1) Breakpoint 1:

SS:SP → 100h

2) Breakpoint 2:

SS:SP

100h
0FFh 98h
0FEh 76h
0FDh 7Eh
0FCh 54h
0FBh ? (CH)
0FAh ? (CL)

3) Breakpoint 4:

SS:SP

100h
0FFh 98h
0FEh 76h
0FDh 7Eh
0FCh 54h
0FBh ? (CH)
0Fah ? (CL)
0F9h IP (h)
0F8h IP (l)
0F7h
0F6h

4) Breakpoint 3:

SS:SP

100h
0FFh 98h
0FEh 76h
0FDh 7Eh
0FCh 54h
0FBh 16h
0Fah 0CAh
0F9h IP (h)
0F8h IP (l)
0F7h ? (BP high 8 bits)
0F6h ? (BP low 8 bits)
5) Breakpoint 5:

\[ SS:SP \rightarrow 00h \]

.data
con1 equ 0ffh
var2 db 0
var3 dw 1234h

___F___ move ax, bx ; incorrect instruction “move”, should be “mov”
___F___ mov ax, cl ; Sizes do not match
___F___ add con1, ah ; destination cannot be an immediate number
___T___ sub ax, con1 ; correct
___F___ push var2 ; push operates on 16-bit word only
___F___ mov ax, [cx] ; cx cannot be used for indirect memory addressing.
___F___ xor ax, var2 ; sizes does not match.
___F___ sub var2, byte ptr var3 ; No memory-to-memory addressing.
___T___ mov ds, ax ; correct, you are doing this in each of your program!
___F___ cmp con1, var2 ; destination cannot be a constant number
proc:     nop
        not    ax                  ;NOT each bit of AX
        inc    ax                  ;ax<-ax+1
        mov    cx, 16               ; set the loop counter, 16 bits

reverse: rcl    ax,1            ;rotate through carry to left, move the MSB bit
              ;into CF
        rcr    bx,1                ;rotate through carry to right, move CF into the
              ;MSB of BX
        loop   reverse             ;keep doing this for 16 bits
        mov    ax,bx
        ret