Lab 9 Spring 2019 CECS 524 Due End of lab today

1. Cut and paste the code below into your favorite 16 bit assembler IDE (assuming GUI Turbo). Save it.
2. Download the file PCMAC.INC from BB and put it the same directory as the code from step one.
3. Assemble, link and run the program. Show it to me.
4. That’s it.

.model small

INCLUDE pcmac.inc

.stack 100h

.data

m1 db 'Factorial of $'

m2 db ' is $'

i dw 4

M32768 db '-32768$'

.code

main proc

mov ax, @data

mov ds, ax

push i

call factorial

mov bx, ax

\_putstr m1

mov ax, i

call putdec

\_putstr m2

mov ax, bx

call putdec

\_putch 13, 10

\_exit

main endp

factorial proc

push bp

mov bp, sp

mov ax, word ptr [bp+4]

cmp ax, 1

jle donef

sub ax, 1

push ax

call factorial

donef:

imul word ptr [bp+4]

pop bp

ret 2

factorial endp

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

; PutDec writes a 16 bit integer value as chars to the display.;

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

PutDec PROC

push ax

push bx

push cx

push dx

cmp ax, -32768 ; -32768 is a special case as there

jne TryNeg ; is no representation of +32768

\_PutStr M32768

jmp Done

TryNeg:

cmp ax, 0 ; If number is negative ...

jge NotNeg

mov bx, ax ; save from it from \_PutCh

neg bx ; make it positive and...

\_PutCh '-' ; display a '-' character

mov ax, bx ; To prepare for PushDigs

NotNeg:

mov cx, 0 ; Initialize digit count

mov bx, 10 ; Base of displayed number

PushDigs:

sub dx, dx ; Convert ax to unsigned double-word

div bx

add dl, '0' ; Compute the Ascii digit...

push dx ; ...push it (can push words only)...

inc cx ; ...and count it

cmp ax, 0 ; Don't display leading zeroes

jne PushDigs

;

PopDigs: ; Loop to display the digits

pop dx ; (in reverse of the order computed)

\_PutCh dl

loop PopDigs

Done:

pop dx ; Restore registers

pop cx

pop bx

pop ax

ret

PutDec ENDP

end main