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

\* Programmer: Arthur Flores

\* Class Account: masc0200

\* Assignment or Title: Assignment 2

\* Filename: prog2.s

\* Date completed: 10/30/14

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

\* Problem statement:

\* Input:

\* Output:

\* Error conditions tested:

\* Included files:

\* Method and/or pseudocode:

\* References:

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

\*

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

\* D0= user input,N,P

\* D1=H,P

\* D2=G,N

\* D3=D,J

\* D4=E,L

\* D5=B,M

\* D6=C,K

\* D7=A

\*

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

\*

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

setEVT \* Error handling routines

\* initF \* For floating point macros only

lineout title \*prints title

lineout title2 \*prints second title

lineout prompt \*prints prompt

linein buffer \*reads user input

cvta2 buffer,D0 \*converts input into 2's complement

move.l D0,D7 \*move D0 to D7

divu #19,D7 \*divides 19 into user input

swap D7 \*switches order so remainder can be used

ext.l D7 \*make D7=A just the remainder #

move.l D0,D6 \*move D0 into D6 and make D6=C

divu #100,D6 \*divide D6 by 100

move.w D6,D5 \*move D6 into D5 for D5=B

swap D6 \*switch remainder and quotient

ext.l D6 \*make D6=C the remainder

move.l D5,D4 \*copy D5 into D4 for dividing

divu #4,D4 \*divide D4 by 4

move.w D4,D3 \*move quotient into D3=D

swap D4 \*switches quotient and remainder

ext.l D4 \*makes D4=E

move.l D5,D2 \*copies D5=B into D2

mulu #8,D2 \*multiply D2=B by 8

addi.l #13,D2 \*add 13 to D2=B

divu #25,D2 \*divide D2=B by 25

ext.l D2 \*ignore remainder and store quotient

move.l D7,D1 \*copy D7=A into D1

mulu #19,D1 \*multiply D1 by 19

add.l D5,D1 \*add D5=B to D1 and store in D1

sub.l D3,D1 \*subtract D3=D from D1.

sub.l D2,D1 \*subtract D2=G from D1

addi.l #15,D1 \*add 15 to D1

divu #30,D1 \*divide D1 by 30

swap D1 \*switch quotient and remainder

ext.l D1 \*make D1 just remainder. D1=H

move.l D1,D5 \*move D1=H into D5

mulu #11,D5 \*multiply D5 by 11

add.l D7,D5 \*add D7=A to D5

divu #319,D5 \*divide D5 by 319

ext.l D5 \*stores only quotient. D5=M

divu #4,D6 \*divide D6=C by 4

move.w D6,D3 \*move quotient into D3

ext.l D3 \*extend D3=J to get rid of old content

swap D6 \*switch quotient and remainder

ext.l D6 \*store only remainder in D6=K

mulu #2,D4 \*multiply 2 by D4=E

mulu #2,D3 \*multiply 2 by D3=J

add.l D3,D4 \*add D4 and D3 and store in D4

sub.l D6,D4 \*subtract D6=K from D4

sub.l D1,D4 \*subtract D1=H from D4

add.l D5,D4 \*add D5=M to D4

addi.l #32,D4 \*add 32 to D4

divu #7,D4 \*divide D4 by 7

swap D4 \*switch quotient and remainder

ext.l D4 \*store only remainder D4=L

move.l D1,D2 \*move D1=H into D2

sub.l D5,D2 \*subtract D5=M from D2

add.l D4,D2 \*add D4=L to D2

addi.l #90,D2 \*add 90 to D2

divu #25,D2 \*divide D2 by 25

ext.l D2 \*only store quotient. D2=N

sub.l D5,D1 \*subtract D5=M from D1=H

add.l D4,D1 \*add D4=L to D1

add.l D2,D1 \*add D2=N to D1

addi.l #19,D1 \*add 19 to D1

divu #32,D1 \*divide D1 by 32

swap D1 \*switch quotient and remainder

ext.l D1 \*stores on the remainder. D1=P

move.l D2,D0 \*move N=month into D0

cvt2a month,#2 \*convert and put 2 bytes of D0 into month

move.l D1,D0 \*move P=day into D0

cvt2a day,#2 \*convert and put 2 bytes of D0 into day

lineout answer \*print answer

break \* Terminate execution

\*

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

\* Storage declarations

title: dc.b '\*\*Program #2, Arthur Flores, masc0200\*\*',0

title2: dc.b '\*\*Program to calculate Easter Sunday\*\*',0

prompt: dc.b 'Enter the year (YYYY):',0

buffer: ds.b 81

answer: dc.b 'Easter Sunday is on: '

month: ds.b 2

bslash: dc.b '/'

day: ds.b 2

dc.b 0

end