

Assembly
Exam Review

12.8.11

page 1

- jmp instruction

first byte is opcode, second byte is distance

it adds the instruction + 2 bytes + distance

3. - ~~call~~ call statement updates EIP & pushes EIP, then replaces EIP the address of the function

9.
eax = FFCE3D02
ebx = 00000E45
esp = 0402AE28

41 = 2AE28

1) B2 2C FF 41

2) subtract 4 from SP - 0402AE24

6. code struct
key dd ?
RecNum dd ?
next dd ?
ends

```

5. int f(n)
{
    if (n == 0)
        return 1;
    else if (n == 1)
        return 2;
    else
        t = f(n-1);
        v = f(n-2);
        w = n * t + (n-1) * v;
}

```

f PROTO stdcall, M: dword

f PROC stdcall uses ebx edx

local t: dword

v: dword

w: dword

```

mov eax, 1 ; assume N=0
cmp dword ptr N, 0 ; are we done
je done ; yes, outta here
mov eax, 2 ; is N==1
cmp dword ptr N, 1 ;
je done ; yes, outta here
mov ebx, N ; need to decrement N
dec N ; N--
INVOKE f, ebx ; find f(n-1)
mov t, eax ; t = f(n-1)
dec ebx ; ebx = n-2
INVOKE f, ebx ; f(n-2)
mov v, eax ; f(n-2)
; eax = f(n-2)

```

EBP [ebp]

EIP [eip]

N [ebp+8]

```
inc ebx      ; ebx = N-1
imul ebx     ; multiply ebx & eax /  $eax = (n-1) \times V$ 
mov v, eax   ; saves eax /  $v = (n-1) \times V$ 
mov eax, N
imul t       ;  $eax = n \times t$ 
add eax, V   ; done
done:
ret
endp
```