MATERIAL ANEXO

UNIDAD IX

PROPOSICIONES DE DECLARACION DE TIPO:

La proposición de declaración de tipo es una proposición de especificación no ejecutable que se usa en el lenguaje FORTRAN para definir explícitamente el modo de un dato almacenado en una variable de cualquier nombre.

Hay cinco formas generales para la proposición de declaración tipo, como sigue:

1. INTEGER $v_1, v_2, v_3, \ldots$
2. REAL $v_1, v_2, v_3, \ldots$
3. DOUBLE PRECISION $v_1, v_2, v_3, \ldots$
4. COMPLEX $v_1, v_2, v_3, \ldots$
5. LOGICAL $v_1, v_2, v_3, \ldots$

donde $v_1, v_2, v_3, \ldots$ es una lista de nombres de variables, o de nombres de arreglos con dimensiones.

Estas proposiciones de declaración preparan las localizaciones del almacenamiento que se nombran en la lista para recibir un tipo particular de datos. Sus funciones son parecidas a las funciones de la proposición DIMENSION ya que debe reservar cualquiera de las variables que se nombran en la lista.

En el caso de una variable que comience, por ejemplo, con la letra M, se considera todavía como entera, en el estado normal de las cosas. Decimos que, a falta de instrucciones contrarias, se aplican las "modalidades normales". Esto quiere decir que I, J, K, L, M y N indican variables enteras. Sin embargo, hay lugar para anular esas modalidades normales, cuando surja la necesidad de hacerlo. Por ejemplo, supongamos que tenemos una variable cuyo nombre es NCA; pero que deseamos que esa variable se considere como real.

Lo único que necesitaremos es una proposición de tipo del que sigue:

```
REAL NCA
```

Esto le indica a la computadora que la variable NCA, aunque normalmente es entera, debe considerarse, para los fines de este programa, como "real"

REAL ABLE, I, J(10)

indica que las localizaciones I y todos los elementos del arreglo J, que ordinariamente serían de modo entero, se van a usar para almacenar valores reales. La Localización ABLE se usaría para almacenar valores reales aunque no se hubiera incluido en la proposición de declaración.

Podemos hacer funcionar este proceso en el otro sentido. Supongamos que las variables A, B y C deben ser enteras. Esta vez, la proposición será:

INTEGER A, B, C

Observen bien las comas que van entre los nombres de variables en la proposición INTEGER (y los REAL), lo mismo que en la proposición DIMENSION.

No es necesario escribir una proposición DIMENSION para algún arreglo (ejemplo, J(10), si éste se incluye en la proposición de declaración de tipo, puesto que la dimensión del arreglo ya se incluye.

Las variables de precisión doble, las variables complejas y las variables lógicas no obedecen a convenciones predefinidas para su modo y, por consiguiente, deben ser siempre explícitamente definidas en una proposición de declaración de tipo.

CONSTANTES DE PRECISION DOBLE:

Tal como su nombre lo indica, un número de PRECISION DOBLE es aquél que posee una precisión igual a dos veces la precisión de un número regular; es decir, este número tiene dos veces el número de dígitos significativos. Todas las constantes reales, se pueden escribir con exponente o sin él. El valor de la constante, sin exponente, se indica literalmente usando de 6 a 12 dígitos decimales con un punto decimal que precede el primer dígito, sigue el último dígito, o se localiza entre dos dígitos cualesquiera. Observa que un número de dígitos más grande que el número de dígitos permitido para representar constantes reales es suficiente para definir, automáticamente, una constante de precisión doble, siempre y cuando el sistema de computación que se use acepte precisión doble.

Los números de precisión doble se escriben siempre con un punto decimal y pueden ser positivos o negativos. Si no se muestra el signo algebraico, entonces se supone que el número es positivo. Cuando se quiere representar con exponente se hace lo mismo que con las constantes reales, a excepción de que se

debe usar una D, en cambio de E, para indicar el exponente. La parte constante del número se puede escribir como una constante de precisión doble sin exponente, o como una constante real sin exponente. Observa que esta opción permite el uso de menos de 10 dígitos, siempre y cuando se use la letra D. El exponente puede ser cualquier número, positivo o negativo, de dos dígitos decimales que pueden ser cero. Los siguientes números son ejemplos de constantes de precisión doble sin exponente.

247826.591562
.00022064219
3.1415926536

Los siguientes números son ejemplos de constantes de precisión doble con exponente:

27182818285.D-10
30.D+6
5.522D+0

VARIABLES DE PRECISION DOBLE:

Todas las variables de precisión doble se representan por nombres que hacen referencia a las localizaciones del almacenamiento del computador. Ejemplo:

DOUBLE PRECISION A, VALUE, ITEM, XRAY

ENTRADA/SALIDA - CONTROL DE FORMATO:

Las variables de precisión doble pueden aparecer en las proposiciones de entrada/salida. Cuando esto ocurre, de un registro de entrada (tarjeta de datos) se leerán los números en precisión doble o se describirán en un registro de salida. Esta operación READ/WRITE debe controlarse por una proposición FORMAT. La forma general del código que describe los números en precisión doble es:

Fw.d si el número aparece sin un exponente

Dw.d si el número aparece con un exponente

donde w es una constante entera que indica el ancho del campo y d es una constante entera que indica el número de dígitos a la derecha del punto decimal.

EXPRESIONES ARITMETICAS.

Una constante o variable en precisión doble puede aparecer en cualquier expresión aritmética exactamente en la misma forma como aparece una variable o constante real. Más aún, los valores en precisión simple (real) y en precisión doble pueden aparecer en la misma expresión aritmética.

La aparición de cualquier cantidad en precisión doble en la expresión aritmética produce la evaluación de la expresión total en precisión doble, dando el resultado en precisión doble. Las reglas para formar expresiones en precisión doble son iguales a aquéllas para formar expresiones aritméticas simples. Los valores enteros no pueden aparecer generalmente en expresiones aritméticas en precisión doble, excepto que un entero pueda utilizarse para expresar la potencia a la cual un valor en precisión doble debe elevarse.

En los siguientes ejemplos de expresiones de precisión doble se supone que las variables A y B son de modo real: la variable D es de modo de precisión doble

```
A + D
A - 7.5D + 0.4
A * B + B
A ** D
D ** B
3.E + 6/D
```

FUNCIONES DE BIBLIOTECA EN PRECISION DOBLE:

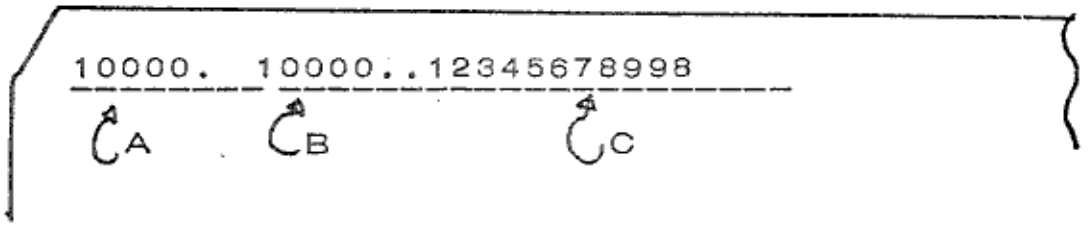
Las funciones tales como, SIN, COS, SQRT, etc. que se han utilizado anteriormente, evalúan la función deseada con una precisión simple. Los cálculos en precisión doble que involucran estas funciones requieren efectuar su evaluación en un grado de precisión mayor. Por tal razón se han creado las siguientes funciones en precisión doble.

```
DSIN
DLOS
DATAN
DLOG
DSQRT
DABS
```

El argumento de estas funciones deben ser cantidades en precisión doble. El valor calculado lo es en precisión doble.

Ejemplo X-1.

Escribir un programa para leer la tarjeta de datos indicada y determinar la suma de los valores listados

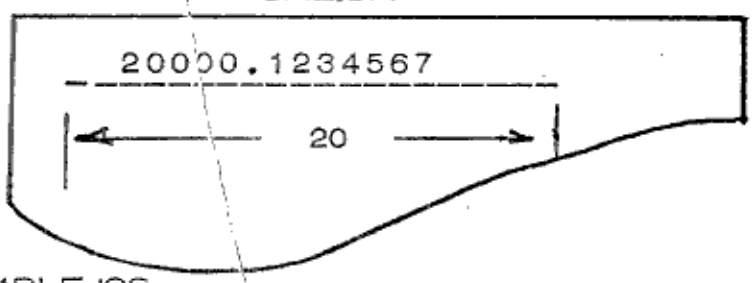


A y B puede almacenarse en forma simple

C, requiere precisión doble por lo tanto la suma será precisión doble.

```
      6
        DOUBLE PRECISION C, SUM
        READ (5 , 7)A, B, C
    7   FORMAT (F6.0, F8.0, F16.15)
        SUM = A + B + C
        WRITE (2 , 8)SUM
    8   FORMAT (F20.12)
        STOP
        END
```

SALIDA



VALORES COMPLEJOS:

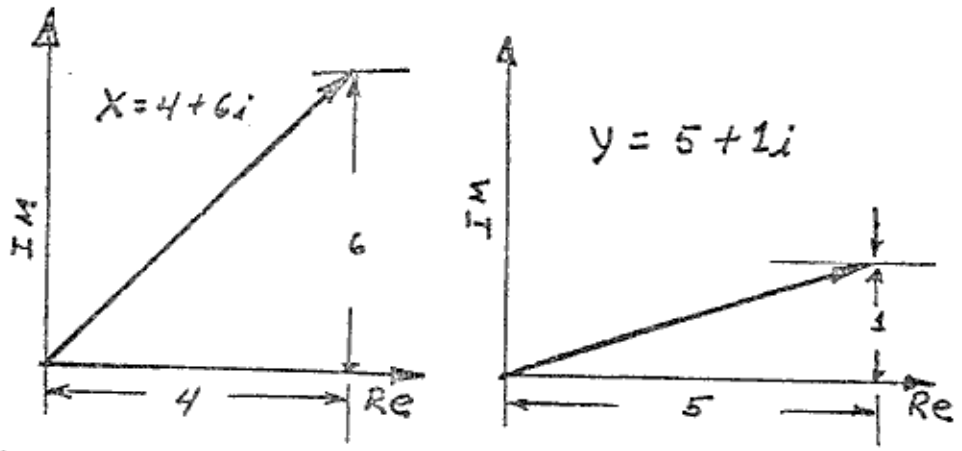
Los números complejos de la forma:

$$(a + bi)$$

frecuentemente ocurren en matemáticas consisten en un par ordenado de números reales (a y b), el último de los cuales se combina con la cantidad "i" que tiene la propiedad:

$$i^2 = \sqrt{-1}$$

Los números complejos pueden utilizarse para representar vectores



La suma de números complejos se define como:

SUMA

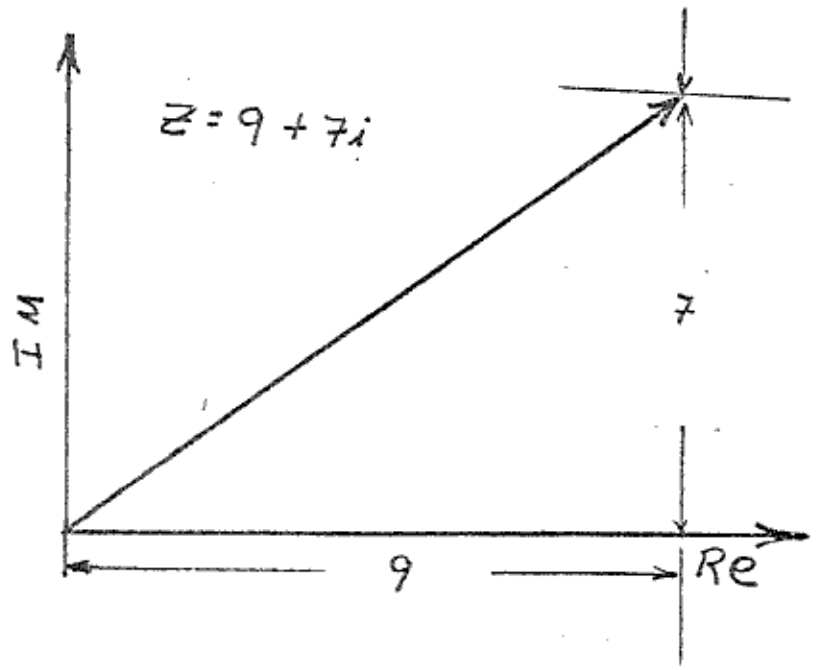
$$(a + bi) + (c + di) = (a + c) + (b + d)i$$

Si usamos X y Y vectores

$$X = 4 + 6i$$
$$Y = 5 + 1i$$

$$Z = (4 + 6i) + (5 + 1i) = (9 + 7i)$$

El resultado es el número complejo Z cuyo significado es la resultante (vector suma) del sistema



El ejemplo anterior demuestra que la identificación y manipulación aritmética de números complejos involucra la identificación y manipulación de las partes real e imaginaria de tales números. Para facilitar esto, algunos compiladores están escritos para permitir el almacenamiento directo y la manipulación directa de los valores complejos.

Específicamente, si una variable se declara en una proposición TIPO como variable compleja, este nombre de variable identificará dos posiciones consecutivas reales en la memoria, en las cuales se almacenan las partes real e imaginaria del número complejo.

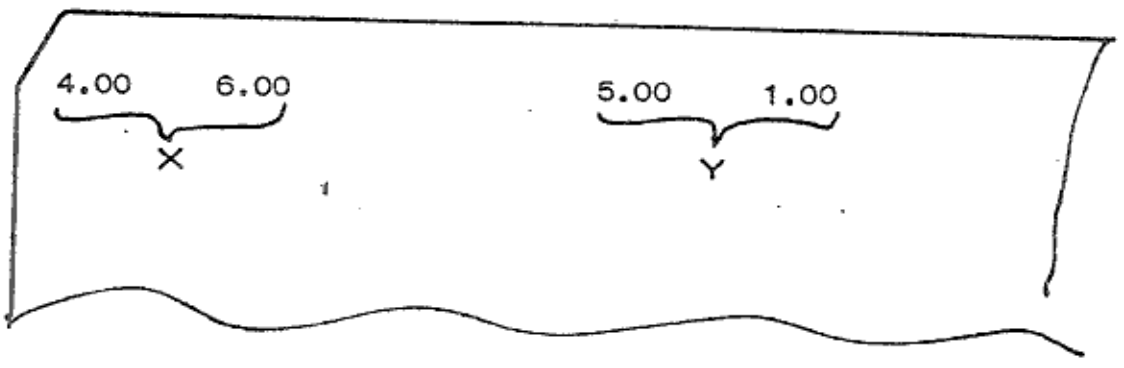
LAS PARTES REAL E IMAGINARIA SE IDENTIFICAN POR UN SOLO NOMBRE

Además, si este nombre de variable aparece en una expresión aritmética, los símbolos operacionales (+, -, *, /, etc.) se interpretan como suma compleja, resta compleja, multiplicación compleja, etc.

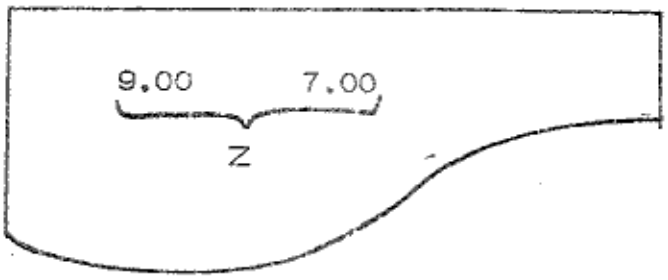
Ejemplo:

```
1    6
C    |SEGMENTO DE PROGRAMA PARA SUMAR LOS VECTORES X y Y
     |COMPLEX X, Y, Z
C    |X y Y SE LEEN DE LOS DATOS
     |READ (5 , 9)X, Y



C    |LA TARJETA DE DATOS CONTIENE CUATRO VALORES
   9 |FORMAT (4F10.2)
     |Z=X+Y
     |WRITE (6 , 10)Z
  10 |FORMAT (2F10.2)
```

Tarjeta de datos



Hoja de resultados.



La proposición tipo se utiliza para identificar a X, Y y Z como nombres de variables complejas. En seguida se ejecuta la proposición READ. De los datos se leen cuatro valores puesto que X y Y se han declarado cantidades complejas. Este segmento del programa calcula el vector Z. En la proposición

$$Z = X + Y$$

el símbolo operacional + se interpreta como una suma compleja.

CONSTANTES COMPLEJAS :

Una constante compleja se forma especificando dos números reales encerrados en paréntesis y separados por una coma.

Estos valores representan las partes real e imaginaria de la constante compleja. Ejemplo

(6.00 , 5.20)	significa	6.00 + 5.20i
(-4.82 , 16.06)		- 4.82 + 16.06i
(0.00 , 4.00)		0.00 + 4.00i
(2.00E+02 , 4.10)		200.00 + 4.10i

VARIABLES COMPLEJAS :

En lenguaje FORTRAN, las variables complejas se representan por medio de nombres (de acuerdo con las reglas para nombrar una variable) y luego se declara como un valor complejo utilizando la proposición TIPO. Ejemplo

```
COMPLEX X, BRN, SUM
COMPLEX A1, A2, A3
COMPLEX X, Y
```

ENTRADA / SALIDA :

Se realiza en forma similar a la manera como los valores reales se transfie-- ren, excepto que : la aparición del nombre de una variable compleja en la lista de una proposición READ/WRITE produce la transferencia de dos números rea_ les.

La proposición FORMAT de control debe proveer especificaciones separadas - que describan cada parte del número complejo.

EXPRESIONES ARITMETICAS :

Las operaciones aritméticas que incluyen números complejos requieren aten- - ción especial.

OPERACIONES COMPLEJAS

SUMA $(a + bi) + (c + di) = (a + c) + (b + d)i$

RESTA $(a + bi) - (c + di) = (a - c) + (b - d)i$

MULTIPLICACION $(a + bi) * (c + di) = (ac - bd) + (bc + ad)i$

DIVISION $(a + bi) \div (c + di) = \dfrac{(ac + bd)^2 + (bc - ad)i^2}{c^2 + d^2}$

EXPONENCIACION $(a + bi)^n =$

$$[(a^2 + b^2)^{1/2} (\cos\theta + i \operatorname{sen}\theta)]^n =$$

$$= (a^2 + b^2)^{n/2} (\cos n\theta + i \operatorname{sen} n\theta)$$

donde

θ = arcotangente $\dfrac{b}{a}$

n = un entero

sólo se pueden elevar a una potencia entera positiva.

