4. Develop an assembly language program to compute *nCr* using recursive procedure. Assume that 'n' and 'r' are non-negative integers.

```
.model small

initds macro
    mov ax,@data        ; initializing the data segment
    mov ds,ax           ; it is ds, not dx
endm

putchar macro char
    mov dl,char        ; load the printable character's hex value in dl
    mov ah,2            ; function number is 9
    int 21h             ; using dos interrupt 21h
endm

exit macro
    mov ah,4ch          ; to terminate
    int 21h
endm

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

.data
            n db 6              ; aim is to find -> 6c3
            r db 3
            answer db 0
 .code
            initds

            mov al,n
            mov bl,r

            call ncr            ; call ncr procedure

             mov al,answer      ; copy that answer to your al
            aam                 ; split al into al & ah
            add ax,3030h        ; convert into ascii
            mov bx,ax           ; take a copy to be safe
            putchar bh          ; display 1st digit
            putchar bl          ; display 2nd digit

            exit
```

```
ncr proc

            cmp bl,0                ; nC0 = 1
            jne go1
            add answer,1
            ret

      go1:  cmp bl,al               ; nCn = 1
            jne go2
            add answer,1
            ret

      go2:  cmp bl,1                ; nC1 = n
            jne go3
            add answer,al
            ret

      go3:  dec al                  ; nCn-1= n
            cmp bl,al
            jne go4
            inc al
            add answer,al
            ret

      go4:  push ax
            push bx       ; n-1
            call ncr        ; C
            pop bx          ;   r
            pop ax

            dec bx
            push ax       ; n-1
            push bx         ; C
            call ncr        ;   r-1
            pop bx
            pop ax
            ret
ncr endp
end
```