```
        lineout  head

reprmpt:
        clr.l    D1
        clr.l    D2
        lineout prompt
        linein   buff

        lea      buff,A1
        clr.l    D3
        clr.l    D4

top:
        cmpi.b   #' ',(A1) *gets length of input
        beq      check
        addi.l   #1,D3
        adda.l   #1,A1
        bra      top



check:

        move.l   D0,D2          *checks if length is 3-5
        clr.l    D0
        cmpi.b   #3,D2
        blt      bot
        cmpi.b   #5,D2
        bgt      bot
        adda.l   #1,A1
        bra      new
```

```
bot:
        lineout inclen
        bra     reprmpt

new:
        lea     buff,A1
        bra     next
next:
        cmpi.b  #'0',(A1)       *checks input to verify it is a number
        blt     bote
        cmpi.b  #'9',(A1)
        bgt     bote
        adda.l  #1,A1
        addi.l  #1,D4
        cmp.l   D4,D2
        bne     next
        bra     prime



bote:
        lineout notnum          *prints not a number
        bra     reprmpt


prime:
        clr.l   D4
        clr.l   D1
        cvta2   buff,D2
        move.l  D0,D4
        addi.l  #1,D1
        clr.l   D5
loop:
        addi.l  #1,D1           *loop to check if prime
        move.l  D4,D5
        divs    D1,D5
        swap    D5
        cmp.l   D1,D4
```

```
        beq     iPrime
        cmpi.w  #0,D5
        beq     nPrime
        bra     loop


iPrime:
        lea     strt,A5         *prints the # is prime
        lea     good,A3
        adda.l  #11,A5
        cvt2a   (A5),D2
        adda.l  D2,A5
        move.b  #' ',0(A5)
        move.b  0(A3),1(A5)
        move.b  1(A3),2(A5)
        move.b  2(A3),3(A5)
        move.b  3(A3),4(A5)
        move.b  4(A3),5(A5)
        move.b  5(A3),6(A5)
        move.b  6(A3),7(A5)
        move.b  7(A3),8(A5)
        move.b  8(A3),9(A5)
        move.b  ',0',10(A5)
        lineout strt
        bra     end
nPrime:
        lea     strt,A5         *prints # is not prime
        lea     bad,A3
        adda.l  #11,A5
        cvt2a   (A5),D2
        adda.l  D2,A5
        move.b  #' ',0(A5)
        move.b  0(A3),1(A5)
        move.b  1(A3),2(A5)
        move.b  2(A3),3(A5)
        move.b  3(A3),4(A5)
        move.b  4(A3),5(A5)
        move.b  5(A3),6(A5)
        move.b  6(A3),7(A5)
```

```
        move.b 7(A3),8(A5)
        move.b 8(A3),9(A5)
        move.b 9(A3),10(A5)
        move.b 10(A3),11(A5)
        move.b 11(A3),12(A5)
        move.b 12(A3),13(A5)
        move.b ',0',14(A5)
        lineout  strt
        bra      end


end:
        break

            * Terminate execution
*
*---------------------------------------------------------------
*       Storage declarations
head:   dc.b     'Program #2, Nathan Azoulay, cssc0406',0
prompt:          dc.b     'Enter a 3-5 digit number:',0
buff:   ds.b     80
inclen: dc.b     'Incorrect input length.',0
notnum:          dc.b     'Not a number.',0
good:   dc.b     'is prime.',0
bad:    dc.b     'is not prime.',0
strt:   dc.b     'The number '

      end
```