Una expresión compleja define una secuencia de operaciones aritméticas a realizarse incluyendo constantes complejas, variables complejas, constantes reales y variables reales. Los valores enteros sólo pueden aparecer como subíndices o como exponentes. Ejemplos

```
COMPLEX  Z, A, B, SUM

Z = (5.28, -6.14) * (7.56, 2.28)

B = Z - (2.26, 3.39)

A = (B/Z) ** 2

SUM = A + B + Z

X = CSQRT ((4.0, 3.0))
```

Aún cuando la formación de una expresión compleja es similar a la formación de una expresión real, siempre existen ciertas diferencias

a) Una cantidad compleja sólo puede elevarse a una potencia entera, los valores reales y complejos no se pueden utilizar como el exponente de una expresión compleja.

b) El resultado de una expresión compleja es un valor complejo y debe almacenarse como tal. Si el lado derecho de una proposición aritmética es una expresión compleja, la variable al lado izquierdo debe ser una variable compleja.

c) Las variables complejas tienen sus propias funciones de biblioteca. Algunas de estas funciones requieren explicación adicional.

NOMBRE	PROPOSITO
CABS	Calcula el valor absoluto de un número complejo.
CMPLX	Transforma dos valores reales en una sola cantidad compleja.
REAL	Separa y da justamente la parte real de un número complejo.

NOMBRE	PROPOSITO
AIMAG	Separa y da justamente la parte imaginaria de un número complejo.
CONJUG	Produce al conjugado de su argumento complejo.
CEXP	e^z
CLOG	$\ln z$
CSIN	$\text{sen } z$
CCOS	$\cos z$
CSQRT	$\sqrt{z}$

CONTROL DE UN PROGRAMA POR MEDIO DE LOGICA FORTRAN:

Hasta este momento las aplicaciones del lenguaje FORTRAN se han restringido a esos problemas que conducen por sí mismos a una solución numérica exacta. Las variables en precisión doble y compleja lógicamente se desarrollaron para aumentar la capacidad de cómputo y la conveniencia del lenguaje FORTRAN para manipular ciertos problemas dentro de esta extensa categoría.

Sin embargo, ahora la atención debe dirigirse a otra clase de problemas que se caracterizan en que no conducen por sí mismos a una representación numérica exacta. Los problemas en diagnósticos médicos, selección de personal o análisis de compuestos químicos se encuentran en esta categoría. La solución se basa en respuestas si/no de preguntas tales como:

¿Se disuelve el compuesto en ácido sulfúrico?

¿Es el C.I del aspirante mayor que 120 pero menor que 140?

El análisis de estos problemas esta íntimamente ligado con la lógica de dos vectores. Por lo tanto la variable lógica y la expresión lógica se han desarrollado para dar al lenguaje FORTRAN un medio conveniente de almacenar y manipular la información típica de estos problemas, o sea "si/no" o "verdadero/falso"

CONSTANTES LOGICAS :

Se representan como sigue :

.TRUE. y .FALSE.

observa que la cadena de símbolos o caracteres que definen estas constantes deben siempre escribirse entre dos puntos decimales.

VARIABLES LOGICAS.

Los nombres de variables utilizados para representar valores lógicos deben declararse en una proposición TIPO LOGICO

LOGICAL X, Y, Z, A, BIT

Una variable lógica solamente puede tomar uno de los siguientes valores :

TRUE	(verdadero)
ó	
FALSE	(falso)

ENTRADA/SALIDA :

La forma general del código de formato que controla las cantidades lógicas es :

Lw

donde w especifica el ancho del campo. En la entrada, el primer caracter diferente a espacio en blanco dentro de cada campo especificado debe ser la letra T o la F para indicar TRUE ó FALSE. Los caracteres restantes, si los hay, no son importantes. Si el campo es espacios en blanco, se le asigna el valor falso. Ejemplos

Campo de Entrada	Especificación	Valor Almacenado
TRUEbb	L6	.TRUE.
FINALb	L6	.FALSE.
ØTOMØ	L5	.TRUE.
ØØF3	L4	.FALSE.
ØØ3F	L4	(ERROR)

En operaciones de salida:

Valor Almacenado	Especificación	Salida
.TRUE.	L4	ØØØT
.FALSE.	L1	F

OPERADORES DE RELACION:

Existen seis operadores binarios de relación que se usan para hacer comparaciones entre los valores numéricos de dos expresiones aritméticas, la tabla siguiente describe estos operadores:

Descripción	Símbolo Algebraico	Símbolo en FORTRAN
Igual a	=	.EQ.
No igual a	≠	.NE.
Menor a	<	.LT.
Menor o igual a	≤	.LE.
Mayor a	>	.GT.
Mayor o igual a	≥	.GE.

Una expresión de relación se forma colocando un operador de relación entre cualquiera de dos expresiones, aritméticas que sean del mismo modo. Las dos expresiones que se comparan pueden ser:

1. Ambas de modo entero
2. Ambas de modo real
3. Ambas en modo de precisión doble
4. Una en modo real, y la otra en modo de precisión doble

En cualquier caso, el valor de la expresión de relación es un valor lógico que puede ser .TRUE. ó .FALSE.; entonces, una expresión de relación es equivalente a una variable lógica. Ejemplos:

EXPRESION LOGICA	VALORES
26.548D + 06.GT.XL	TRUE
A.GT.B	FALSE
N.LE.10	T
B.NE.3.0	T
(A.GT.B).AND.(B.EQ.2.0)	F

OPERADORES LOGICOS:

En FORTRAN dispone de tres operadores lógicos.

	Operación	Significado	Símbolo FORTRAN
1.	Negación	no	.NOT.
2.	Intersección (producto lógico)	y	.AND.
3.	Disyunción (suma lógica)	ó	.OR.

.NOT. es un operador unitario que opera sobre el valor lógico que aparece a su derecha. .AND. y .OR. son operadores binarios que operan con los valores lógicos que aparecen antes y después del operador. Los siguientes ejemplos de expresiones lógicas, junto con su significado, suponen que A, B, C son variables lógicas, mientras que X y Y son variables reales.

1. .NOT.A — Si A tiene el valor .TRUE., el valor de la expresión es .FALSE. (niega su valor)

2. B.AND.C — Si B y C tienen ambas el valor .TRUE., el valor de la expresión es .TRUE.

   Si una cualquiera de las variables B o C, tiene el valor .FALSE., el valor de la expresión es .FALSE.

