

ÖDEV 4: ASSEMBLER DERLEYİCİSİ

Programlama Dilleri (derleyiciler) giriş olarak yazılan bir programın kaynak kodunu alır ve ondan sonra kaynak kod program ne iş yapıyorsa, kendisi de aynı işi yapan programı yazar. Bu düşünce ile 8-bitlik bir **El-Ceziri** işlemcisinin olduğu varsayalım. Bu işlemcinin RAM 8-bitlik 256 byte (hafıza) bulunmaktadır. Aynı zamanda üzerinde AX, BX, CX, DX şeklin dört tane 8-bitlik kayıtçı (register) bulunmaktadır. Bu işlemcinin komut kümesi aşağıdaki listede verilmiştir.

TOP OP1,OP2
CRP OP1,OP2
BOL OP1,OP2
CIK OP1,OP2
DS ETIKET
DSD ETIKET
DAL ETIKET
DK ETIKET
DB ETIKET
ATM OP1,OP2
VE OP1,OP2
VEY OP1,OP2

El-Ceziri işlemcisinin Assembleri yazılmış bir program bir **metin (text)** dosyasında olacaktır. Hazırlanacak program, bu dosya giriş olarak alacak ve bu Assembler ile yazılan program (dosyada olan program) ne iş yapıyorsa, hazırlanan program da aynı işi yapacaktır. Program bitiminde bütün registerların içeriğini ekrana yazacaktır ve eğer bellek kullanıldıysa, sadece belleğin o elemanı veya elemanları ekrana yazılacaktır.

Kurallar:

- 1) Programın ismi assembler.c veya assembler.cpp şeklinde olacaktır.
- 2) Indentation kuralına uyulacaktır.
- 3) assembler.c programı girdi dosyasının ismini klavyeden alacaktır.
- 4) Programınız Ax,BX,CX ve DX içeriklerini ekrana yazacaktır.
- 5) Ödevleriniz 16.12.2022 tarihinde saat 23:59'a kadar mail ile gönderilmesi gerekmektedir.
- 6) Kaynak kod Arş.Gör.Hüseyin Enes OKUTAN'ın belirlediği yere gönderilecektir.

İpucu: TOP komutu için hazırlanan C kodu aşağıdaki gibidir.

```
if (Komutlar[i]=='T'){
    if (Komutlar[i+4]=='A'){
        if (Komutlar[i+7]=='A'){
            AX=AX+AX;
            i=i+10;
        }
        elseif (Komutlar[i+7]=='B'){
            AX=AX+BX;
            i=i+10;
        }
    }
}
```

```

    }
    elseif (Komutlar[i+7]=='C'){
        AX=AX+CX;
        i=i+10;
    }
    elseif (Komutlar[i+7]=='D'){
        AX=AX+DX;
        i=i+10;
    }
    elseif (Komutlar[i+7]=='I'){
        if (Komutlar[i+9]=='I'){
            AX=AX+RAM[(int)Komutlar[i+8]-48];
            i=i+10;
        }
        elseif (Komutlar[i+10]=='A'){
            AX=AX+RAM[((int)Komutlar[i+8]-48)*10+((int)Komutlar[i+9]-48)];
            i=i+11;
        }
        else {
            AX=AX+RAM[((int)Komutlar[i+8]-48)*100+((int)Komutlar[i+9]-
48)*10+((int)Komutlar[i+10]-48)];
            i=i+12;
        }
    }
    else {
        if ((int)Komutlar[i+8]==13){
            AX=AX+(int)Komutlar[i+7]-48;
            i=i+9;
        }
        elseif ((int)Komutlar[i+9]==13){
            AX=AX+((int)Komutlar[i+7]-48)*10+((int)Komutlar[i+8]-48);
            i=i+10;
        }
        else {
            AX=AX+((int)Komutlar[i+7]-48)*100+((int)Komutlar[i+8]-
48)*10+((int)Komutlar[i+9]-48);
            i=i+11;
        }
    }
}
}

```