Correcciones:

Para los ámbitos de las funciones, más que concatenar una letra "X", era más claro utilizar para la concatenación el nombre de la función. Por la definición del lenguaje, nunca iba a poder haber funciones anidadas, de forma que solo hay dos opciones de ámbitos: el main y la función.

Intenté ensamblar el .asm que adjuntaron. No se puede. Tiene muchos errores.

En primer lugar, que utilizan caracteres inválidos. Por ejemplo, el # al inicio del nombre de una variable. Una vez que se soluciona eso, aparece todo lo siguiente:

pruebaSinErrores.asm(24) : error A2008: syntax error : Barbie

pruebaSinErrores.asm(25) : error A2008: syntax error : Mauro

pruebaSinErrores.asm(45) : error A2008: syntax error : ,

pruebaSinErrores.asm(47) : error A2008: syntax error : ,

pruebaSinErrores.asm(49) : error A2008: syntax error : ,

pruebaSinErrores.asm(56) : error A2044: invalid character in file

pruebaSinErrores.asm(57) : error A2044: invalid character in file

pruebaSinErrores.asm(63) : error A2044: invalid character in file

pruebaSinErrores.asm(70) : error A2008: syntax error : sumar

pruebaSinErrores.asm(73) : error A2008: syntax error : Label18

pruebaSinErrores.asm(77) : error A2008: syntax error : Label21

pruebaSinErrores.asm(79) : error A2044: invalid character in file

pruebaSinErrores.asm(84) : error A2084: constant value too large

pruebaSinErrores.asm(84) : error A2114: INVOKE argument type mismatch : argument : 3

pruebaSinErrores.asm(84) : error A2114: INVOKE argument type mismatch : argument : 2

pruebaSinErrores.asm(86) : error A2008: syntax error : Label27

pruebaSinErrores.asm(87) : error A2084: constant value too large

pruebaSinErrores.asm(87) : error A2114: INVOKE argument type mismatch : argument : 3

pruebaSinErrores.asm(87) : error A2114: INVOKE argument type mismatch : argument : 2

pruebaSinErrores.asm(88) : error A2008: syntax error : Label28

pruebaSinErrores.asm(91) : error A2008: syntax error : Label29

pruebaSinErrores.asm(34) : error A2006: undefined symbol : R1

pruebaSinErrores.asm(35) : error A2006: undefined symbol : \_a

pruebaSinErrores.asm(36) : error A2006: undefined symbol : R1

pruebaSinErrores.asm(37) : error A2006: undefined symbol : \_b

pruebaSinErrores.asm(38) : error A2006: undefined symbol : R1

pruebaSinErrores.asm(39) : error A2006: undefined symbol : \_a

pruebaSinErrores.asm(40) : error A2006: undefined symbol : R1

pruebaSinErrores.asm(41) : error A2006: undefined symbol : \_a

pruebaSinErrores.asm(42) : error A2006: undefined symbol : R1

pruebaSinErrores.asm(43) : error A2006: undefined symbol : \_b

pruebaSinErrores.asm(46) : error A2006: undefined symbol : \_c

pruebaSinErrores.asm(48) : error A2006: undefined symbol : \_e

pruebaSinErrores.asm(50) : error A2006: undefined symbol : \_d

pruebaSinErrores.asm(58) : error A2006: undefined symbol : \_c

pruebaSinErrores.asm(59) : error A2006: undefined symbol : R1

pruebaSinErrores.asm(60) : error A2006: undefined symbol : \_j

pruebaSinErrores.asm(62) : error A2006: undefined symbol : sumar

pruebaSinErrores.asm(64) : error A2006: undefined symbol : \_a

pruebaSinErrores.asm(65) : error A2006: undefined symbol : R2

pruebaSinErrores.asm(66) : error A2006: undefined symbol : \_a

pruebaSinErrores.asm(67) : error A2006: undefined symbol : Label18

pruebaSinErrores.asm(68) : error A2006: undefined symbol : R1

pruebaSinErrores.asm(69) : error A2006: undefined symbol : \_a

pruebaSinErrores.asm(71) : error A2006: undefined symbol : R2

pruebaSinErrores.asm(72) : error A2006: undefined symbol : \_a

pruebaSinErrores.asm(74) : error A2006: undefined symbol : Label21

pruebaSinErrores.asm(75) : error A2006: undefined symbol : R1

pruebaSinErrores.asm(76) : error A2006: undefined symbol : \_a

pruebaSinErrores.asm(78) : error A2006: undefined symbol : suma

pruebaSinErrores.asm(80) : error A2006: undefined symbol : \_a

pruebaSinErrores.asm(81) : error A2006: undefined symbol : R2

pruebaSinErrores.asm(82) : error A2006: undefined symbol : \_a

pruebaSinErrores.asm(83) : error A2006: undefined symbol : Label27

pruebaSinErrores.asm(85) : error A2006: undefined symbol : Label28

pruebaSinErrores.asm(89) : error A2006: undefined symbol : R1

pruebaSinErrores.asm(90) : error A2006: undefined symbol : \_a

Entre los problemas que aparecen, siguen utilizando los registros para operaciones con doubles. Por ejemplo,

c = 6,7.

