**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**2019-2020 Öğretim Yılı Bahar Yarıyılı, Bilgisayar Organizasyonu Dersi**

**Ev Ödevi #2**

Öğretim Elemanı Unvanı / Adı Soyadı: Prof.Dr. İsmail KADAYIF Veriliş Tarihi: 20/04/2020

Öğrenci Adı Soyadı: Öğrenci No: Teslim Tarihi: 30/04/2020

**1) Aşağıdaki soruları cevaplayınız.**

**a)** Word tipinde tanımlanmış -88 sayısının 0x42A0B284 adresinden itibaren belleğe little endian olarak yerleştirilmesinin nasıl olacağını gösteriniz. **(10 puan)**

|  |  |
| --- | --- |
| **Adres** | **Değer** |
| 0x42A0B284 |  |
| 0x42A0B285 |  |
| 0x42A0B286 |  |
| 0x42A0B287 |  |

**b)** Word tipinde tanımlanmış -88 sayısının 0x42A0B284 adresinden itibaren belleğe big endian olarak yerleştirilmesinin nasıl olacağını gösteriniz. **(10 puan)**

|  |  |
| --- | --- |
| **Adres** | **Değer** |
| 0x42A0B284 |  |
| 0x42A0B285 |  |
| 0x42A0B286 |  |
| 0x42A0B287 |  |

**2)**

|  |  |
| --- | --- |
| dizi: | .space 512 |

Tanımlaması veriliyor. Data segmentin başlangıç adresinin 0x10010000 olduğunu kabul edelim. Buna göre,

**a)** dizi değişkeninin tek boyutlu bir tamsayıları tutan bir dizi olduğunu düşünürsek, dizi[14] elemanının adresini onaltılık düzende hesaplayınız**. (10 puan)**

**b)** dizi değişkeninin iki boyutlu 4x32’lik bir tamsayı dizisi olduğunu ve belleğe row-major olarak yerleştirildiğini kabul edersek, dizi[2][26] elemanının adresini hesaplayınız. **(10 puan)**

**c)** dizi değişkeninin iki boyutlu 4x32’lik bir tamsayı dizisi olduğunu ve belleğe column-major olarak yerleştirildiğini kabul edersek, dizi[2][26] elemanının adresini hesaplayınız. **(10 puan)**

**3)** Aşağıdaki tanımlama verilmektedir.

|  |  |
| --- | --- |
| dizi: | .space 512 |

Buna göre, dizi değişkeninin iki boyutlu 8x16’lik bir tamsayı dizisi olduğunu ve belleğe column-major olarak yerleştirildiğini kabul edersek, **dizi[5][14] = 27** işlemini ait kod parçasını MAL dilinde yazınız. **(25 puan)**

4) Derste gördüğümüz ve kodu aşağıda verilen **buble sort** sıralama algoritmasını düşünelim.

|  |
| --- |
| .data  ar: .word -1, -10, 21, 40, 41, 60, -11, -20, 211, 120 .word 14, 70, 91, 409, 21, 40, 81, 90, 901, 110  .word 1, 10, 11, 10, 11, 10, 111, 110, 11, 1230  .word 13, 120, 321, -40, -61, 550, 411, -340, 13, 50  .word 14, 50, 41, -560, 300, -200, 1, 0, 1, 40    # $8 -- a flag, 1 if the algorithm is done  # $9 -- an offset to the correct element of the array  # $10 -- address of the element to compare  # $11 -- the array element for comparison  # $12 -- the neighbor of the array element for comparison  # $14 -- base address of array ar    .text  .globl main main: la $14, ar loop: li $8, 1 # flag = true  li $9, 0  **for: add $10, $14, $9 lw $11, ($10) # load element**  **lw $12, 4($10) # load next element**  **sub $13, $11, $12**  **blez $13, noswap # if they are in order, don’t swap**  **li $8, 0**  **sw $11, 4($10) # swap elements sw $12, ($10)**  **noswap: add $9, $9, 4**  **sub $13, $9, 196 # see if end of the array reached bltz $13,for**  beq $8, $0, loop # loop until done li $v0, 10 syscall |

İçteki döngüye bakarsak (koyulaştırılmış kod kısmı) dış döngünün kaçıncı iterasyonda olduğuna bakmaksızın her defasında dizinin son elemanına kadar kontrol yapılmaktadır. Oysa buna gerek yoktur. Örneğin, dış döngünün 3. iterasyonu düşünüldüğünde dizinin sondan iki

**MÜHENDİSLİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**2019-2020 Öğretim Yılı Bahar Yarıyılı, Bilgisayar Organizasyonu Dersi**

**Ev Ödevi #2**

Öğretim Elemanı Unvanı / Adı Soyadı: Prof.Dr. İsmail KADAYIF Veriliş Tarihi: 20/04/2020

Öğrenci Adı Soyadı: Öğrenci No: Teslim Tarihi: 30/04/2020

elemanı zaten doğru pozisyonunda bulundukları için kontrolün dizinin sondan üçüncü elemanına kadar yapılması performansı artıracaktır. Buna göre, yukarıdaki kodu tekrardan yazarak bu kodu daha performanslı hale dönüştürünüz. **(25 puan)**

**Not:** Eklemiş olduğunuz kod satırlarını farklı renkte (örneğin mavi) yazınız. Böylece eklediğiniz satırları daha kolay takip edebileceğim.

|  |
| --- |
|  |