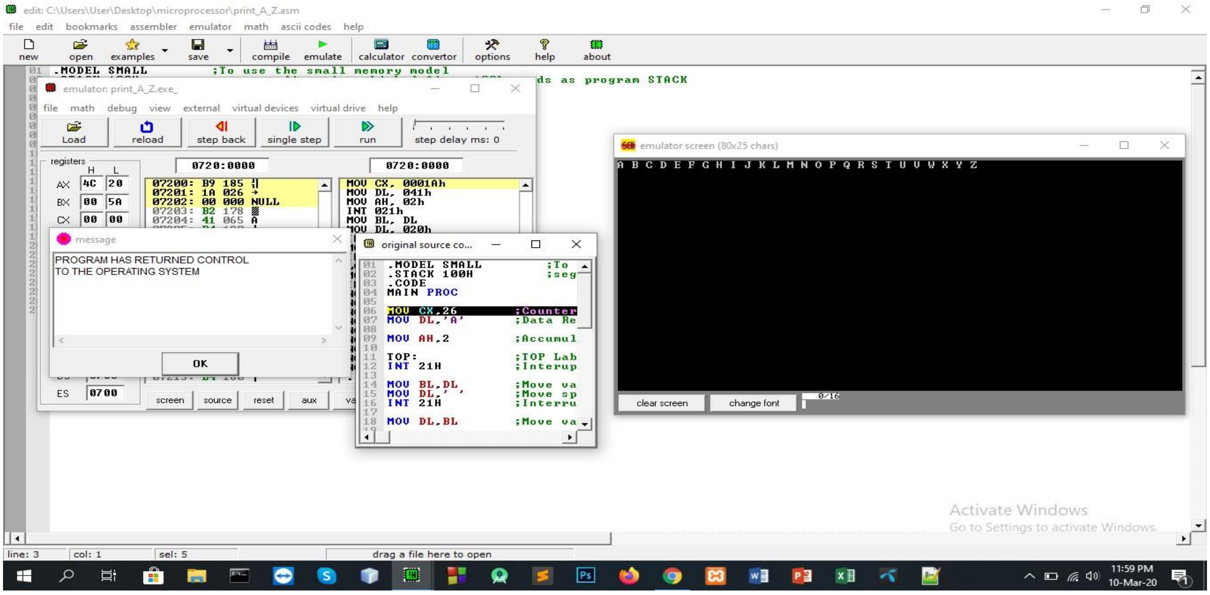
Q1. Write a program in assembly language to print alphabets from A-Z

Solution:

```
.MODEL SMALL       ;To use the small memory model
.STACK 100H        ;segment directive which defines 100h words as program STACK
.CODE
MAIN PROC

    MOV CX,26       ;Counter Register Assign 26
    MOV DL,'A'      ;Data Register Assign A

    MOV AH,2        ;Accumulator Register for Output

    TOP:            ;Label
    INT 21H         ;Interupt

    MOV BL,DL       ;Move value DL to BL
    MOV DL,' '      ;Move space into DL
    INT 21H         ;Interrupt

    MOV DL,BL       ;Move value BL into DL

    INC DL          ;Increment DL and Decrement CX
    LOOP TOP

MOV AH,4CH
INT 21H             ;INT 21h function 4Ch is preferred Action

MAIN ENDP           ;Denotes the end of(PROC) your procedure
END MAIN            ;Marks the end of the file
```

Output:



edit: C:\Users\User\Desktop\microprocessor\print_A_Z.asm

emulator: print_A_Z.exe_

```
07200: B9 185 ╣
07201: 1A 026 →
07202: 00 000 NULL
07203: B2 178 ▓
07204: 41 065 A
```

```
MOV CX, 0001Ah
MOV DL, 041h
MOV AH, 02h
INT 021h
MOV BL, DL
MOV DL, 020h
```

message

PROGRAM HAS RETURNED CONTROL
TO THE OPERATING SYSTEM

original source co...

```
.MODEL SMALL                ;To
.STACK 100H                 ;seg
.CODE
MAIN PROC

MOV CX,26                   ;Counter
MOV DL,'A'                  ;Data Re

MOV AH,2                    ;Accumul

TOP:                        ;TOP Lab
INT 21H                     ;Interup

MOV BL,DL                   ;Move va
MOV DL,' '                  ;Move sp
INT 21H                     ;Interru

MOV DL,BL                   ;Move va
```

emulator screen (80x25 chars)

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
file math debug view external virtual devices virtual drive help
Load reload step back single step run step delay ms: 0
0720:0000
0720:0000
registers
AX 4C 20
BX 00 5A
CX 00 00
ES 0700
OK
screen source reset aux
clear screen change font 0/16
line: 3 col: 1 sel: 5 drag a file here to open
Activate Windows
Go to Settings to activate Windows.
11:59 PM
10-Mar-20