Genera:

MOV R1,6,7

Lo cual, es incorrecto.

Asimismo, sigue habiendo RET en el medio del código, lo cual NO corresponde.

El código para las funciones está ubicado en cualquier lado. Los códigos correspondientes a las funciones tienen que estar separados. Al final, o en el sector donde tengan la parte de reporte de errores.

Tienen variables duplicadas.

Invocan labels que no existen.

Utilizando el siguiente código:

a,b,c : LONG.

a = 3.

b = 4.

c = a + b.

c = a - b.

c = a \* b.

c = a / b.

e,f,g: DOUBLE.

e = 3,0.

f = 4,0.

g = e + f.

g = e - f.

g = e \* f.

g = e / f.

Se genera el assembler, pero no es posible ensamblarlo. Obtengo esto:

test10\_ops.asm(34) : error A2044: invalid character in file

test10\_ops.asm(36) : error A2044: invalid character in file

test10\_ops.asm(40) : error A2044: invalid character in file

test10\_ops.asm(41) : error A2044: invalid character in file

test10\_ops.asm(44) : error A2008: syntax error : ,

test10\_ops.asm(45) : error A2044: invalid character in file

test10\_ops.asm(47) : error A2044: invalid character in file

test10\_ops.asm(49) : error A2008: syntax error : ,

test10\_ops.asm(52) : error A2008: syntax error : ,

test10\_ops.asm(53) : error A2044: invalid character in file

test10\_ops.asm(55) : error A2044: invalid character in file

test10\_ops.asm(57) : error A2008: syntax error : ,

test10\_ops.asm(59) : error A2008: syntax error : ,

test10\_ops.asm(64) : error A2044: invalid character in file

test10\_ops.asm(65) : error A2044: invalid character in file

test10\_ops.asm(70) : error A2044: invalid character in file

test10\_ops.asm(71) : error A2044: invalid character in file

test10\_ops.asm(75) : error A2044: invalid character in file

test10\_ops.asm(77) : error A2044: invalid character in file

test10\_ops.asm(84) : error A2044: invalid character in file

test10\_ops.asm(85) : error A2044: invalid character in file

test10\_ops.asm(87) : error A2008: syntax error : Label20

test10\_ops.asm(28) : error A2006: undefined symbol : R1

test10\_ops.asm(29) : error A2006: undefined symbol : \_a

test10\_ops.asm(30) : error A2006: undefined symbol : R1

test10\_ops.asm(31) : error A2006: undefined symbol : \_b

test10\_ops.asm(32) : error A2006: undefined symbol : R1

test10\_ops.asm(33) : error A2006: undefined symbol : R1

test10\_ops.asm(37) : error A2006: undefined symbol : \_c

test10\_ops.asm(38) : error A2006: undefined symbol : R1

test10\_ops.asm(39) : error A2006: undefined symbol : R1

test10\_ops.asm(42) : error A2006: undefined symbol : \_c

test10\_ops.asm(43) : error A2006: undefined symbol : R1

test10\_ops.asm(48) : error A2006: undefined symbol : \_c

test10\_ops.asm(51) : error A2006: undefined symbol : R1

test10\_ops.asm(56) : error A2006: undefined symbol : \_c

test10\_ops.asm(58) : error A2006: undefined symbol : \_e

test10\_ops.asm(60) : error A2006: undefined symbol : \_f

test10\_ops.asm(66) : error A2006: undefined symbol : \_g

test10\_ops.asm(72) : error A2006: undefined symbol : \_g

test10\_ops.asm(78) : error A2006: undefined symbol : \_g

test10\_ops.asm(86) : error A2006: undefined symbol : \_g

Si la operación: MOV R1,3 intentaba cargar en el registro 1 (AX) el valor 3, no es correcto. Dado que el assembler reconoce los registros por su nombre AX, BX, … .

Usan variables que no están definidas, por ejemplo MOV #aux2,R1.

El control de overflow se tiene que hacer directamente después de la operación, no primero el mov y luego el control, dado que si hubo overflow, no hay que hacer el almacenamiento del resultado.

En algún lugar se generó la instrucción MOV R2,\_bCMP 0,R2. No se corresponde la cantidad de parámetros.

Después de una división aparece un JO por overflow, eso solo lo tienen que controlar para las sumas.

En algún momento empiezan a aparecer variables declaradas como e@Variable. Por qué la diferencia entre los nombres de las variables?

La sintaxis de las labels es incorrecta. Tienen que llevar ":" luego del nombre del label.

El siguiente código:

e,f,g: DOUBLE.

e = 3,0.

f = 4,0.

g = e + f.

g = e - f.

g = e \* f.

g = e / f.

Genera:

MOV R1,3,0

MOV \_e,R1

MOV R1,4,0

MOV \_f,R1

FLD e@Variable

FLD e@Variable

FADD

FSTP #aux2

MOV R1,#aux2

MOV \_g,R1

FLD e@Variable

FLD e@Variable

FSUB

FSTP #aux4

MOV R1,#aux4

MOV \_g,R1

FLD e@Variable

FLD e@Variable

MOV #aux6,R1

JO \_overflow

MOV R1,#aux6

MOV \_g,R1

FTST

JE \_division\_cero

FLD e@Variable

FLD e@Variable

FDIV

FSTP #aux8

MOV R1,#aux8

MOV \_g,R1

Label10

invoke ExitProcess, 0

Usan asignaciones en registros cuando las variables son de tipo double. Eso no corresponde. El assembler genera los siguientes errores:

test10\_ops.asm(25) : error A2008: syntax error : ,

test10\_ops.asm(27) : error A2008: syntax error : ,

test10\_ops.asm(32) : error A2044: invalid character in file

test10\_ops.asm(33) : error A2044: invalid character in file

test10\_ops.asm(38) : error A2044: invalid character in file

test10\_ops.asm(39) : error A2044: invalid character in file

test10\_ops.asm(43) : error A2044: invalid character in file

test10\_ops.asm(45) : error A2044: invalid character in file

test10\_ops.asm(52) : error A2044: invalid character in file

test10\_ops.asm(53) : error A2044: invalid character in file

test10\_ops.asm(55) : error A2008: syntax error : Label10

test10\_ops.asm(26) : error A2006: undefined symbol : \_e

test10\_ops.asm(28) : error A2006: undefined symbol : \_f

test10\_ops.asm(34) : error A2006: undefined symbol : \_g

test10\_ops.asm(40) : error A2006: undefined symbol : \_g

test10\_ops.asm(46) : error A2006: undefined symbol : \_g

test10\_ops.asm(54) : error A2006: undefined symbol : \_g

El overflow está mal hecho. Cuando se opera en la pila los flags no se copian de forma directa. Tienen que usar las instrucciones que aparecen en los slides para eso. De todas formas, el control de overflow en double no es tan directo como el de los tipos de punto fijo, dado que las operaciones no funcionan de la misma forma. Por ejemplo, pueden generar el valor de Infinity y -Infinity. Con lo que el control de overflow en punto flotante implica detección de infinitos o control de mayor o menor que los límites del rango.

El código:

OUT("Imprime").

Genera el siguiente assembler:

.386

.model flat, stdcall

option casemap :none

include \masm32\include\windows.inc

include \masm32\include\kernel32.inc

include \masm32\include\user32.inc

includelib \masm32\lib\kernel32.lib

includelib \masm32\lib\user32.lib

.data

\_\_MIN DD  -2147483648

\_\_MAX DD  2147483647

\_msjDC DB "Error: Division por cero", 0

\_msjOverflow DB "Error: Overflow", 0

"Imprime" DB "Imprime", 0

.code

\_division\_cero:

invoke MessageBox, NULL, addr \_msjDC, addr \_msjDC, MB\_OK

invoke ExitProcess, 0

\_overflow:

invoke MessageBox, NULL, addr \_msjOverflow, addr \_msjOverflow, MB\_OK

invoke ExitProcess, 0

start:

invoke MessageBox, NULL, addr "Imprime", addr "Imprime", MB\_OK

Label1

invoke ExitProcess, 0

end start

Cuando quiero ensamblar genera:

test10\_out.asm(14) : error A2008: syntax error : Imprime

test10\_out.asm(23) : error A2084: constant value too large

test10\_out.asm(23) : error A2114: INVOKE argument type mismatch : argument : 3

test10\_out.asm(23) : error A2114: INVOKE argument type mismatch : argument : 2

test10\_out.asm(24) : error A2008: syntax error : Label1

Ninguno de los códigos que probé dio lugar a un ensamblado exitoso. Si no arreglan la generación del assembler, no puedo seguir ni corrigiendo ni haciendo pruebas.

Por favor, hagan las correcciones a la brevedad. Así puedo seguir corrigiendo.

Saludos,

Antonela