# HOMEWORK #9

1. True
2. Pushad
3. False, use the stack

|  |  |
| --- | --- |
| var | offset |
| Varx db ‘hello’ | 0 |
| Var3 db 4 | 5 |
| Msg2 dd 1 | 5+1=6 |
| Mp dw –10 | 6+4=10 |

1. mov ax, 10  
   add ax, [k]  
   add ax, [q]  
   sub ax, [w]  
   mov [p],ax
2. (×) 32bit  
   (×) native 80386  
   (×) native pentium  
   (×)paging  
   (×)segmentation  
   (×)memory protection
3. Mov ecx, 5  
   input:  
    GetInt AX  
    push ax  
   loop input ;*get all inputs*  
   call MyAvg  
   .EXIT  
     
   MyAvg:  
    mov ecx, 5  
    mov ax, 0  
    sum1:  
    pop bx  
    add ax, bx  
    loop sum1  
    mov bx, 5  
    cwd  
    idiv bx  
    PutInt ax  
    ret

|  |  |
| --- | --- |
| flags | instr |
| CF – carry flag | Cli, sti, jc, jnc |
| OF - overflow | Jo, jno |
| PF - parity | Jp, jnp |
| ZF - zero | Je, jne, jz, jnz |
| IF - interrupt | Sti, cli |

1. 1. Ax = 360, OF =1 … overflowed AL
   2. Ax = 90, OF = 0
2. 100times, because bx+ax never becomes 0, so only ecx is used for the loop counter.
3. GetInt AX  
   mov BX, 0  
   mov ECX, 16  
   countB:  
    shl AX, 1  
    jnc got0  
    inc bx ;got 1  
   got0:  
    loop countB  
   PutInt BX ; numbr of bits

|  |  |
| --- | --- |
| AX + 23 | That makes |
| -23 | ZF=1 |
| -24 | SF=1 |
| 0 | SF=0 |
| 300 | CF=1 |
| 105 | OF=1 |
| 104 | OF=0 |
| 5 | ZF=0 |