Problem statement:

Input a decimal number (3or more digits) using indec procedure.Output the number using outdec procedure.Check if the number is prime.

Code:

```
.MODEL SMALL
.STACK
.DATA

  NUM DB ?
  MSG2 DB 10,13,'NOT PRIME
  MSG3 DB 10,13,'PRIME
.CODE
MAIN PROC
    MOV AX,@DATA
    MOV DS,AX
    ;INPUT A NUMBER
    CALL INDEC  ;NUMBER IN AX
    PUSH AX     ;SAVE NUMBER
    ;MOVE CURSOR TO A NEWLINE
    MOV AH,2
    MOV DL,0DH
    INT 21H
    MOV DL,0AH
    INT 21H
    ;OUTPUT THE NUMBER
    POP AX      ;RETRIEVE NUMBER
    CALL OUTDEC
    ;DOS EXIT
    MOV AH,4CH
    INT 21H
MAIN ENDP

;OUTDEC
OUTDEC PROC
    ;PRINTS AX AS A SIGNED DECIMAL INTEGER
    ;INPUT: AX
    ;OUTPUT: NONE
    PUSH AX     ;SAVE REGISTERS
    PUSH BX
    PUSH CX
    PUSH DX
    ;IF AX<0
    OR AX,AX    ;AX<0?
    JGE @END_IF1;NO, >0
    ;THEN
    PUSH AX     ;SAVE NUMBER
    MOV DL,'-'  ;GET '-'
    MOV AH,2    ;PRINT CHAR FUNCTION
    INT 21H     ;PRINT '-'
    POP AX      ;GET AX BACK
    NEG AX      ;AX = -AX
```

```
@END_IF1:
;GET DECIMAL DIGITS
    XOR CX,CX    ;CX COUNTS DIGITS
    MOV BX,10D   ;BX HAS DIVISOR
@REPEAT1:
    XOR DX,DX    ;PREPARE HIGH WORD OF DIVIDEND
    DIV BX       ;AX=QUOTIENT, DX=REMAINDER
    PUSH DX      ;SAVE REMAINDER ON STACK
    INC CX       ;COUNT=COUNT+1
;UNTIL
    OR AX,AX     ;QUOTIENT=0?
    JNE @REPEAT1;NO, KEEP GOING
;CONVERT DIGITS TO CHARACTERS AND PRINT
    MOV AH,2     ;PRINT CHAR FUNCTION
;FOR COUNT TIMES DO
@PRINT_LOOP:
    POP DX       ;DIGIT IN DL
    OR DL,30H    ;CONVERT TO CHARACTER
    INT 21H      ;PRINT DIGIT
    LOOP @PRINT_LOOP;LOOP UNTIL DONE
;END_FOR
    POP DX       ;RESTORE REGISTERS
    POP CX
    POP BX
    POP AX
    RET
OUTDEC ENDP



;INDEC

INDEC PROC
    ;READS A NUMBER IN RANGE 032768 TO 32767
    ;INPUT: NONE
    ;OUTPUT: AX=BINARY EQUIVALENT OF NUMBER
    PUSH BX      ;SAVE REGISTERS USED
    PUSH CX
    PUSH DX
;PRINT PROMPT
@BEGIN:
    MOV AH,2
    MOV DL,'?'
    INT 21H      ;PRINT '?'
;TOTAL=0
    XOR BX,BX    ;BX HOLDS TOTAL
    ;NEGATIVE=FALSE
    XOR CX,CX    ;CX HOLDS SIGN
;READ A CHARACTER
    MOV AH,1
    INT 21H      ;CHARACTER IN AL
    ;CASE CHARACTER OF
    CMP AL,'-'   ;MINUS SIGN?
```

```
    JE @MINUS    ;YES, SET SIGN
    CMP AL,'+'   ;PLUS SIGN
    JE  @PLUS    ;UES, GET ANOTHER CHARACTER
    JMP @REPEAT2;START PROCESSING CHARACTERS
@MINUS:
    MOV CX,1     ;NEGATIVE=TRUE
@PLUS:
    INT 21H      ;READ A CHARACTER
;END_CASE
@REPEAT2:
;IF CHARACTER IS BETWEEN '0' AND '9'
    CMP AL,'0'   ;CHARACTER>='0'?
    JNGE @NOT_DIGIT ;NO, ILLEGAL CHARACTER
;THEN CONVERT CHARACTER TO A DIGIT
    AND AX,000FH ;CONVERT TO DIGIT
    PUSH AX      ;SAVE ON STACK
;TOTAL = TOTAL*10+DIGIT
    MOV AX,10    ;GET 10
    MUL BX       ;AX=TOTAL*10
    POP BX       ;RETRIEVE DIGIT
    ADD BX,AX    ;TOTAL=TOTAL*10+DIGIT
;READ A CHARACTER
    MOV AH,1
    INT 21H
    CMP AL,0DH   ;CARRIAGE RETURN?
    JNE @REPEAT2;NO, KEEP GOING
;UNTIL CR
    MOV AX,BX    ;STORE NUMBER IN AX

    ;OUR NUMBER IS NOW IN AX
    ;CHECKING PRIME NUMBER

   PUSH AX

   MOV NUM,AL

   CMP AL,1
   JLE LBL2
   MOV AH,00
   CMP AL,3
   JLE LBL3
   MOV AH,00

   MOV CL,2
   DIV CL
   MOV CL,AL ;NOW QUOTIENT IS IN CL

LBL1:

    MOV AH,00
    MOV AL,NUM
    DIV CL

    CMP AH,00;CHECKING IF REMAINDER IS 0
```

```
    JZ LBL2
    DEC CL
    CMP CL,1
    JNE LBL1
    JMP LBL3

LBL2:
    MOV AH,9
    LEA DX,MSG2
    INT 21H
    JMP AFTER

LBL3:
    MOV AH,9
    LEA DX,MSG3
    INT 21H

AFTER:

;IF NEGATIVE
    OR CX,CX     ;NEGATIVE NUMBER
    JE @EXIT     ;NO, EXIT
;THEN
    NEG AX       ;YES, NEGATE
;END_IF
@EXIT:
    POP AX
    POP DX       ;RESTORE REGISTERS
    POP CX
    POP BX
    RET          ;AND RETURN
;HERE IF ILLEGAL CHARACTER ENTERED
@NOT_DIGIT:
    MOV AH,2     ;MOVE CURSOR TO A NEW LINE
    MOV AL,0DH
    INT 21H
    MOV DL,0AH
    INT 21H
    JMP @BEGIN   ;GO TO BEGINNING
INDEC ENDP

END MAIN
```

Output: