Lab 6

Write a program to print the hexadecimal representation of the numbers included in your Roll number on screen

Mechanism to Conduct Lab:

Students and teacher communicate through Skype/Adobe Connect. Students will write code using Notepad or Programmer's Notepad and will share code and screen output.

Nasm: https://vulms.vu.edu.pk/Courses/CS401/Downloads/AssmSoft.zip

DosBOX: http://sourceforge.net/projects/dosbox/files/dosbox/0.74-2/DOSBox0.74-2-win32-installer.exe/download

Programmers Notepad: https://github.com/simonsteele/pn/releases/download/v2.4.2/portable-pn2421440.zip

Solution:

```
[0x0100]
jmp start
; subroutine to clear the screen
clrscr: push es
push ax
push di
mov ax, 0xb800
mov es, ax ; point es to video base
mov di, 0 ; point di to top left column
nextloc: mov word [es:di], 0x0720 ; clear next char on screen
add di, 2 ; move to next screen location
cmp di, 4000 ; has the whole screen cleared
jne nextloc ; if no clear next position
pop di
pop ax
pop es
```

```
ret
; subroutine to print a number at top left of screen
; takes the number to be printed as its parameter
printnum: push bp
mov bp, sp
push es
push ax
push bx
push cx
push dx
push di
mov ax, 0xb800
mov es, ax ; point es to video base
mov ax, [bp+4] ; load number in ax
mov bx, 16 ; use base 10 for division
mov cx, 0 ; initialize count of digits
nextdigit: mov dx, 0 ; zero upper half of dividend
div bx ; divide by 10
add dl, 0x30 ; convert digit into ascii value
push dx ; save ascii value on stack
inc cx ; increment count of values
cmp ax, 0 ; is the quotient zero
jnz nextdigit ; if no divide it again
mov di, 0 ; point di to top left column
nextpos: pop dx ; remove a digit from the stack
mov dh, 0x07 ; use normal attribute
mov [es:di], dx ; print char on screen
```

```
add di, 2 ; move to next screen location
loop nextpos ; repeat for all digits on stack
pop di
pop dx
pop cx
pop bx
pop ax
pop es
pop bp
ret 2
start: call clrscr ; call the clrscr subroutine
mov ax, 123456789
push ax ; place number on stack
call printnum ; call the printnum subroutine
mov ax, 0x4c00 ; terminate program
int 0x21
```