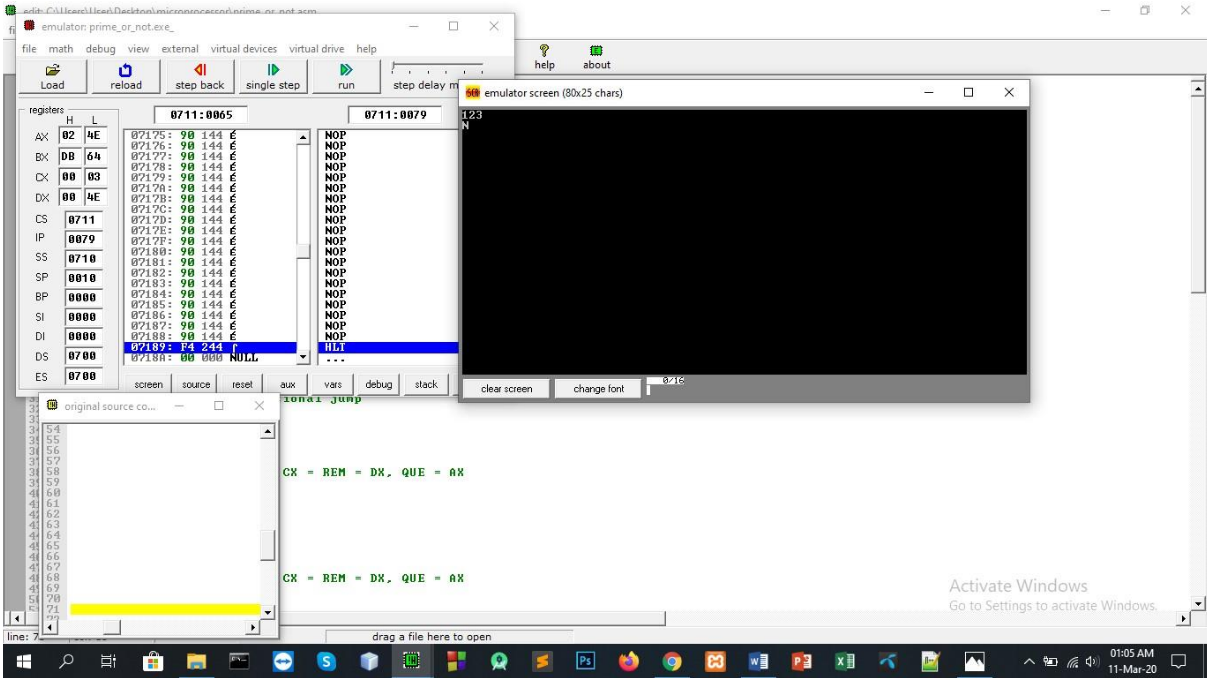
Q2. Write a program in assembly language to check whether the number inputted prime or not prime

Solution:

```
.MODEL SMALL
.STACK 10H
.CODE
MAIN PROC
   MOV BH,0
   MOV BL,100
   INPUT:
   MOV AH,1
   INT 21H
   CMP AL,13D
   JNE NUMBER
   JMP CHECK

   NUMBER:
   SUB AL,30H
   MOV CL,AL
   MOV AL,BH
   MUL BL
   ADD AL,CL
   MOV BH,AL
   JMP INPUT

   CHECK:
   CMP BH,1
   JLE NOT_PRIME   ;a conditional jump
   MOV CX,2
   AND AX,0
   AND DX,0
   MOV AL,BH
   DIV CX          ;DX:AX / CX = REM = DX, QUE = AX
   MOV CX,AX

   ISPRIME:
   CMP CX,2
   JL PRIME
   AND AX,0
   AND DX,0
   MOV AL,BH
   DIV CX          ;DX:AX / CX = REM = DX, QUE = AX
   DEC CX
   CMP DX,0
   JE NOT_PRIME
```

```
    JMP ISPRIME

    PRIME:
    MOV AH,2
    MOV DL,0AH
    INT 21H
    MOV DL,0DH
    INT 21H
    MOV DL,'P'
    INT 21H
    JMP EXIT

    NOT_PRIME:
    MOV AH,2
    MOV DL,0AH
    INT 21H
    MOV DL,0DH
    INT 21H
    MOV DL,'N'
    INT 21H

    EXIT:

MAIN ENDP
END MAIN
```

Output:



```
123
N
```

registers

	H	L
AX	02	4E
BX	DB	64
CX	00	03
DX	00	4E
CS	0711	
IP	0079	
SS	0710	
SP	0010	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

