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

\* Programmer: Arthur Flores

\* Class Account: masc0200

\* Assignment or Title: Program 4

\* Filename: prog4.s

\* Date completed: 12/09/14

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

\* Problem statement: Print reduced fraction

\* Input: a fraction (X/Y)

\* Output: reduced fraction (X/Y)

\* Error conditions tested:

\* Included files: prog4.s, reduce.s, GCD.s

\* Method and/or pseudocode: subroutines and recursion

\* References:

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

\*

getInput: EQU $6000

reduce: EQU $8000

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/ma/cs237/bsvc/iomacs.s

#minclude /home/ma/cs237/bsvc/evtmacs.s

\*

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

\*

\* Register use

\*

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

\*

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

setEVT \* Error handling routines

\* initF \* For floating point macros only

lineout title \*print title

pea denominator \*push denominator onto stack

pea numerator \*push numerator onto stack

pea message \*push message onto stack

bra get \*branch to get

ivalid: lineout new \*tells user to enter a valid input

get: jsr getInput \*go to getInput

bvs ivalid \*if v is set branch to invalid

adda.l #12,SP \*get rid of garbage

\*\*\*\*\*\*\*\*\*\*\*

pea denominator \*push address of denominator

pea numerator \*push address of numerator

jsr reduce \*go to reduce

adda.l #8,SP \*get rid of garbage

\*\*\*\*\*\*\*\*\*\*\*

lea fraction,A0 \*load fraction address into A0

move.w numerator,D0\*move numerator into D0

ext.l D0 \*extend D0

cvt2a (A0),#5 \*convert and store into fraction

stripp (A0),#5 \*strip numerator in fraction

adda.l D0,A0 \*add length of numerator to A0

move.b #'/',(A0)+ \*add a backslash in fraction

move.w denominator,D0 \*move denominator into D0

ext.l D0 \*extend D0

cvt2a (A0),#5 \*cvt denom to asci in fraction

stripp (A0),#5 \*stripp denominator

adda.l D0,A0 \*add length of denomi into fraction

clr.b (A0) \*null terminate fraction

lineout answer \*print reduced fraction

\*\*\*\*\*\*\*\*\*\*\*

break \* Terminate execution

\*

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

\* Storage declarations

title: dc.b 'Program #4, Arthur Flores, masc0200',0

message: dc.b 'Enter a fraction to reduce:',0

numerator: ds.w 1

denominator: ds.w 1

new: dc.b 'Please enter a valid fraction',0

answer: dc.b 'The reduced fraction is '

fraction: ds.b 12

end