

## Instrucciones:

Se le solicita crear un lenguaje de programación llamado NeoScript. Este lenguaje tendrá una sintaxis inspirada en Visual Basic (VB), pero con algunas diferencias clave y utilizando el punto y coma (;) como separador de líneas, y nombres especiales para las sentencias de control.

### Comentario de una línea

Para indicar un comentario de una línea se comenzará con el signo #, y todo el texto seguido en la línea deberá ser ignorado por el compilador.

```
# Este es un comentario de una línea
```

### Comentario de varias líneas

Para indicar un comentario de una línea se comenzará con el signo /\*, y todo el texto seguido en la línea deberá ser ignorado por el compilador.

```
/* Este es un comentario de  
varias líneas */
```

## Tipos de Variables

- int: Para números enteros.
- char: Para caracteres individuales.
- boolean: Para valores de verdadero o falso.
- string: Para cadenas de texto.
- double: Para variables decimales

Para declarar una variable o grupo de variables, se hará uso de la palabra Dim, y luego se definirá el tipo de variable con la palabra reservada como, los ejemplos será como los siguientes:

```
Dim contador Como int = 0;
```

```
Dim cadena, texto, caracteres Como string = "";
```

```
Dim caracter Como char = 'A';
```

```
Dim decimal, numero Como double = 0.0;
```

## Operadores aritméticos

Los operadores aritméticos permitidos serán los siguientes

- + : suma
- - : resta
- \* : multiplicación
- / : división
- ^ : potencia
- % : modulo (residuo de la división)

## Operadores Lógicos

Los operadores lógicos permitidos serán los siguientes:

- > : mayor que
- < : menor que
- >= : mayor o igual
- <= : menor o igual
- == : igual que
- != : diferente que

## Sentencia Imprimir

Será una sentencia que se usará para imprimir en consola el resultado de una variable, para usarla deberá venir la palabra reservada **Imprimir** seguido de varios valores de variables o variables separados por **&**. Ejemplo:

```
Imprimir "Hola mundo";  
Imprimir "Valor de i: " & i;  
Imprimir "Valor de i: " & i & "Valor j: " & j;
```

## Sentencia if/else

La estructura del código será la siguiente:

```
Si condición Entonces  
    # Código a ejecutar si la condición es verdadera  
No  
    # Código a ejecutar si la condición es falsa  
FinSi;
```

## Sentencia switch:

La estructura del código será la siguiente:

```
Seleccionar Caso variable  
    Caso valor1:
```

```

    # Código a ejecutar si variable es igual a valor1
FinCaso;
Caso valor2:
    # Código a ejecutar si variable es igual a valor2
    FinCaso;
Caso Contrario:
    # Código a ejecutar si ninguno de los casos anteriores se cumple es el default
    FinCaso;
FinSeleccionar;

```

## Sentencia While

La estructura del código será la siguiente:

```

Mientras condición
    # Código a ejecutar mientras la condición sea verdadera
FinMientras;

```

## Sentencia Do While

La estructura del código será la siguiente:

```

Hacer
    # Código a ejecutar al menos una vez y luego mientras la condición sea verdadera
RepetirMientras condición;

```

## Sentencia For

La estructura del código será la siguiente:

```

Para i = inicialización Hasta condición Paso actualización
    # Código a ejecutar mientras la condición sea verdadera
Siguiente i;

```

## Manejo de errores:

La aplicación deberá de actuar de la siguiente forma si detecta uno de los siguientes errores:

### Errores Léxicos:

Si se ingresa un símbolo no reconocido por el lenguaje deberá de ignorar el carácter e indicar la fila y columna donde se encuentra, por ejemplo:

Dim letra Como string = @;

Deberá marcar como error el carácter @, la línea donde se declara la variable podrá ignorarse.

### **Errores Sintácticos:**

Si se ingresa una línea con una instrucción desordenada el sistema deberá ignorar la sentencia donde se encuentra la línea incorrecta y marcar en donde se generó el error, por ejemplo:

Dim letra Como int = string;

Deberá marcar que string es una palabra no esperada y deberá ignorar el bloque de código que contenga la línea mencionada.

### **Errores Semánticos:**

Los únicos errores semánticos que se validarán serán cuando se asigne a una variable un tipo de dato que no sea el permitido por ejemplo:

Dim num Como int = 1.5;

En este caso la aplicación lanzará un error de ejecución y no se podrá ignorar la línea, simplemente se detendrá y ejecutará las instrucciones hasta donde encontró el error.

Tomar en cuenta que la comprobación de tipos deberá hacerse en cualquier asignación no solo en la declaración de variables.

### **Impresión de tabla de variables:**

Las variables sin importar en qué parte del código sean generadas siempre deberán de imprimir una tabla donde se muestran los siguientes datos:

- Nombre de la variable
- Tipo de la variable
- Ambito
- Valor final que tuvo la variable al finalizar el programa.

