# Rubric

name: Denary to Roman Numerals

init: qinit

accept: qfin

//Currently works only with input symbols 0,1,2

//Example tape input: 1201

//Intermediate |M|CC||I|

# qinit

qinit,0

qinit,0,>

qinit,1

qinit,1,>

qinit,2

qinit,2,>

qinit,3

qinit,3,>

qinit,4

qinit,4,>

qinit,5

qinit,5,>

qinit,\_

return,],<

# readNextDigit

//read digit

readNextDigit,0

write0,[,>

readNextDigit,1

write1,[,>

readNextDigit,2

write2,[,>

readNextDigit,3

write3,[,>

readNextDigit,4

write4,[,>

readNextDigit,5

write5,[,>

readNextDigit,]

gotoEnd,\_,>

## writeN

## write0

//write0

write0,0 //0

write0,0,>

write0,1

write0,1,>

write0,2

write0,2,>

write0,3

write0,3,>

write0,4

write0,4,>

write0,5

write0,5,>

write0,]

write0,],>

write0,I

write0,I,>

write0,V

write0,V,>

write0,|

write0,|,>

write0,\_

return,|,< //Write0, uniquely, moves left here

## write1

//write1

write1,0

write1,0,>

write1,1

write1,1,>

write1,2

write1,2,>

write1,3

write1,3,>

write1,4

write1,4,>

write1,5

write1,5,>

write1,]

write1,],>

write1,I

write1,I,>

write1,V

write1,V,>

write1,|

write1,|,>

write1,\_

write0,I,>

## write2

//write2

write2,0

write2,0,>

write2,1

write2,1,>

write2,2

write2,2,>

write2,3

write2,3,>

write2,4

write2,4,>

write2,5

write2,5,>

write2,]

write2,],>

write2,I

write2,I,>

write2,V

write2,V,>

write2,|

write2,|,>

write2,\_

write1,I,>

## write3

//write3

write3,0

write3,0,>

write3,1

write3,1,>

write3,2

write3,2,>

write3,3

write3,3,>

write3,4

write3,4,>

write3,5

write3,5,>

write3,]

write3,],>

write3,I

write3,I,>

write3,V

write3,V,>

write3,|

write3,|,>

write3,\_

write2,I,>

//write4

write4,0

write4,0,>

write4,1

write4,1,>

write4,2

write4,2,>

write4,3

write4,3,>

write4,4

write4,4,>

write4,5

write4,5,>

write4,]

write4,],>

write4,I

write4,I,>

write4,V

write4,V,>

write4,|

write4,|,>

write4,\_

write5,I,>

//write5

write5,0

write5,0,>

write5,1

write5,1,>

write5,2

write5,2,>

write5,3

write5,3,>

write5,4

write5,4,>

write5,5

write5,5,>

write5,]

write5,],>

write5,I

write5,I,>

write5,V

write5,V,>

write5,|

write5,|,>

write5,\_

write0,V,>

# return

//return

return,0

return,0,<

return,1

return,1,<

return,2

return,2,<

return,3

return,3,<

return,4

return,4,<

return,5

return,5,<

return,I

return,I,<

return,V

return,V,<

return,|

return,|,<

return,]

return,],<

return,[

readNextDigit,\_,>

return,\_

readNextDigit,\_,>

//gotoEnd - only after deleting input.

