3) WAP (recursive func NCR)

```
.model small
.data
n dw 4
r dw 2
ncr dw 0

.code
mov ax,@data
mov ds, ax

mov ax,n
mov bx,r

call ncrpro
call disp
jmp final

ncrpro    proc near
          cmp ax,bx        ;r=n
          je  res1
          cmp bx,0         ;r=0
          je  res1
          cmp bx,1         ;r=1
          je  resn
          dec ax
          cmp bx,ax
```

```
        je incr
        push ax
        push bx
        call ncrpro

        pop bx
        pop ax
        dec bx
        push ax
        push bx
        call ncrpro
        pop bx
        pop ax
        ret

reo1:   inc ncr
  ret

incr:   inc ncr
rlen :  add ncr, ax        ; 1+2  3+3 = 6
  ret
ncrpro endp

disp    proc near
        mov bx, ncr
        add bx, 3030h
        mov dl, bh
        mov ah, 02h
        int 21h
        ret
       disp endp

final :    mov ah, 4ch
           int 21h
           end
```

2) WAP to binary count:

```
.Model small
.code
        MOV CL,00
       MOV AH,00H
       MOV AL,03H
        int 10H

BACK :    MOV BH,00H
          MOV DH ,00H
          MOV DL , 00H
          MOV AH , 02H
           INT 10H
        MOV AL,CL
        ADD AL,00H
        AAM                          ; 0000
        ADD AX,3050H                 ; 3030
        MOV CH,AL
        MOV DL,AH
        MOV AH,02H
        int 21H
        MOV DL,CH
        MOV AH,02H
        int 21H
        Call DELAY
        INC CL
        XOR AX,AX
        CMP CL,100D
         JNE BACH
         JE LAST

        DELAY   PROC  NEAR
         push  AX
         push  BX
         Push  CX
         MOV CX,00FFH

AG:       MOV BX,00FFH

AG1:        NOP
          DEC BX
          JNZ AG1
          DEC CX
          JNZ AG
           POP CX
           POP BX
           POP AX
            RET
```

```
DELAY ENDP

LAST:  MOV AH, 4Ch
       int 21H
       END
```

prog to place the cursor, (accept x & y coordinate)

```
• model small
disp macro msg
LEA DX,MSG
   MOV AH, 09H
   int 21H
ENDM

• Data
~~Row DB 0DH, 0AH, "enter x coordinate : $"~~
Row DB 02 DUP (0)
col DB 02 DUP (0)
```

```
msg1   DB  ODH , OAH ," enter x coodinate:$"
msg2   DB  ODH , DAH ,"enter y coordinate:$",
.code  msg3 DB ODH , OAH ,"cursor placed at correct position$"
        MOV AX,@data
        MOV DS,AX
        DISP MSG1
        MOV SI ,OFFSET ROW
         CALL READ
          DISP MSG2
        MOV SI, DFFSET COL
        CALL READ
        MOV SI,OFFSET ROW
        MOU AH,[SI]
        INC SI
        MOU AL , [SI]
        SUB AX , 3030H
        AAD
        MOV DH, AL
        MOV SI,OFFSET COL
        MOV AH [SI]
         INC SI
         MOV AL , [SI]
        SUB AX, 3030H
         AAD
        MOV DL, AL

        MOV AH,00
        MOV AL,03H
        INT 10H
        MOV AH,OOH
        int 10H
        disp msg3
        JMP FINAL
```

```
READ PROC NEAR
      MOV CX,02H

BACK : MOV AH,01H
       int 21H
       MOV [SI], AL
      INC SI
      DEC CX
     JNZ BACK
     RET
 READ ENDP

 FINAL :MOV AH, 01H
          INT 21H
          MOV AH, 4CH
          int 21H
          END
```