Assembly

- No class next Thursday

- findChar(&str, bChar); int & (returns address)

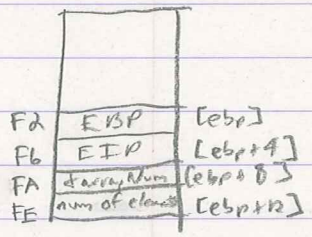
```
char byte 'c'
strMsg byte "John went to town", 0
charaddress dword ?
;charaddress = findChar(&strMsg, char
mov ax, 0
mov al, char
push ax
push offset strMsg
call findChar
add esp, 6
mov char, eax          ; then check for -1

findChar proc Near32
    push ebp           ; preserving base ptr
    mov ebp,esp        ; set new stack ptr
    push ebx; push ecx;   preserve registers
    mov ebx, [ebp+8]   ; ebx points to strMsg
    mov eax, -1        ; assume no char found
    mov cx, [ebp+12]   ; cl = character we are looking for
findCharLoop:
    cmp byte ptr [ebx], 0   ; is this end of string
    je finish          ; leave loop
    cmp [ebx], cl      ; is this the character
    je finish          ; character found
    inc ebx            ; points to next char
    jmp findCharLoop
```

```
found:
    mov eax, ebx        ; eax points to character

finished:
    pop ecx             ; restore register
    pop ebx             ; restore register
    pop ebp             ; restore registers
    ret                 ; return
ENDP

    arrayNum word 50 dup(?)
    dSum  dword ?
    ; dSum = sum (arrayNum, wSize)
    mov ax, 50          ; 50 values to add
    push ax
    push offset arrayNum
    call sum
    mov dSum, eax
    add esp, 6
```




```
Sum proc Near32
    push ebp
    push ebp, esp
    push ebx; push ecx; push edx;
    xor eax, eax        ; total = 0
    mov ebx, [ebp+8]    ; ebx → element in the array
    mov ecx, 0          ; make sure all bytes of ecx are 0
    mov cx, [ebp+12]    ; number of elements
stloop:
    xor edx, edx        ; clear edx
    movsx edx, word ptr [ebx]   ; 2 byte value to 4 byte value
    add eax, edx                ; add
    add ebx, 2                  ; increment pointer
```

```
    loop sLoop

    pop edx
    pop ecx
    pop ebx
    pop ebp
    ret
sum ENDP
```

- putchar (dword) : void
  → display a char on the screen

- getchar ( ) : dword
  → get a character from keyboard (doesn't display)

```
call getchar
mov char, al
```

- getcharecho ( ) : dword
  → same as getchar() and displays the char

```
call getchar
mov char, al
```