinar si las rectas son paralelas o se cortan.
  - c. En caso de que las rectas se corten, encontrar el punto de intersección.
  - d. Imprimir los datos y los resultados.

### ENTRADA DE PROGRAMAS FORTRAN EN UNA TERMINAL.

El procedimiento empleado para introducir un programa por terminal son diferentes para cada sistema de computación. Por ahora usaremos el siguiente :

Después de leer la guía de usuario para el sistema batch/3000, hasta la página 4. El inciso 5 dice que debes de escribir tu archivo en FORTRAN. La pantalla te da un número de línea, y el cursor aparece en la columna 11 (que es la columna 1 para tu programa). Debes presionar el espaciador seis veces, escribir la proposición fortran y después oprimir la tecla RETURN para escribir otra proposición. Las mismas instrucciones que se siguen para perforar una tarjeta, se utilizan aquí.

Al terminar de teclear el programa (escribir STOP, END) se escribe la instrucción de control !EOD en la columna 1.

### Entrada de Datos.

Siempre que el programa lleve proposiciones READ, es necesario escribir los datos de las variables tal como aparecen en la instrucción READ y el formato correspondiente. Recuerden, el orden de los valores debe corresponder al de las variables y deben separarse por comas si es formato libre, empezando desde la columna 1.

En este momento, ya se tiene el programa completo en un archivo local. Para proceder a guardarlo sigue las instrucciones de la guía del usuario para el sistema Batch/3000.

Nota: Debes tener tu programa codificado y probado antes de sentarte a una terminal, ya que el tiempo que tienes para escribir tu programa en la terminal es limitado. Otra opción es el servicio que te ofrece la coordinación; dar tu codificación perfectamente escrita y en hoja de codificación para que ellos hagan el archivo y lo guarden en tu cuenta.