   Si B y C tienen ambas el valor .FALSE., el valor de la expresión es .FALSE.

3. A.OR.C — Si A y C tienen ambas el valor .FALSE., el valor de la expresión es .FALSE.

   Si A y C tienen ambas el valor .TRUE., el valor de la expresión es .TRUE.

Si una cualquiera de las variables A o C, tienen el valor .TRUE., el valor de la expresión es .TRUE.

4. B.AND.X.LT.Y Si la expresión de relación (X.LT.Y) se reemplaza por C en la expresión (2), entonces esta expresión es equivalente a la expresión que se analiza.

NOT	A = F	A = T
	T	F

AND	A = F	A = T
B = F	F	F
B = T	F	T

OR	A = F	A = T
B = F	F	T
B = T	T	T

JERARQUIA DE LAS OPERACIONES:

Esta jerarquía comprende ahora operadores aritméticos, lógicos y de relación.

1. Exponenciación **
2. Operaciones Aritméticas *, /, +, -
3. Operaciones de Relación .GT., .GE., .EQ., .LT., .LE., .NE.
4. Operaciones Lógicas

1º	.NOT.
2º	.AND.
3º	.OR.

Cuando existe más de una operación del mismo nivel de procedencia, entonces estas operaciones del mismo orden se procesan de acuerdo con su localización de izquierda a derecha. También, lo mismo que en las expresiones aritméticas, el orden se puede cambiar haciendo uso del paréntesis. Ejemplos

A.AND.B.OR.C.AND.X.LT.Y

se evalúa de la siguiente manera:

1. Se determina el valor lógico (X.LT.Y) que es de más alta procedencia.
2. Se determina el valor lógico de (A.AND.B).

3. Se determina el valor lógico C.AND. (valor lógico encontrado en el paso 1.

4. Se determina el valor el valor lógico de (resultado del paso 2) .OR. (resultado del paso 3).

En resumen, la expresión se evaluará como si se hubiera escrito:

(A.AND.B).OR.(C.AND(X.LT.Y))

Escribe los pasos de como se evalúa la siguiente expresión lógica.

A.AND.(B.OR.C).AND.X.LT.Y

1º Se determina el valor lógico de (X.LT.Y)
2º Se determina el valor lógico de (B.OR.C)
3º Se determina el valor lógico de (A.AND.resultado paso 2.)
4º Se determina el valor lógico de (paso 3.AND.paso 1)

PROPOSICIONES DE ASIGNACION LOGICA:

La proposición de asignación lógica se escribe:

V = exp

donde

V es una variable lógica y exp es cualquier expresión lógica. Ejemplos

```
A = .TRUE.
B = A.OR.X.LT.Y
X = A.GT.6.2.OR..NOT.T
C = J.LE.M
```

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. La forma general es:

```
      6
      IF (exp) prop 1
      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.

La acción producida por la proposición IF lógico es:

1. Se determina el valor lógico de la expresión.

2. Si el valor de la expresión es .TRUE. el programa pasa a procesar la proposición prop 1. Si prop 1 es una proposición de transferencia, el control del programa se dirige de acuerdo con prop 1. Si prop 1 no es una proposición de transferencia. Se lleva a cabo el proceso correspondiente a prop 1 y luego se pasa el control a prop 2.

3. Si el valor de la expresión es .FALSE., el programa pasa inmediatamente a procesar prop 2.

Ejemplos:

```
IF(X.LT.Y)  Z = SQRT(Y-X)
P = A + Z
```

prop 2 — prop 1

```
IF(A.OR.B)  GØ TØ 25
STOP
```

prop 2 — prop 1

Ejemplo.

Funciones de distribución de probabilidad: Un cálculo tipico de probabilidad que usa dos variables aleatorias es:

"Encuentre la probabilidad de que $X \leq x$ y $Y \leq y$"

Este problema se puede escribir usando la siguiente notación

$$P(X \leq x \text{ y } Y \leq y)$$

Una probabilidad de este tipo se calcula por medio de la función de distribución conjunto $F(x, y)$. Es decir:

$$F(x,y) = \begin{cases} 0 & \text{si} \quad x < 0 \ \text{ó} \ y < 0 \\ (1 - e^{-x})y^3 & \text{si} \quad 0 \leq x \ , \ 0 \leq y \leq 1 \\ (1 - e^{-x}) & \text{si} \quad 0 \leq x \ , \ 1 < y \end{cases}$$

Observa que la función toma valores diferentes a medida que el intervalo de las variables cambia. Ejemplo:

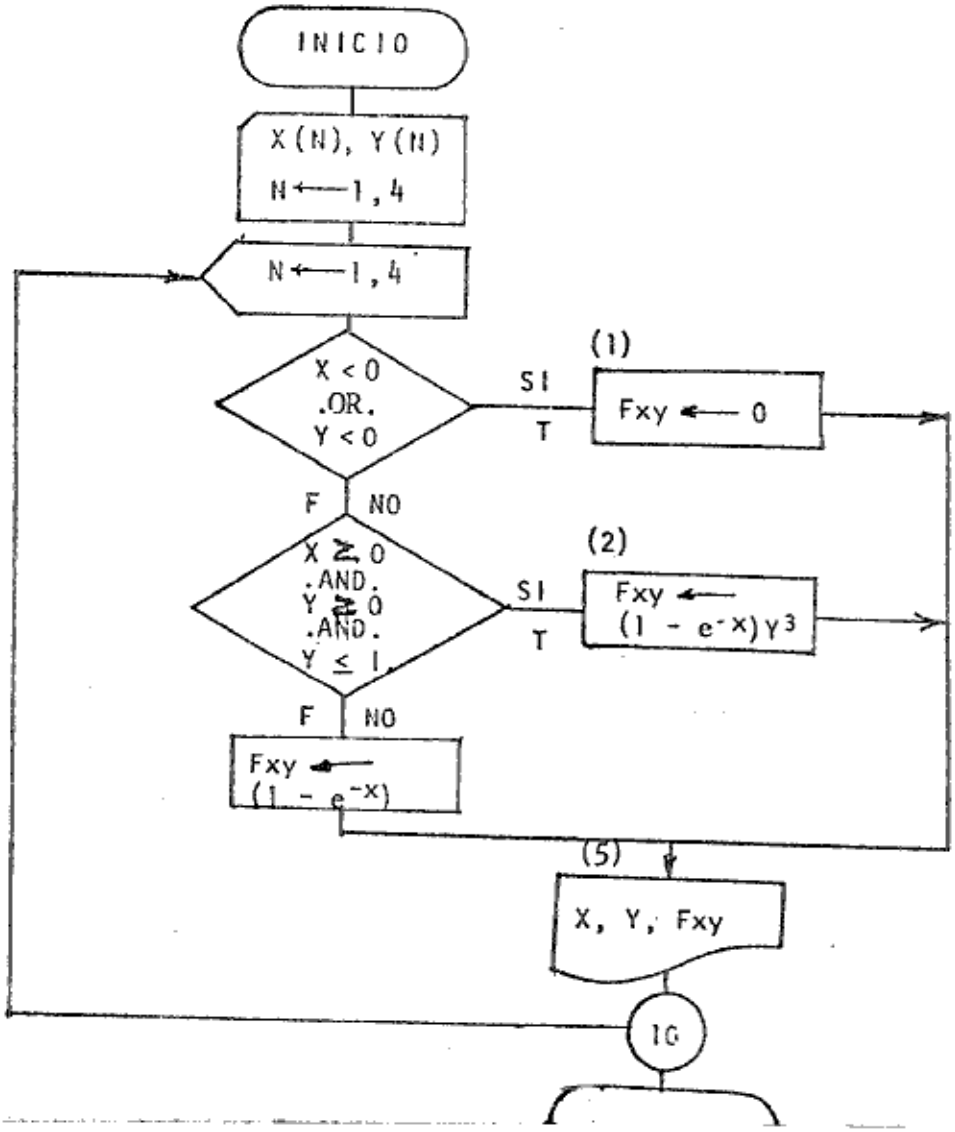
$P(X \leq 0.1 \ \text{y} \ Y \leq -0.1) = F(0.1 \ , \ -0.1) = 0$

$P(X \leq -0.1 \ \text{y} \ Y \leq 0.1) = F(-0.1 \ , \ 0.1) = 0$

$P(0 < X \leq 1 \ \text{y} \ 0 < y \leq 0.5) = F(1 \ , \ 0.5) = (1 - e^{-1}) \ 0.5^5 = 0.0790$

$P(0 < x \leq 1 \ \text{y} \ 1 < y) = F(1 \ , \ y > 1) = (1 - e^{-1}) = 0.632$

El programa para calcular las probabilidades justamente descritas es el siguiente.