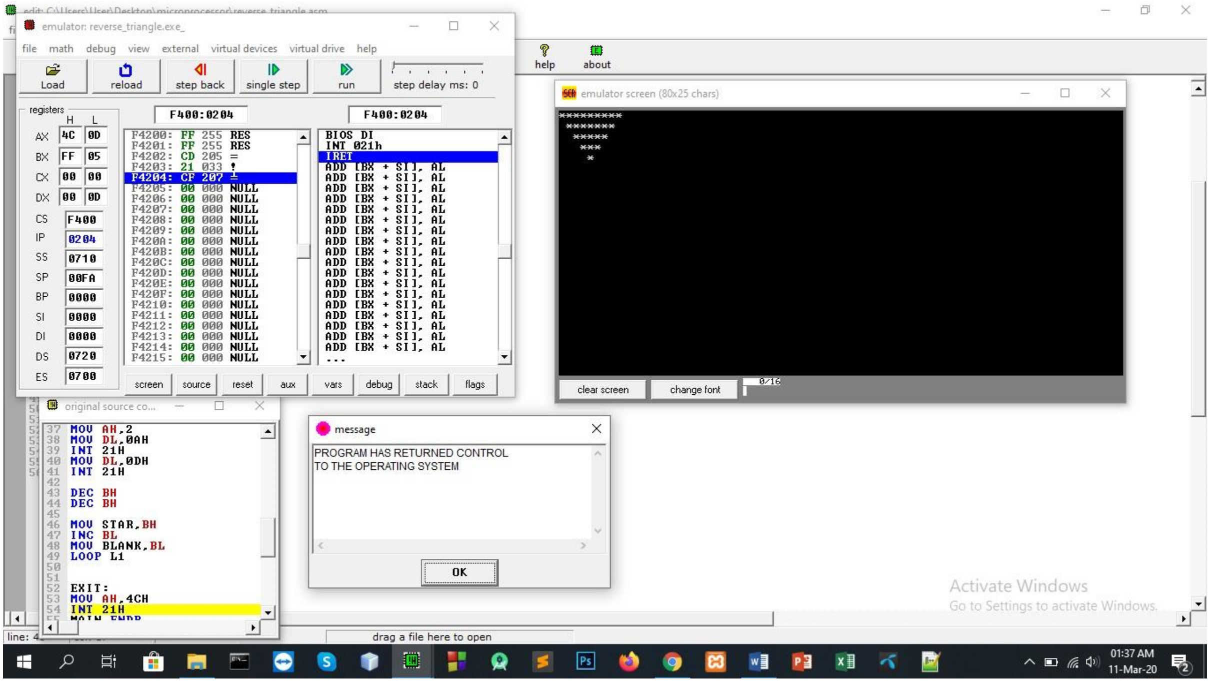
```
0711:0065                    0711:0079
07175: 90 144 É              NOP
07176: 90 144 É              NOP
07177: 90 144 É              NOP
07178: 90 144 É              NOP
07179: 90 144 É              NOP
0717A: 90 144 É              NOP
0717B: 90 144 É              NOP
0717C: 90 144 É              NOP
0717D: 90 144 É              NOP
0717E: 90 144 É              NOP
0717F: 90 144 É              NOP
07180: 90 144 É              NOP
07181: 90 144 É              NOP
07182: 90 144 É              NOP
07183: 90 144 É              NOP
07184: 90 144 É              NOP
07185: 90 144 É              NOP
07186: 90 144 É              NOP
07187: 90 144 É              NOP
07188: 90 144 É              NOP
07189: F4 244 ſ              HLT
0718A: 00 000 NULL           ...
```

```
ional jump

CX = REM = DX, QUE = AX

CX = REM = DX, QUE = AX
```

Q3. Write a program in assembly language to print a reverse triangle
Solution:

```
.MODEL SMALL
.STACK 100H
.DATA
STAR DB ?
BLANK DB ?
.CODE
MAIN PROC
   MOV AX,@DATA
   MOV DS,AX

   MOV CX,5
   MOV BH,9
   MOV BL,0

   MOV STAR,BH
   MOV BLANK,BL

   L1:
      CMP BLANK,0
      JE L2

      MOV AH,2
      MOV DL,32
      INT 21H
      DEC BLANK

      JMP L1

   L2:
      MOV AH,2
      MOV DL,'*'
      INT 21H
      DEC STAR
      CMP STAR,0
      JNE L2

   MOV AH,2
   MOV DL,0AH
   INT 21H
   MOV DL,0DH
   INT 21H

   DEC BH
   DEC BH

   MOV STAR,BH
   INC BL
   MOV BLANK,BL
   LOOP L1


   EXIT:
   MOV AH,4CH
   INT 21H
MAIN ENDP
END MAIN
```

