

ELEMENTOS DE FORTRAN 90

Conjunto de caracteres

Letras:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z

Dígitos:

0 1 2 3 4 5 6 7 8 9

Caracteres especiales:

= + - * / () , . ' : ! " % & ; < > ? \$ _ espacio

Las letras mayúsculas son equivalentes a las minúsculas excepto en constantes de caracteres,

Ejemplo:

ABC es igual abc
'UNAM' es diferente a 'unam'

Formas de escritura del código fuente

Un programa FORTRAN consiste de una serie de instrucciones. Estas instrucciones pueden ser ejecutables y no-ejecutables.

Las instrucciones ejecutables describen las acciones que el programa realizará cuando este sea ejecutado (adición, multiplicación...).

Las instrucciones no-ejecutables proveen información necesaria para la operación del programa.

FORTRAN 90 ofrece dos formas de escribir el código fuente: fija y libre

Forma fija:

- Columnas de la 1 a la 5 están reservadas para etiquetas.
- Las etiquetas pueden contener espacios.
- La columna 6 es usada para indicar la continuación de la línea anterior mediante un carácter de continuación (cualquiera, diferente del espacio o del 0). Puede haber hasta 19 líneas de continuación. Las líneas de continuación no pueden estar etiquetadas. Una instrucción END no puede ser continuada.
- Columnas de la 7 a la 72 son usadas para las instrucciones FORTRAN.
- Las columnas de la 73 en adelante son ignoradas.
- Los comentarios se indican comenzando una línea con una C o un * en la columna 1. Adicionalmente, los comentarios al final de una instrucción pueden ser escritos en cualquier columna (excepto la 6), antecediendo a su texto un !.
- Las líneas de comentarios no pueden ser continuadas.
- El ; puede ser usado para separar varias instrucciones escritas en una misma línea.
- Los espacios son ignorados, excepto en constantes de caracteres.

Forma libre:

- Una línea puede tener una longitud hasta de 132 caracteres.
- Los espacios son usados para separar nombres, constantes o etiquetas de nombres, constantes o etiquetas adyacentes. Los espacios también son utilizados para separar nombres de las instrucciones FORTRAN.
- Un ! inicia un comentario, incluyendo a los escritos al final de una línea.
- El ; puede ser usado para separar instrucciones en una misma línea.
- El & , empleado como último carácter (no espacio, no comentario) de una línea, indica que ésta va a ser continuada en la siguiente línea distinta de un comentario.
- Si un nombre, instrucción o etiqueta no son terminados en una línea, el primer carácter diferente del espacio en la línea de continuación deberá ser un & seguido de los caracteres restantes.
- Se permiten hasta 39 líneas de continuación.
- Los comentarios no pueden ser continuados en otra línea.

- FORTRAN 77 sólo admite el código fuente en forma fija, pero excluye a los comentarios con ! y no es posible escribir más de una instrucción en una línea.

Estructura de un programa FORTRAN

Los programas FORTRAN consisten de una serie de instrucciones ejecutables y no_ejecutables; se dividen en tres secciones:

- La *sección de declaración* consiste en un grupo de instrucciones no_ejecutables al comienzo del programa que definen el nombre del programa y el número y tipo de variables usadas.
- La *sección de ejecución* consiste de un grupo de instrucciones que describen las acciones que realizará el programa.
- La *sección de terminación* consiste en las instrucciones que detienen la ejecución del programa y le avisan al compilador que el programa está completo

La sección de declaración

La Instrucción **PROGRAM** es una instrucción no ejecutable que especifica un nombre lógico para un programa principal. Por ejemplo:

```
PROGRAM Eliminacion_de_Gauss
PROGRAM principal
```

El nombre dado en una instrucción **PROGRAM** no podrá ser empleado como nombre de un subprograma externo o de un módulo en el mismo programa ejecutable.

La sección de ejecución

La instrucción **READ** permite leer los valores que serán asignados a una lista de variables. Los datos serán proporcionados a través de la unidad estándar de lectura (el teclado de la computadora), separados entre sí por comas o por espacios. **PRINT** o **WRITE** imprimen en la unidad estándar de escritura (la pantalla de la computadora) los valores de una lista de variables, constantes o expresiones.

La sección de terminación

La sección de terminación consiste de las instrucciones **STOP** y **END PROGRAM**. La instrucción **END PROGRAM** le dice al compilador que no hay

más instrucciones en el programa. La instrucción **STOP** termina la ejecución de un programa.

Nombres

Los nombres son usados en FORTRAN para identificar entidades tales como variables o unidades de programa. Un nombre comienza con una letra, puede ser hasta de 31 caracteres de longitud y consiste de una combinación de letras, dígitos y el carácter `_`.

En FORTRAN 77, los nombres tienen una longitud máxima de 6 caracteres y no pueden incluir al carácter `_`.

Etiquetas

Las instrucciones FORTRAN pueden tener etiquetas para su identificación. Una etiqueta consiste de 1 a 5 dígitos, al menos uno de ellos, diferente de 0. Ejemplos:

123 5000 9 0037 37

La etiqueta 0037 es equivalente a 37

Constantes y Variables

Una **constante en FORTRAN** es un objeto de datos que se define antes de que se ejecute el programa y que no cambia su valor durante la ejecución del programa.

Una **variable en FORTRAN** es un objeto de datos que cambia su valor durante la ejecución del programa. Las variables en FORTRAN tienen un solo nombre, este puede contener hasta 31 caracteres los cuales pueden ser caracteres alfabéticos, dígitos y el carácter `_`; El primer carácter debe ser alfabético.

