

## ÖDEV 4 ve ÖDEV 5

Ödev 4 ve Ödev 5 kapsamında Python ile bir makine dili emülatörü yazmanız beklenmektedir. Yazacağınız kod, ders kitabının 581 ve 582. sayfasında (Ek C) yer alan ve aşağıda kolaylık açısından tekrar sunulan makine dili ile yazılan herhangi bir kodu çalıştıracaktır.

1. Makine mimarisi bilgileri (yazmaç uzunluğu, hücre sayısı, vb.) için ders kitabı sayfa 581'e bakınız.
2. Başlangıçta yazmaç ve bellek değerlerinin resetlenmiş (sıfırlanmış) olduğunu kabul ediniz.
3. Başlangıçta program sayacı 00'dadır.
4. Çalıştırılacak olan kod bulunulan klasörde yer alan *input.txt* isimli dosyadan okutulacak ve çalıştırılmadan önce belleğe 00 adresinden başlamak kaydıyla kopyalandığı kabul edilecektir. Input dosyasında her komut ayrı satırda yer almaktadır.

Örnek (input.txt dosyası):

**21 0B**

**14 04**

**C0 00**

5. Yazmış olduğunuz emülator programı çalıştıktan sonra ekrana tüm yazmaç değerlerini aşağıdaki gibi yazmalı ve sonlanmalıdır.

**R0: 00**

**R1: 0B**

**R2: 00**

**R3: 00**

**R4: C0**

**R5: 00**

**R6: 00**

**R7: 00**

**R8: 00**

**R9: 00**

**R10: 00**

**R11: 00**

**R12: 00**

**R13: 00**

**R14: 00**

**R15: 00**

6. Emülatörünüzün JUMP komutu haricinde çalışan kısmı Ödev 4, JUMP komutunu gerçekleştirdiğiniz kısmı Ödev 5 olarak değerlendirilecektir. Fakat sizin yükleyeceğiniz ödev .py uzantılı sadece bir python dosyası içermelidir. Dosyayı verilen link üzerinden Google Drive'a yükleyin. İsimlendirmede Türkçe karakter kullanmayın.
7. Son gönderim tarihi: 26 Kasım 2015 23:59 (uzatma yapılmayacaktır).
8. Puanlama aşağıdaki şekilde yapılacaktır:

İşlevsellik: 7

Modülerlik: 1,5 (Her bir komutun ayrı ayrı fonksiyonlarda gerçekleştirilmesi, vb.)

Okunabilirlik: 1,5 (Program içi açıklamalar, değişken isimleri, vb)

Op-

| code | Operand | Description   |
|------|---------|---|
| 1    | RXY     | LOAD the register R with the bit pattern found in the memory cell whose address is XY.<br><i>Example:</i> 14A3 would cause the contents of the memory cell located at address A3 to be placed in register 4.  |
| 2    | RXY     | LOAD the register R with the bit pattern XY.<br><i>Example:</i> 20A3 would cause the value A3 to be placed in register 0.   |
| 3    | RXY     | STORE the bit pattern found in register R in the memory cell whose address is XY.<br><i>Example:</i> 35B1 would cause the contents of register 5 to be placed in the memory cell whose address is B1.   |
| 4    | 0RS     | MOVE the bit pattern found in register R to register S.<br><i>Example:</i> 40A4 would cause the contents of register A to be copied into register 4.  |
| 5    | RST     | ADD the bit patterns in registers S and T as though they were two's complement representations and leave the result in register R.<br><i>Example:</i> 5726 would cause the binary values in registers 2 and 6 to be added and the sum placed in register 7.   |
| 6    | RST     | ADD the bit patterns in registers S and T as though they represented values in floating-point notation and leave the floating-point result in register R.<br><i>Example:</i> 634E would cause the values in registers 4 and E to be added as floating-point values and the result to be placed in register 3.   |
| 7    | RST     | OR the bit patterns in registers S and T and place the result in register R.<br><i>Example:</i> 7CB4 would cause the result of ORing the contents of registers B and 4 to be placed in register C.  |
| 8    | RST     | AND the bit patterns in register S and T and place the result in register R.<br><i>Example:</i> 8045 would cause the result of ANDing the contents of registers 4 and 5 to be placed in register 0.   |
| 9    | RST     | EXCLUSIVE OR the bit patterns in registers S and T and place the result in register R.<br><i>Example:</i> 95F3 would cause the result of EXCLUSIVE ORing the contents of registers F and 3 to be placed in register 5.  |
| A    | R0X     | ROTATE the bit pattern in register R one bit to the right X times. Each time place the bit that started at the low-order end at the high-order end.<br><i>Example:</i> A403 would cause the contents of register 4 to be rotated 3 bits to the right in a circular fashion.   |
| B    | RXY     | JUMP to the instruction located in the memory cell at address XY if the bit pattern in register R is equal to the bit pattern in register number 0. Otherwise, continue with the normal sequence of execution. (The jump is implemented by copying XY into the program counter during the execute phase.)<br><i>Example:</i> B43C would first compare the contents of register 4 with the contents of register 0. If the two were equal, the pattern 3C would be placed in the program counter so that the next instruction executed would be the one located at that memory address. Otherwise, nothing would be done and program execution would continue in its normal sequence. |
| C    | 000     | HALT execution.<br><i>Example:</i> C000 would cause program execution to stop.  |