

5.

1. **segment .data**
2. **sayi db 1**
3. **segment .text**
4. **global \_start**
5. **\_start:**
6. **mov ecx , sayi**
7. **inc ecx**
8. **mov [sayi] , ecx**
9. **mov eax , 4**
10. **mov ebx , 1**
11. **mov ecx , sayi**
12. **mov edx , 4**
13. **int 0x80**
14. **mov eax , 1**
15. **mov ebx , 0**
16. **int 0x80**

6.

1. The following LDIR instruction copies from HL to DE and the number of bytes from BC

**LD HL, &0000**

**LD DE, &C000**

**LD BC, &4000**

**LDIR**

**RET**

1. The following instructions are very similar but move the starting address at 0xA000 to 0xB000 and store it in BC

**LD HL, &0xa000**

**LD DE, &0xb000**

**LD BC, &0xc000**

**LDIR**

**RET**

1. LDIR was introduced to bridge the gap between assembly and high-level languages, because it provided a way to directly load data from the memory into the registers. It also compatible with RISC design principles and makes coding efficient and easy to read.

Sources:

<https://landley.net/history/mirror/cpm/z80.html>

https://en.wikipedia.org/wiki/Zilog\_Z80