### **Ejemplos de entrada para el lenguaje:**

#### **Ejemplo 1:**

Dim calificacion Como double = 0.0;  
Dim letraGrado Como String= "B+";

# Determinar la letra del grado basado en la calificación  
Seleccionar Caso calificacion

Caso "A+"

calificacion = 10;

FinCaso;

```

Caso "A"
    calificacion= 9.5;
FinCaso;
Caso "B+"
    calificacion= 9;
FinCaso;
Caso "B"
    calificacion= 8.5;
FinCaso;
Caso Contrario
    calificacion=6;
FinCaso;
FinSeleccionar;

```

```

Imprimir "La calificación es: " & calificacion & " y el grado es: " & letraGrado
Dim contador Como int = 0;
Dim suma Como double = 0.0;

```

```

# Sumar números hasta que la suma sea mayor o igual a 10
Hacer

```

```

    suma = suma + 1.5;
    contador = contador + 1;
    Imprimir "Suma actual: " & suma & ", Contador: " & contador;
RepetirMientras suma < 10.0;

```

```

Imprimir "La suma final es: " & suma & " después de " & contador & " iteraciones.";

```

### Salida de consola ejemplo 1:

```

La calificación es: 9.0 y el grado es: B+
La suma final es: 15.0 después de 10 iteraciones.

```

### Salida de tabla de variables ejemplo 1:

Nombre	Tipo	Ámbito	Valor final
calificacion	double	-	9.0
letraGrado	String	-	B+
contador	int	-	10
suma	double	-	15.0

### Ejemplo 2:

```

# Este es un comentario de una línea
Dim contador Como int = 0;

Mientras contador < 5
    Imprimir "Contador es: " & contador; # Imprime el valor del contador
    contador = contador + 1;
FinMientras;

/*
Este es un comentario de múltiples líneas
que puede abarcar varias líneas de código
*/
Para i = 0 Hasta 11 Paso 2
    Dim j Como int = 3 * i + 1;
    Imprimir "Valor de i: " & i & " Valor de j: " & j;
    Siguiente i;

Dim letra1, letra2 Como char = 'A';
Si letra1 == letra2 Entonces
    Imprimir "Es verdadero";
No
    Imprimir "Es falso";
FinSi;

```

### Salida de consola ejemplo 2:

```

Contador es: 0
Contador es: 1
Contador es: 2
Contador es: 3
Contador es: 4
Valor de i: 0 Valor de j: 1
Valor de i: 2 Valor de j: 7
Valor de i: 4 Valor de j: 13
Valor de i: 6 Valor de j: 19
Valor de i: 8 Valor de j: 25
Valor de i: 10 Valor de j: 31
Es verdadero

```

### Salida de tabla de variables ejemplo 2:

Nombre	Tipo	Ámbito	Valor final
contador	int	-	5
i	int	para2	12

j	int	para2	31
letra1	char	-	A
letra2	char	-	A

### Ejemplo 3:

Dim temperatura Como double = 25.5;

# Verificar si la temperatura es mayor a 30 grados

Si temperatura > 30.0 Entonces

Dim cadena como string = "La temperatura es caliente.";

Imprimir cadena;

# Mientras la temperatura sea mayor a 20 grados, reducirla en 1.5 grados

Mientras temperatura > 20.0

Imprimir "Temperatura actual: " & temperatura;

temperatura = temperatura - 1.5;

FinMientras;

No

Dim cadena2 como string = "La temperatura es moderada.";

Imprimir cadena2;

# Mientras la temperatura sea menor a 30 grados, aumentarla en 1.0 grados

Mientras temperatura < 30.0

Dim temperaturaAnterior como double = temperatura;

temperatura = temperatura + 1.0;

Imprimir "Temperatura actual: " & temperatura & "Temperatura anterior: " & temperaturaAnterior ;

FinMientras;

FinSi;

Imprimir "Proceso completado.";

### Salida de consola ejemplo 3:

La temperatura es moderada.

Temperatura actual: 26.5 Temperatura anterior: 25.5

Temperatura actual: 27.5 Temperatura anterior: 26.5

Temperatura actual: 28.5 Temperatura anterior: 27.5

Temperatura actual: 29.5 Temperatura anterior: 28.5

Temperatura actual: 30.5 Temperatura anterior: 29.5

Proceso completado.

### Salida de tabla de variables ejemplo 3:

Nombre	Tipo	Ámbito	Valor final
temperatura	double		30.5
cadena	string	si1	
cadena2	string	si1	La temperatura es moderada.
temperaturaAnterior	double	no3.mientras4	29.5

### Ejemplo 4:

Dim numero Como double= 10;

Dim par Como double= numero % 2;

# Verificar si el número es par o impar

Si par == 0 Entonces

Imprimir "El número es par.";

numero = numero ^ 3 / 3;

No

Imprimir "El número es impar.";

numero = numero ^ 3 / 4;

FinSi;

imprimir numero;

### Salida de consola ejemplo 4:

El número es par.

333.333

### Salida de tabla de variables ejemplo 4:

Nombre	Tipo	Ámbito	Valor final
numero	double	-	333.333
par	double	-	0