```
$ FORGO
C       ESTE PROGRAMA CALCULA PROBABILIDADES CONJUNTAS USANDO
C       UNA FUNCION DE DISTRIBUCION DE PROBABILIDAD.
        DIMENSION X(4), Y(4)
        READ, (X(N), N=1,4), (Y(N), N=1,4)
        DØ 10 N=1 , 4
        IF (X(N).LT.O..OR.Y(N).LT.O.) GØ TØ 1
        IF(X(N).GE.O..AND.(Y(N).GE.O..AND.Y(N).LE.1.)) GØ TØ 2
        FXY=(1.-EXP(-X(N))
    5   WRITE(6,9) X(N), Y(N), FXY
        STØP
    2   FXY=(1.-EXP(-X(N)))*Y(N)**3
        GØ TØ 5
    1   FXY=0.
        GØ TØ 5
    9   FORMAT('X=',F10.4,'Y=',F10.4,'PROBABILIDAD CONJUNTA=',F10.4)
        END
: END
.1   -.1   1.   1.   -0.1   0.1   0.5   2.
```

Ejemplo

Cálculo de π . Un problema de interés para los matemáticos de varias generaciones, ha sido el cálculo de la constante π , con un alto grado de precisión. Para este propósito se han propuesto varios métodos, siendo entre ellos uno de los más conocidos el que usa la fórmula

$$\pi = 4 \left[4 \tan^{-1} \frac{1}{5} - \tan^{-1} \frac{1}{239} \right]$$

la función arcotangente se puede evaluar por medio de la siguiente serie infinita.

$$\tan^{-1} X = X - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} \cdots \quad |X| \leq 1$$

Esta serie se puede escribir en una forma más compacta, como sigue:

$$\tan^{-1} X = \sum \frac{(-1)^{n-1}}{(2n-1)} (X)^{2n-1}$$

$$n = 1, 2, 3.....$$

El programa para calcular π con 16 dígitos significativos se muestra a continuación:

```
C       ESTE PROGRAMA ILUSTRA EL USO DE VARIABLES DE PRECISION
C       DOBLE PARA EL CALCULO DE PI
        DOUBLE PRECISION A, B, TEMP, PI, DABS
        A=0.0
        B=0.0
        TEMP=0.0
        DØ 1 I=1,1000
        Z=(2*I-1)
        TEMP=16.*((-1)**(I-1)*(1./5.DO)**(2*I-1))/Z
        IF(DABS(TEMP)-1.D-25)2,2,1
      1 A=A+TEMP
      2 TEMP=0.0
        DØ 3 I=1,1000
        Z=(2*I-1)
        TEMP=4.*((-1)**(I-1)*(1./239.DO)**(2*I-1))/Z
        IF(DABS(TEMP)-1.D-25)4,4,3
      3 B=B+TEMP
      4 PI=A+B
        WRITE(6,10)PI
     10 FØRMAT(4X,'PI=',D24.16)
        STØP
        END
```

donde

$$\pi = A + B = 16 \tan^{-1} \frac{1}{5} - \tan^{-1} \frac{1}{239}$$

$$TEMP = 16 \left(\frac{(-1)^n - 1}{(2n-1)} (X)^{2n-1} \right)$$

$$Z = 2n - 1$$

Se declaran las variables A, B, TEMP y PI como variables de precisión doble. La primera serie que define la función arcotangente se calcula usando X = 1/5, la variable TEMP toma el valor numérico de producto de la constante 16 por cada uno de los términos de la serie; la suma de todos los términos así calculados se almacenan en A para el primer DØ. Las iteraciones continuan hasta que $|TEMP| \leq 10^{-25}$.

El segundo ciclo DØ es igual al primero, excepto que el valor de X = 1/239 y la suma de los términos se almacenan en B.

Ejemplos.

Hacer las instrucciones para leer los valores de las siguientes variables complejas.

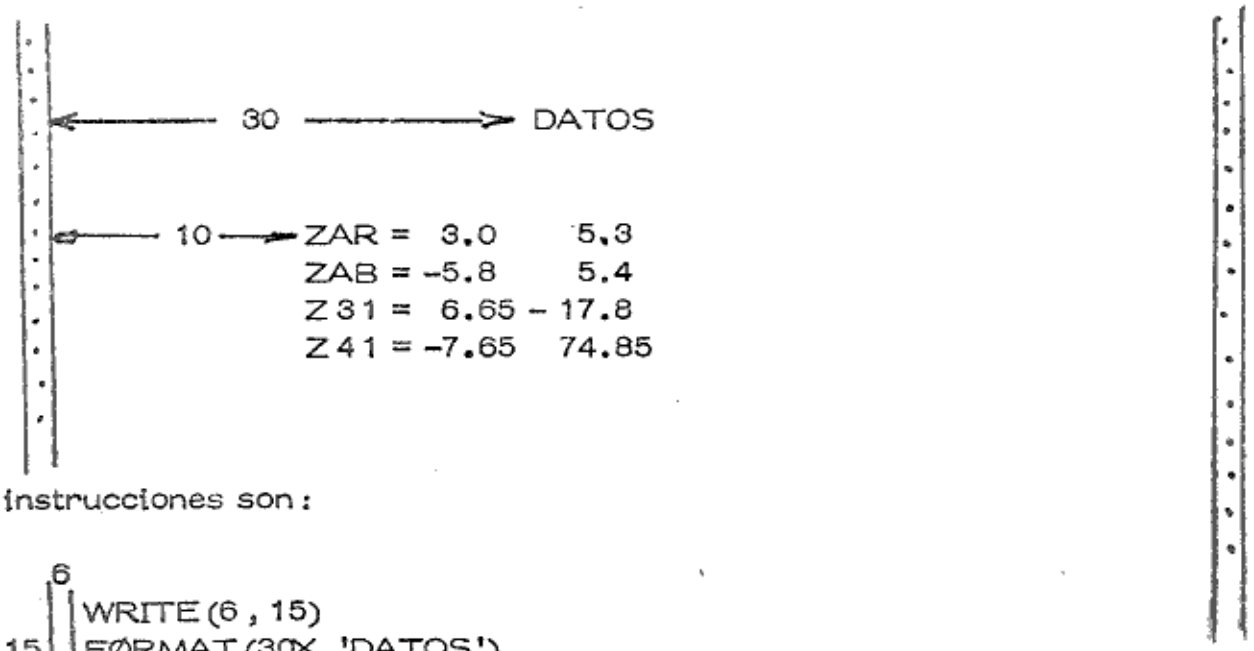
ZAR = 3.0 + i 5.3
ZAB = -5.8 + i 5.4
Z31 = 6.65 - -i17.5
Z41 = -7.65 + i74.85

Además de las instrucciones para leer, se requiere poner al principio del programa la declaración que indica los nombres de las variables complejas.

Las instrucciones son:

```
      6
      |  CØMPLEX ZAR, ZAB, Z31, Z41
      |  READ (5 , 8) ZAR, ZAB, Z31, Z41
    8 |  FØRMAT (4 (2 F6.2))
```

Hacer las instrucciones para imprimir los valores de las variables complejas del ejemplo anterior, se deberá presentar el siguiente reporte.



Las instrucciones son:

```
      6
      |WRITE (6 , 15)
15    |FØRMAT (30X, 'DATOS')
      |WRITE (6 , 16) ZAR, ZAB, Z31, Z41
16    |FØRMAT (10X, 2F6.2)
```

Hacer las instrucciones para leer los valores de un arreglo llamado ZA que es de 1 dimensión y 10 lugares.

ZA =

3.1 + i 5.2
4.3 - i 3.5
8.1 - i 6.5
14.3 - i 15.6
18.5 - i 38.4
6.5 - i 37.6
16.8 - i 5.4
4.5 - i 9.2
16.5 - i 11.3
5.8 - i 13.4

Solución.

En este caso, se requiere declarar que el arreglo ZA contiene variables complejas haciendo la declaración antes de la declaración de DIMENSIØN.

Las instrucciones son:

```
      6
      COMPLEX ZA
      DIMENSION ZA(10)
      READ (5 , 8) (ZA (I), I=1,10)
    8 FORMAT (10F8.2)
```

y los datos aparecerían en la tarjeta en la siguiente forma

```
  6.5  -37.6   16.8  -5.4   4.5  - 9.2   16.5  -11.3   5.8  -13.4
3.1  5.2   4.3  - 3.5   8.1  -5.6   14.3  -15.6   18.5  -38.4
```

Problemas.

Dadas las siguientes matrices de números complejos

$$A = \begin{bmatrix} 1.5 + i5.2 & -3.5 + i2.3 & 4.2 + i4.1 \\ 8.3 + i6.2 & 2.4 + i7.3 & 5.6 + i4.1 \\ 7.2 + i5.4 & 4.1 + i3.2 & 5.4 + i2.4 \end{bmatrix}$$

$$B = \begin{bmatrix} -3.2 + i2.1 & -4.3 + i2.2 & 4.2 + i2.3 \\ 5.1 + i2.4 & 5.2 + i2.5 & 5.3 - i2.6 \\ 6.5 + i2.7 & 6.6 - i2.8 & 6.7 + i2.9 \end{bmatrix}$$

Hacer el diagrama de flujo para sumar A y B y guardar el resultado en una matriz C.

a) Usando variables reales.

b) Usando variables complejas.

Independientemente de que se usen variables reales o variables complejas sabemos que:

$$C_{ij} = \sum_{\substack{i=1,3 \\ j=1,3}} A_{ij} + B_{ij}$$

a) Solución usando variables reales.

La solución dentro de la computadora considera la parte real de cada matriz como un arreglo y la parte imaginaria como otro y se realizan las operaciones respetando las reglas de los números complejos.

Así se pueden usar los siguientes arreglos:

Para representar la matriz $\underline{\underline{A}}$ se usan AR y AI.

$$AR = \begin{bmatrix} 1.5 & -3.5 & 4.2 \\ 8.3 & 2.4 & 5.6 \\ 7.2 & 4.1 & 5.4 \end{bmatrix}$$

$$AI = \begin{bmatrix} 5.2 & 2.3 & 4.1 \\ 6.2 & 7.3 & 4.1 \\ 5.4 & 3.2 & 2.4 \end{bmatrix}$$

Para representar la matriz $\underline{\underline{B}}$ se usan los arreglos BR y BI

$$BR = \begin{bmatrix} -3.2 & -4.3 & 4.2 \\ 5.1 & 5.2 & 5.3 \\ 6.5 & 6.6 & 6.7 \end{bmatrix}$$

$$BI = \begin{bmatrix} 2.1 & 2.2 & 2.3 \\ 2.4 & 2.5 & 2.6 \\ 2.7 & -2.8 & 2.9 \end{bmatrix}$$