Output:



```
    H  L
AX  4C 0D
BX  FF 05
CX  00 00
DX  00 0D
CS  F400
IP  0204
SS  0710
SP  00FA
BP  0000
SI  0000
DI  0000
DS  0720
ES  0700
```

```
F400:0204
F4200: FF 255 RES
F4201: FF 255 RES
F4202: CD 205 =
F4203: 21 033 !
F4204: CF 207 ±
F4205: 00 000 NULL
F4206: 00 000 NULL
F4207: 00 000 NULL
F4208: 00 000 NULL
F4209: 00 000 NULL
F420A: 00 000 NULL
F420B: 00 000 NULL
F420C: 00 000 NULL
F420D: 00 000 NULL
F420E: 00 000 NULL
F420F: 00 000 NULL
F4210: 00 000 NULL
F4211: 00 000 NULL
F4212: 00 000 NULL
F4213: 00 000 NULL
F4214: 00 000 NULL
F4215: 00 000 NULL
```

```
F400:0204
BIOS DI
INT 021h
IRET
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
...
```

```
*********
 *******
  *****
   ***
    *
```

```
MOV AH,2
MOV DL,0AH
INT 21H
MOV DL,0DH
INT 21H

DEC BH
DEC BH

MOV STAR,BH
INC BL
MOV BLANK,BL
LOOP L1


EXIT:
MOV AH,4CH
INT 21H
MAIN ENDP
```

message

PROGRAM HAS RETURNED CONTROL
TO THE OPERATING SYSTEM

OK

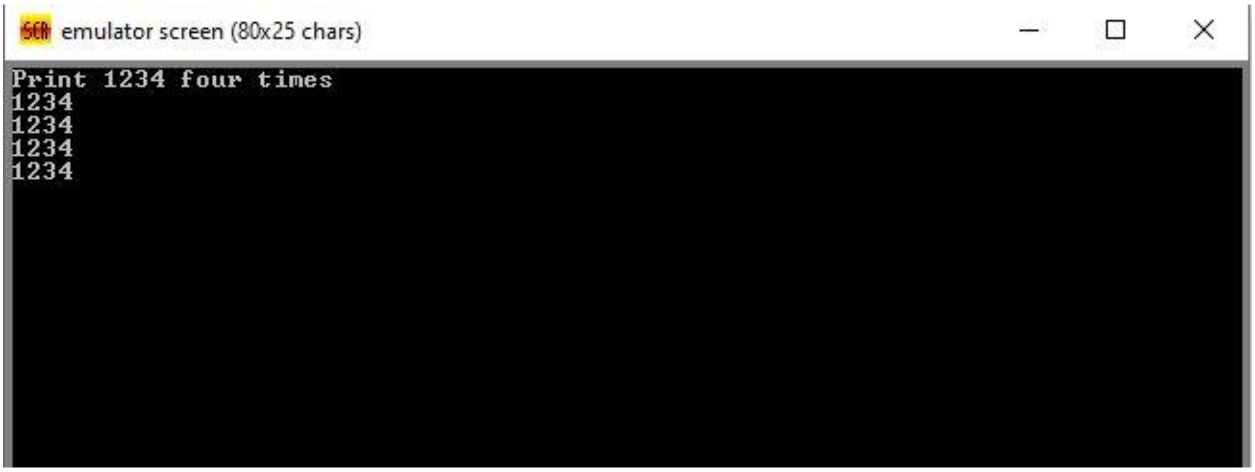
Q4. Write an assembly program same as following-

```
1234
1234
1234
1234
```

Solution:

```
 .MODEL SMALL
 .STACK 100H

 .DATA
    PRINT  DB  "Print 1234 four times $"
    D DB 10,13,"$"
    M DB 10,13," $"
.CODE
   MAIN PROC
     MOV AX, @DATA                          ; initialize DS
     MOV DS, AX

     LEA DX, PRINT                          ; load and print PRINT
     MOV AH, 9
     INT 21H
     MOV BL,4
     CHECK:
     LEA DX,D
     MOV AH,9
     INT 21H
                                            ; initialize CX
     MOV CX, 4
                                            ; set output function
     MOV AH, 2                              ; set DL with 0
     MOV DL, 49

     @LOOP:                                 ; loop label
          INT 21H                           ; print character

          INC DL                            ; increment DL to next ASCII
          DEC CX                            character
   JNZ @LOOP                                ; decrement CX
      DEC BL                                ; jump to label @LOOP if CX is 0
      JNZ CHECK

     MOV AH, 4CH
     INT 21H                                ; return control to DOS
   MAIN ENDP
  END MAIN
```

Output:



```
Print 1234 four times
1234
1234
1234
1234
```