

Avance Laboratorio 4 INF 265

2020-2

Grupo id y nombre:

Grupo 7 – “Segmentation fault”

Integrantes (nombre y código):

Christian Andre Carhuancho Rodríguez	20172665
Angel Jesus Gonzales Aliaga	20175488
Marco Antonio Bossio Principe	20142886
Carlos Santos Toro Vera	20171878

Descripción de Lenguaje:

El lenguaje “**obsidian**” permitirá trabajar con datos y emitir reportes financieros.

Modificaciones realizadas al programa en la implementación de interpretación de código

Se realizaron cambios en las instrucciones de **Carlos Santos Toro Vera 20171878**, ahora sus instrucciones manejan datos de tipo fecha. Además, se realizaron cambios en las instrucciones de asignación de **Marco Antonio Bossio Principe 20142886**, el lenguaje ahora solo soporta asignaciones simples.

Se implementó la interpretación de código de todas las instrucciones, incluso de las que no son detalladas en el informe, pero son necesarias para poder manejar estas instrucciones.

Se adjunta un archivo de prueba que permite verificar el correcto funcionamiento del programa, además se pide al evaluador que lea el archivo README.txt y ejecute todo lo indicado ahí.

Instrucciones

1) *Instrucciones Marco Bossio 20142886*

a) **Declaración:** permite declarar una variable asignándole un tipo de dato, permite la declaración de una o más variables sin y con valor inicial. Se emplea la palabra reservada **Declare**. Vale indicar que el caracter ‘#’ determina el final de las sentencias.

- Declaración simple:
 - Estructura: Declare **nombre_variable** ‘:’ **tipo_de_dato** ‘#’
Declare variableA : number #

- Declaración múltiple:
 - Estructura: Declare **nom_var1 , nom_var2 ' : ' tipo_de_dato '#**
Declare variableA, variableB, variableC : number #
 - La declaración es válida para distintos tipos de datos:
 - Tipo numérico: Declare **nombre_variable : number '#**
Declare variableA , variableB, variableC as number #
 - Tipo char: Declare **"nombre_variable" 'caracter' "as" char '#**
Declare variableA , variableB, variableC as char #
 - Tipo string: Declare **nombre_variable ' : ' str '#**
Declare variableA, variableB : str #
 - Tipo file: Declare **nombre_variable ' : ' file '#**
Declare variableA, variableB : file #

b) Asignación: La asignación utilizada en el lenguaje permite asignar un valor a una variable. Esta asignación no diferencia el tipo de dato de la variable ni del valor a asignar, solo lo asigna.

Variable ' : ' 1 '#

Un ejemplo de asignación es **cantNotas : 0 #**

c) Salida de datos: Como el presente lenguaje está pensado para la impresión de reportes, consideramos necesarias instrucciones especiales para dar formato a los archivos. Los caracteres especiales requieren un \ delante para su impresión:

- Show: imprime en la salida estándar, se puede separar por comas para impresión múltiple
 - impresión directa: **"show" valor #**
 - show "holamundo" # how 456 # show 42 #
 - imprime una variable: **"show" "nombre_variable" #**
 - show variableA #
 - imprime variable y salta la línea: **"show" "nombre_variable" #**
 - Imprime solo salto de línea
 - show endl #
- Showline para impresión de caracteres repetidos, como una línea separadora
 - imprime una línea de 100 caracteres **showline 'caracter' '#**
 - showline '=' #
 - Ejemplo: =====
- Showname para impresión de palabras dando formato de primera letra mayúscula, realiza automáticamente la conversión de caracteres.

