

**Name: MD. Ezazul Haque Mahi**

**ID:22301221**

**Section:01**

**Final Assignment**

; You may customize this and other start-up templates;

; The location of this template is c:\emu8086\inc\0\_com\_template.txt

org 100h

.MODEL SMALL

**factorial macro X**

**mov cx,X**

**Cmp cx,0**

**Je o!**

**mov ax,1**

**mov bx,1**

**loop1:**

**mul bx**

**inc bx**

**loop loop1**

**;you can see the output in Dx register**

**mov dx,ax**

**o!:**

**mov dx,1 ;0! is one**

**endm**

**nCr macro n,r**

```
mov bx,n
mov ax,r
sub bx,ax
mov nr,bx
;getting n!
mov ax,n
mov temp,ax
call fact
mov n,ax
```

```
;getting r!
mov ax,r
mov temp,ax
call fact
mov r,ax
```

```
;getting n-r!
mov ax,nr
mov temp,ax
call fact
mov nr,ax
;getting r!(n-r!)
mov bx,nr
mov ax,r
mul bx
```

```
;getting n/r!(n-r!)
mov bx,ax
mov ax,n
div bx
mov cx,ax ;ans is in cx register
```

```
endm
```

**expression macro arr1,lenght**

```
mov di,0
mov cx,lenght
mov si,0
```

**;loop to find closing bracket if it finds a closing bracket then it will perform the operation and when its done again it will try to find another closing bracket**

```
loop_arr:
inc di
cmp di,lenght
je exit2
mov ax,arr1[si]
cmp ax,")"
je perform
push ax
inc si
loop loop_arr
```

**perform: ;it will check which operator**

```
add sp,1
pop bx
pop dx
add sp,1
cmp dx,"+"
je addition
cmp dx,"-"
je subs
cmp dx,"/"
je division
cmp dx,"*"
je mult
```

**addition:**

```
pop ax
add sp,1
add ax,bx
pop dx ;pop the starting bracket
add sp,1
```

```
push ax ;push the ans
inc si
jmp loop_arr
```

```
subs:
add sp,1
pop ax
sub ax,bx
pop dx
push ax
inc si
jmp loop_arr
```

```
division:
add sp,1
pop ax
```

```
div bl
add sp,1
pop dx
push ax
inc si
jmp loop_arr
```

```
mult:
add sp,1
pop ax
mov al,ah
mov ah,0
```

```
mul bl
add sp,1
pop dx
push ax
inc si
jmp loop_arr
```

```
exit2:
pop dx ;the ans is in dx
```

```
endm
.STACK 100H
```

```
.DATA
str db "For calculating Factorial type 1$"
str1 db "For calculating nCr type 2$"
str2 db "For calculating an expression type 3 and put the expression
in arr1$"
str3 db "choose your n and r$"
arr dw "(",12,"+", "(","(",67,"-",30,")","/",3,")","*",4,")"
l dw 18
N dw ?
R dw ?
temp dw ?
nr dw ?
; declare variables here
```

```
.CODE
MAIN PROC
```

```
; initialize DS
```

```
MOV AX,@DATA
```

MOV DS,AX

; enter your code here  
;printing the messages

lea dx,str  
mov ah,9  
int 21h

mov dl,0Ah  
mov ah,2  
int 21h  
mov dl,0Dh  
mov ah,2  
int 21h

lea dx,str1  
mov ah,9  
int 21h

mov dl,0Ah  
mov ah,2  
int 21h  
mov dl,0Dh  
mov ah,2  
int 21h

lea dx,str2  
mov ah,9  
int 21h

**;taking input**

mov dl,0Ah  
mov ah,2  
int 21h

```
mov dl,0Dh
mov ah,2
int 21h
```

**;checking**

```
mov ah,1
int 21h
sub al,30h
```

```
cmp al,1
je factorial1
```

```
cmp al,2
je nCr_block
```

```
cmp al,3
je expression_block
```

**; calling the factorial macro**

**factorial1:**

```
mov dl,0Ah
mov ah,2
int 21h
mov dl,0Dh
mov ah,2
int 21h
```

```
mov ah,1
int 21h
sub al,30h
mov ah,0
factorial ax
jmp exit
```

**;calling the nCr macro**

```
nCr_block:
mov dl,0Ah
mov ah,2
int 21h
mov dl,0Dh
mov ah,2
int 21h
lea dx,str3
mov ah,9
int 21h
```

```
mov ah,1
int 21h
sub al,30h
mov ah,0
mov N,ax
```

```
mov ah,1
int 21h
sub al,30h
mov ah,0
mov R,ax
nCr N,R
jmp exit
```

### **expression\_block:**

```
;put the value in arr1
;put your lenght in l variable
expression arr,l ;call the expression macro
jmp exit
```

```
exit:
;exit to DOS
```

```
MOV AX,4C00H
```



INT 21H

MAIN ENDP

**fact proc**

**mov cx,temp**

**mov ax,1**

**mov bx,1**

**loop2:**

**mul bx**

**inc bx**

**loop loop2**

**ret**

**fact endp**

**END MAIN**

...

**ret**