```
   TARJETAS DE CØNTRØL
   DIMENSIØN AR(3,3),AI(3,3),BR(3,3),BI(3,3),CR(3,3),CI(3,3)
   READ(5,8)((AR(I,J),J=1,3),I=1,3)
   READ(5,8)((AI(I,J),J=1,3),I=1,3)
   READ(5,8)((BR(I,J),J=1,3),I=1,3)
   READ(5,8)((BI(I,J),J=1,3),I=1,3)
 8 FØRMAT(9F8.2)
   DØ 20 I=1,3
   DØ 20 J=1,3
   CR(I,J)=AR(I,J)+BR(I,J)
   CI(I,J)=AI(I,J)+BI(I,J)
20 CØNTINUE
   WRITE(6,9)((AR(I,J),J=1,3),I=1,3)
   WRITE(6,9)((AI(I,J),J=1,3),I=1,3)
   WRITE(6,9)((BR(I,J),J=1,3),I=1,3)
   WRITE(6,9)((BI(I,J),J=1,3),I=1,3)
   WRITE(6,9)((CR(I,J),J=1,3),I=(1,3)
   WRITE(6,9)((CI(I,J),J=1,3),I=1,3)
 9 FØRMAT(3F8.3)
   STOP
   END
   TARJETAS DE CØNTRØL
   DATOS
```

```
$CONTROL USLINIT
C SUMA DE MATRICES COMPLEJAS
      COMPLEX A,B,C
      DIMENSION A(3,3),B(3,3),C(3,3)
      READ(5,8)((A(I,J),J=1,3),I=1,3)
      READ(5,8)((B(I,J),J=1,3),I=1,3)
      DO 20 I=1,3
      DO 20 J=1,3
   20 C(I,J)=A(I,J)+B(I,J)
      WRITE(6,10)
   10 FORMAT(5X,'LOS DATOS DEL ARREGLO A SON:')
      WRITE(6,11)((A(I,J),J=1,3),I=1,3)
   11 FORMAT(3(2X,F8.2,2X,F8.2))
      WRITE(6,25)
   25 FORMAT(5X,'LOS DATOS DEL ARREGLO B SON')
      WRITE(6,11)((B(I,J),J=1,3),I=1,3)
      WRITE(6,30)
   30 FORMAT(5X,'LOS VALORES DEL ARREGLO C SON')
      WRITE(6,11)((C(I,J),J=1,3),I=1,3)
    8 FORMAT(6F8.2)
      STOP
      END
```

```
LOS DATOS DEL ARREGLO A SON:
 1.56       5.20      -3.50       2.30       4.20       4.10
 8.30       6.20       2.40       7.50       5.60       4.10
 7.20       5.40       4.10       3.20       5.40       2.40
LOS DATOS DEL ARREGLO B SON
-3.20       2.10      -4.30       2.20       4.20       2.30
 5.10       2.40       5.20       2.50       5.30       2.60
 6.50       2.70       6.60      -2.80       6.70       2.90
LOS VALORES DEL ARREGLO C SON
-1.64       7.30      -7.80       4.50       8.40       6.40
13.40       8.60       7.60       9.80      10.90       6.70
13.70       8.10      10.70        .40      12.10       5.30
```

```
00006000  C      ESTE PROGRAMA MULTIPLICA MATRICES COMPLEJAS.
00007000         COMPLEX A,B,C
00008000         DIMENSION A(5,5),B(5),C(5)
00009000         READ(5,*)((A(I,J),J=1,5),I=1,5)
00010000         READ(5,*)(B(I),J=1,5)
00011000         WRITE(6,1)((A(I,J),J=1,5),I=1,5)
00012000         WRITE(6,4)(B(J),J=1,5)
00013000       4 FORMAT(5(10X,F6.3,"+",F6.3,"I",/),/,
00014000        *10X,"PRODUCTO DE MATRICES COMPLEJAS")
00015000         DO 10 I=1,5
00016000         C(I)=0
00017000         DO 11 J=1,5
00018000         C(I)=C(I)+A(I,J)*B(J)
00019000      11 CONTINUE
00020000         WRITE(6,2)C(I)
00021000      10 CONTINUE
00022000       1 FORMAT(1H1,20X,"LAS MATRICES ORIGINALES SON",/,25X,
00023000        *"MATRIZ A",/,5(5(F5.2,2X,"+",F5.2,"I"),/),//,20X,
00024000        *"MATRIZ B")
00025000       2 FORMAT(10X,F6.2,2X,"+",F7.2,"I")
00026000         STOP
00027000         END
```

```
                    LAS MATRICES ORIGINALES SON
                          MATRIZ A
1.10  + 2.20I 3.30  + 4.40I 5.50  + 6.60I 7.70  + 8.80I 9.90  + 1.10I
2.20  + 3.30I 4.40  + 5.50I 6.60  + 7.70I 8.80  + 9.90I 3.40  + 5.60I
6.70  + 8.90I 5.60  + 7.80I 3.40  + 5.60I 7.80  + 2.30I 4.70  + 8.90I
8.70  + 7.50I 4.50  + 4.50I 3.50  + 5.60I 1.80  + 3.60I 7.50  + 4.30I
5.20  + 4.60I 7.80  + 3.40I 5.70  + 8.30I 2.20  + 4.60I 6.70  + 7.90I

                          MATRIZ B
                1.100+ 2.200I
                3.300+ 4.400I
                5.500+ 6.600I
                7.700+ 8.800I
                7.700+ 2.200I

              PRODUCTO DE MATRICES COMPLEJAS
                30.25  + 272.25I
               -34.54  + 336.16I
                10.12  + 293.37I
                -2.20  + 205.59I
                20.02  + 256.56I
```