FORTRAN tiene 5 tipos de datos intrínsecos que son:

Numéricos : INTEGER, REAL, COMPLEX
Lógicos : LOGICAL
Cadena de caracteres: CHARACTER

Operadores aritméticos

Los operadores aritméticos disponibles en FORTRAN son:

Suma, resta	+, -
Multiplicación, división	*, /
Exponenciación	**

Operadores lógicos

Sólo pueden tomar un valor: verdadero (.TRUE.) o falso (.FALSE.). Los operadores de relación permiten construir expresiones lógicas.

OPERADOR		SIGNIFICADO
Nuevo estilo	Estilo anterior	
.EQ.	==	igual
.NE.	/=	diferente
.LT.	<	menor que
.LE.	<=	menor o igual que
.GT.	>	mayor que
.GE.	>=	mayor o igual que

Declaración de datos tipo character

Para el tipo CHARACTER, existe un parámetro adicional:

(LEN=número máximo de caracteres que puede representar un objeto de datos tipo CHARACTER).

Parametrización de datos en FORTRAN

En FORTRAN 90 los tipos de datos pueden ser parametrizados. Esto puede aplicarse para asignar un nombre a un valor determinado para garantizar que no cambie su valor durante la ejecución del programa, ejemplo:

Type, PARAMETER:: nombre=valor

ESTRUCTURAS DE decisión Y DISEÑO DE PROGRAMACIÓN

Técnica "top-down" para el diseño de programación

- Establecer claramente el problema que se quiere resolver
- Definir las variables de entrada requeridas por el programa y la salida que resultara del programa
- Diseñar el algoritmo que se implementará en el programa
- Escribir el algoritmo resultante (pseudocódigo) en instrucciones FORTRAN
- Probar el funcionamiento del programa

Algoritmos: un algoritmo es un procedimiento paso por paso para encontrar la solución a un problema.

La descripción de un algoritmo debe realizarse de tal forma que sea fácil de entender y que permite la implementación en instrucciones FORTRAN.

Pseudocódigos y diagramas de flujo

En los pseudocódigos se expresa línea por línea cada idea o segmento del código.

Los diagramas de flujo describen algoritmos de una forma gráfica. En estos diagramas se utilizan distintos símbolos gráficos para representar operaciones distintas en el algoritmo.

ESTRUCTURAS DE DECISIÓN

Instrucción IF

IF (expresión lógica) instrucción ejecutable

Si la expresión lógica encerrada entre paréntesis en el IF es verdadera, entonces se ejecuta la instrucción indicada. Si la expresión es falsa, no se ejecuta dicha instrucción. La instrucción debe ser diferente a otro IF o al END de un programa, función o subrutina. Ejemplos:

```
IF (a > b) THEN write (*,*) a
ELSE y ELSE IF
```

Estas instrucciones se emplean si se quiere ejecutar un conjunto de instrucciones si alguna condición es verdadera y otro conjunto distinto de instrucciones si otras condiciones son verdaderas, Ejemplo:

```
IF (expresión lógica 1 ) THEN
```

```
Instrucción 1
```

```
Instrucción 2
```

```
ELSE IF (expresión logica 2) THEN
```

```
Instrucción 1
```

```
Instrucción 2
```

```
ELSE
```

```
Instrucción 1
```

```
Instrucción 2
```

```
END IF
```

Construcción CASE

La construcción CASE es usada para seleccionar entre varios bloques de código ejecutable, basándose en el valor de una expresión. Necesariamente, la expresión debe ser del tipo INTEGER, LOGICAL o CHARACTER.

La forma general de la instrucción CASE es:

```

SELECT CASE(case expresión)
CASE(case expresión 1)
  Instrucción 1
  Instrucción 2
.
.
CASE(case expresión 2)
  Instrucción 1
  Instrucción 2
.
.
CASE DEFAULT
  Instrucción 1
  Instrucción 2
.
.
END SELECT

```

CICLOS

Los ciclos en FORTRAN son construcciones que nos permiten ejecutar una secuencia de instrucciones más de una vez. Las dos formas básicas de los constructores de ciclos son ciclos WHILE y ciclos ITERATIVOS. La diferencia entre estos dos tipos de ciclos es la forma en que controlan el proceso repetitivo.

Ciclo WHILE

Un ciclo WHILE es un bloque de instrucciones que son repetidos indefinidamente hasta que alguna condición se satisface. La forma general de un ciclo WHILE es:

```

DO
.....
IF (expresión lógica) EXIT
.....
END DO

```

Ciclo DO WHILE

Permite la repetición de un conjunto de instrucciones (rango del ciclo) bajo el control de una expresión lógica.

El rango del ciclo se ejecutará mientras la expresión lógica sea verdadera. Si la expresión lógica es falsa, entonces el control se transferirá a la siguiente instrucción ejecutable después del rango del ciclo.

DO WHILE (expresión lógica)

.....

.....

.....

END DO

Ciclos iterativos

Ciclo DO

Un ciclo DO permite repetir un número conocido de veces un conjunto de instrucciones. Dicho conjunto de instrucciones se conoce como rango del ciclo DO:

DO índice=valor inicial,valor final,incremento

Instrucción 1

Instrucción 2

Instrucción n

END DO

donde:

índice es una variable escalar del tipo INTEGER que actúa como el índice del ciclo DO.

Valor inicial, valor final e incremento son constantes, variables o expresiones de tipo INTEGER. Si es omitido el incremento, el compilador supone que es igual a 1. El incremento debe ser diferente de cero. Valor inicial, valor final e incremento controlan el número de veces que será ejecutado el rango del ciclo.

Las siguientes son observaciones sobre el uso del ciclo DO:

Cuando el valor inicial es mayor que el valor final y el incremento es positivo, entonces el ciclo no es ejecutado

Var, valor inicial, valor final o incremento no se pueden modificar dentro de un ciclo DO