# gotoEnd

gotoEnd,|

gotoEnd,|,>

gotoEnd,I

gotoEnd,I,>

gotoEnd,V

gotoEnd,V,>

gotoEnd,\_

symbols0,\_,<

# symbolsN

//symbols1

symbols0,|

symbols1,|,<

symbols1,I

symbols1,I,<

symbols1,V

symbols1,V,<

symbols1,|

symbols10,|,<

symbols1,\_

removeBars,|,<

symbols10,I

symbols10,X,<

symbols10,V

symbols10,L,<

symbols10,|

symbols100,|,<

symbols10,\_

removeBars,|,<

symbols100,I

symbols100,C,<

symbols100,V

symbols100,D,<

symbols100,|

symbols1000,|,<

symbols100,\_

removeBars,|,<

symbols1000,I

symbols1000,M,<

symbols1000,\_

removeBars,|,<

# removeBars

//Remove bars

removeBars,\_

removeBars,\_,>

removeBars,|

removeBars,\_,>

removeBars,I

moveI,\_,>

removeBars,V

moveV,\_,>

removeBars,X

moveX,\_,>

removeBars,L

moveL,\_,>

removeBars,C

moveC,\_,>

removeBars,D

moveD,\_,>

removeBars,M

moveM,\_,>

# nextChar

nextChar,|

nextChar,|,>

nextChar,\_

return2,\_,<

nextChar,I

moveI,|,>

nextChar,V

moveV,|,>

nextChar,X

moveX,|,>

nextChar,L

moveL,|,>

nextChar,C

moveC,|,>

nextChar,D

moveD,|,>

nextChar,M

moveM,|,>

# moveN

//moveI

moveI,|

nextChar,I,>

moveI,I

writeI,I,<

moveI,V

writeI,V,<

moveI,X

writeI,X,<

moveI,\_

writeI,\_,<

//moveV

moveV,|

nextChar,V,>

moveV,I

writeV,I,<

moveV,\_

writeV,\_,<

//moveX

moveX,|

nextChar,X,>

moveX,I

writeX,I,<

moveX,V

writeX,V,<

moveX,X

writeX,X,<

moveX,L

writeX,L,<

moveX,C

writeX,C,<

moveX,\_

writeX,\_,<

//moveL

moveL,|

nextChar,L,>

moveL,I

writeL,I,<

moveL,V

writeL,V,<

moveL,X

writeL,X,<

moveL,\_

writeL,\_,<

//moveC

moveC,|

nextChar,C,>

moveC,I

writeC,I,<

moveC,V

writeC,V,<

moveC,X

writeC,X,<

moveC,L

writeC,L,<

moveC,C

writeC,C,<

moveC,D

writeC,D,<

moveC,M

writeC,M,<

moveC,\_

writeC,\_,<

//moveD

moveD,|

nextChar,D,>

moveD,I

writeD,I,<

moveD,V

writeD,V,<

moveD,X

writeD,X,<

moveD,L

writeD,L,<

moveD,\_

writeD,\_,<

//moveM

moveM,|

nextChar,M,>

moveM,I

writeM,I,<

moveM,V

writeM,V,<

moveM,X

writeM,X,<

moveM,L

writeM,L,<

moveM,C

writeM,C,<

moveM,D

writeM,D,<

moveM,M

writeM,M,<

moveM,\_

writeM,\_,<

# writeS

//write symbol

writeI,|

nextChar,I,>

writeI,\_

nextChar,I,>

writeV,|

nextChar,V,>

writeV,\_

nextChar,V,>

writeX,|

nextChar,X,>

writeX,\_

nextChar,X,>

writeL,|

nextChar,L,>

writeL,\_

nextChar,L,>

writeC,|

nextChar,C,>

writeC,\_

nextChar,C,>

writeD,|

nextChar,D,>

writeD,\_

nextChar,D,>

writeM,|

nextChar,M,>

writeM,\_

nextChar,M,>

# return2

//return2

return2,I

return2,I,<

return2,V

return2,V,<

return2,X

return2,X,<

return2,L

return2,L,<

return2,C

return2,C,<

return2,D

return2,D,<

return2,M

return2,M,<

return2,|

continue,|,<

return2,\_

qfin,\_,>

# continue

//continue

continue,I

continue,I,<

continue,V

continue,V,<

continue,X

continue,X,<

continue,L

continue,L,<

continue,C

continue,C,<

continue,D

continue,D,<

continue,M

continue,M,<

continue,|

continue,|,<

continue,\_

removeBars,\_,>