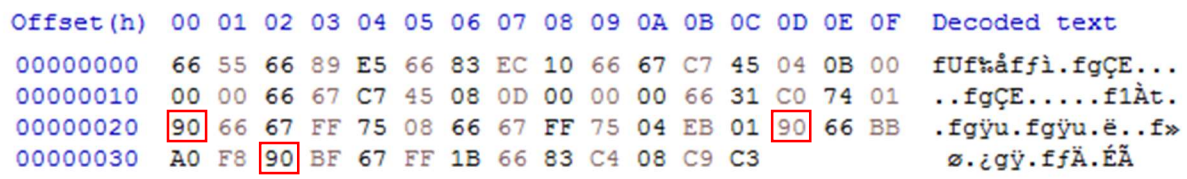
Assignment 7
Gunnar Yonker

HxD:



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	66	55	66	89	E5	66	83	EC	10	66	67	C7	45	04	0B	00	fUf‰åfƒì.fgÇE...
00000010	00	00	66	67	C7	45	08	0D	00	00	00	66	31	C0	74	01	..fgÇE.....f1Àt.
00000020	90	66	67	FF	75	08	66	67	FF	75	04	EB	01	90	66	BB	.fgÿu.fgÿu.ë..f»
00000030	A0	F8	90	BF	67	FF	1B	66	83	C4	08	C9	C3				ø.¿gÿ.fƒÄ.ÉÃ

Disassembly using ndisasm:

```
C:\Users\sysadmin\Desktop\Assignment 7>ndisasm -b 32 -o 0x0 sample7
00000000  6655              push bp
00000002  6689E5            mov bp,sp
00000005  6683EC10          sub sp,byte +0x10
00000009  6667C745040B00    mov word [di+0x4],0xb
00000010  0000              add [eax],al
00000012  6667C745080D00    mov word [di+0x8],0xd
00000019  0000              add [eax],al
0000001B  6631C0            xor ax,ax
0000001E  7401              jz 0x21
00000020  90                nop
00000021  6667FF7508        push word [di+0x8]
00000026  6667FF7504        push word [di+0x4]
0000002B  EB01              jmp short 0x2e
0000002D  90                nop
0000002E  66BBA0F8          mov bx,0xf8a0
00000032  90                nop
00000033  BF67FF1B66        mov edi,0x661bff67
00000038  83C408            add esp,byte +0x8
0000003B  C9                leave
0000003C  C3                ret
```

Notes:

66 55: push ebp

66 89 E5: mov ebp, esp

66 83 EC 10: sub esp, 0x10 - sub because second operand is an immediate value, subtract 16 from esp

66 67 C7 45 04 0B 00 00 00 - mov dword [ebp+0x4], 0xb

66 67 C7 45 08 0D 00 00 00 - mov dword [ebp+0x8], 0xd

66 31 C0 - xor eax, eax - seems like a valid way to clear the eax register

74 01 - jmp

FF: potentially a rogue byte, 66 67 are prefix bytes for the next FF

66 67 FF 75 08 - push word [ebp+0x8]

66 67 FF 75 04 - push word [ebp+0x4]

Assignment 7
Gunnar Yonker

EB 01: jmp, valid displacement

A0: nop potententially a rogue byte, not needed before 66 BB

66 BB A0 F8 - mov bx, 0xf8a0

FF: nop

BF FF BF 67 FF 1B 66 - mov edi, 0x661bff67

83 C4 08 - add esp, 0x8, add based on second operand

C9 - leave

C3 – ret