- imprime una palabra **"showname" "nombre_variable" #**
 - showname "lenguaje obsidian" #
La salida será: Lenguaje Obsidian
- Showtitle para impresión de palabras en mayúsculas, realiza automáticamente la conversión de caracteres
 - imprime una palabra **"showtitle" "nombre_variable" '#'**
 - showtitle "Ventas" #
La salida será:

Ventas

2) **Instrucciones Christian Carhuacho** **20172665**

- a) **Iterativa de flujo controlado por cantidad:** La instrucción permite realizar una cantidad fija de iteraciones brindada como parámetro. La estructura de la instrucción es la siguiente:

```
For Var Loop:  /* Cantidad determinada por la variable Var */

.....

(Otras instrucciones)

.....

#                /* Fin del cuerpo del bucle definido con el caracter # */
```

Por ejemplo, un uso de la sentencia For Loop es:

```
cant : 10 #
For cant Loop:      //Se ejecutará una cantidad de 10 veces

.....

show "Se ha entrado al bucle" #

.....

#
```

Internamente se disminuye en 1 el valor contenido en cant, luego se verifica que cant sea mayor a 0, en caso sea menor o igual se termina el bucle

- b) **Decremento**: La instrucción permite disminuir en 1 una o más variables. El fin de la instrucción se delimita con el fin de línea.

Dec a # /* Disminuye en 1 la variable a */

Dec a, b, c # /* Disminuye en 1 las variables a, b y c */

Internamente se disminuye en 1 cada una de las variables hasta encontrar el carácter '#'.

- c) **Estructura de control Select**: La instrucción permite realizar una selección entre una cantidad de casos, utilizando una variable como selector. En caso no se cumpla ninguno de los casos, no se ejecutará nada y continuará con el programa.

Select var :

Case 1: ... #

Case 2: ... #

...

#

Un ejemplo de la estructura Select es:

Select num :

Case 0 : show "Letra A" ##

Case 1 : show "Letra B" ##

Case 2 : show "Cualquier otra letra" ##

#

3) ***Instrucciones Carlos Toro*** ***20171878***

a) Instrucción showDate:

Se utiliza para mostrar una fecha con formato "dd/mm/yyyy" como salida.

showdate fecha '#'

Por ejemplo:

fecha : 20201002 #

showdate fecha #

La salida de la anterior instrucción sería:

02/10/2020

b) Instrucción showage

Esta instrucción permite obtener y mostrar la resta de dos fechas (variables o números), esto es útil para calcular cuántos años han transcurrido desde la fecha inicial hasta la final. Este dato es muy utilizado en la emisión de reportes.

showage fechaIni ',' fechaFin '#'

Ejemplo:

showage 20201001 , 20211001 #

La salida será:

1 año

c) Instrucción toDate

Esta instrucción permite asignar una fecha a una variable de tipo fecha. La fecha es brindada en formato de cadena "dd/mm/yyyy".

toDate fecha ':' "dd/mm/yyyy" '#'

Un ejemplo de uso de esta instrucción es:

Declare fecha : date #

toDate fecha : "10/10/2020" #

showdate fecha #

La salida será:

"10/10/2020"

4) Instrucciones Angel Gonzales

20175488

a) Incremento unitario: esta instrucción permite incrementar en uno a la variable. Es decir, a la variable se le suma uno. Esta instrucción es muy útil para este lenguaje debido a que permitiría contar el número de cuentas, depósitos y retiros. Se utilizaría para un contador.

Inc numClientes #/* Está incrementando en uno la variable */

Inc numClientes, numDias, numCuentas #

/* Está incrementando en uno a estas variables hasta encontrar carácter '#' */

b) **Estructura iterativa (while)**: esta instrucción permite repetir un bloque de instrucciones hasta que se cumpla la condición que nosotros ingresemos.

● ● ● ● ● ●

While (condición) Loop :

(Se realizan las instrucciones)

• • • • •

```
# /* Fin de la estructura iterativa delimitada por # */
```

(Otras instrucciones que no pertenecen a la estructura)

● ● ● ● ● ●

c) **Estructura selectiva Condition** : esta instrucción sirve para verificar una operación lógica. Esta operación lógica devuelve true o false. Si es true entonces ejecutará unas instrucciones y si es falsa ejecutará otras instrucciones.

Condition (operación lógica) :

(Se realizan las instrucciones)

• • • • •

```
# /* Fin de la estructura Condition */
```