\*------------------------------------------------------------------------------------------------

\* Programmer: Quang Trinh

\* Class Account: cssc0855

\* Assignment or Title: Program #3

\* Filename: prog3.s

\* Date completed: 04.19.18

\*------------------------------------------------------------------------------------------------

\* Problem statement: The program will take a currency input, then

\* output a list of bills and coins equal to the

\* input.

\* Input: A currency.

\* Output: List of bills and coins.

\* Error conditions tested: N/A

\* Included files: prog3.s

\* Method and/or pseudocode: We will use branching and looping method.

\* First, we break the number into whole number

\* part and fraction part. Then break them into

\* bills and coins using branching and looping.

\* References: N/A

\*-------------------------------------------------------------------------------------------------

\*

ORG $0

DC.L $3000 \* Stack pointer value after a reset

DC.L start \* Program counter value after a reset

ORG $3000 \* Start at location 3000 Hex

\*

\*-------------------------------------------------------------------------------------------------

\*

#minclude /home/cs/faculty/riggins/bsvc/macros/iomacs.s

#minclude /home/cs/faculty/riggins/bsvc/macros/evtmacs.s

\*

\*-------------------------------------------------------------------------------------------------

\*

\* Register use

\*

\*-------------------------------------------------------------------------------------------------

\*

start: initIO \* Initialize (required for I/O)

setEVT \* Error handling routines

\* initF \* For floating point macros only

lineout title

lineout prompt1

linein buffer \* User input

\* Check decimal place

lea buffer,A1

move.l D0,D2 \* String length

adda.l D2,A1 \* Jump A1 to the end of string

suba.l #3,A1 \* A1 points to decimal place

cmpi.b #'.',(A1)

bne ERROR

\* Check valid digits

lea buffer,A0

clr.w D1 \* Loop control

subq.l #3,D2

For: cmp.w D2,D1 \* Loop from 0 to the decimal place

bge EndFor

cmpi.b #'0',(A0)

blo ERROR

cmpi.b #'9',(A0)+

bhi ERROR

addq.w #1,D1 \* Increment loop control

bra For

EndFor:

\* Separate bills and coins

cvta2 buffer,D2

move.l D0,D6 \* Bills -> D6

lea buffer,A1

adda.l D2,A1

addq.l #1,A1

cvta2 (A1),#2 \* A1 points to Coins part

lineout prompt2

\* Bills seperation

move.w D6,D0 \* Bills -> D0

divu #100,D0 \* Amount of $100

ext.l D0

move.l D0,D7 \* Amount -> D7

cmp.w #0,D0 \* Compare amount to 0

ble CheckFifty

cvt2a a100,#3

lineout a100

mulu #100,D7

sub.w D7,D6

CheckFifty:

move.w D6,D0

divu #50,D0

ext.l D0

move.l D0,D7

cmp.w #0,D0

ble CheckTwenty

cvt2a a50,#3

lineout a50

mulu #50,D7

sub.w D7,D6

CheckTwenty:

move.w D6,D0

divu #20,D0

ext.l D0

move.l D0,D7

cmp.w #0,D0

ble CheckTen

cvt2a a20,#3

lineout a20

mulu #20,D7

sub.w D7,D6

CheckTen:

move.w D6,D0

divu #10,D0

ext.l D0

move.l D0,D7

cmp.w #0,D0

ble CheckFive

cvt2a a10,#3

lineout a10

mulu #10,D7

sub.w D7,D6

CheckFive:

move.w D6,D0

divu #5,D0

ext.l D0

move.l D0,D7

cmp.w #0,D0

ble CheckOne

cvt2a a05,#3

lineout a05

mulu #5,D7

sub.w D7,D6

CheckOne:

move.w D6,D0

ext.l D0

move.l D0,D7

cmp.w #0,D0

ble DONE

cvt2a a01,#3

lineout a01

DONE:

cvta2 (A1),#2

move.l D0,D5 \* Coins part

\* Separate Coins

divu #50,D0

ext.l D0

move.l D0,D7

cmp.w #0,D0

ble Check25

cvt2a a50c,#3

lineout a50c

mulu #50,D7

sub.w D7,D5

Check25:

move.l D5,D0

divu #25,D0

ext.l D0

move.l D0,D7

cmp.w #0,D0

ble Check10

cvt2a a25c,#3

lineout a25c

mulu #25,D7

sub.w D7,D5

Check10:

move.l D5,D0

divu #10,D0

ext.l D0

move.l D0,D7

cmp.w #0,D0

ble Check5

cvt2a a10c,#3

lineout a10c

mulu #10,D7

sub.w D7,D5

Check5:

move.l D5,D0

divu #5,D0

ext.l D0

move.l D0,D7

cmp.w #0,D0

ble Check1

cvt2a a5c,#3

lineout a5c

mulu #5,D7

sub.w D7,D5

Check1:

move.l D5,D0

ext.l D0

move.l D0,D7

cmp.w #0,D0

ble ENDPROG

cvt2a a1c,#3

lineout a1c

ENDPROG:

break

ERROR:

lineout warning \* If no decimal place, print error

break \* Terminate the program

\*--------------------------------------------------------------------------------------------------

\* Storage declarations

title: dc.b 'Program #3, cssc0855, Quang Trinh',0

prompt1: dc.b 'Enter an amount in U.S. Dollars (no $ sign): ',0

warning: dc.b 'Sorry, invalid entry.',0

buffer: ds.b 80

prompt2: dc.b 'That amount is: ',0

a100: ds.b 3

hundred: dc.b ' x $100',0

a50: ds.b 3

fifty: dc.b ' x $50 ',0

a20: ds.b 3

twenty: dc.b ' x $20 ',0

a10: ds.b 3

ten: dc.b ' x $10 ',0

a05: ds.b 3

five: dc.b ' x $5 ',0

a01: ds.b 3

one: dc.b ' x $1 ',0

a50c: ds.b 3

fiftycent: dc.b ' x 50',$A2,0

a25c: ds.b 3

quarter: dc.b ' x 25',$A2,0

a10c: ds.b 3

dime: dc.b ' x 10',$A2,0

a5c: ds.b 3

nickle: dc.b ' x 5',$A2,0

a1c: ds.b 3

cent: dc.b ' x 1',$A2,0

end