

BİLGİSAYAR MÜHENDİSLİĞİ  
BLM2067 ARASINAV  
120 Dakika

1. (30 p) Node isimli düğüm yapısını kullanan aşağıdaki fonksiyonların ne yaptığını kısaca anlatınız.

(Explain briefly what the following functions that use the node structure below do.)

```
struct node {
    int data;
    struct node* next;
};
```

|     |                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (a) | <pre>void fonksiyon1(struct node** a, struct node** b) {     struct node* cur;     if (*a == NULL) {         *a = *b;     }     else {         cur = *a;         while (cur -&gt;next != NULL) {             cur = cur -&gt;next;         }         cur -&gt;next = *b;     }     *b=NULL; }</pre>                                                                                                               |
| (b) | <pre>static void fonksiyon2(struct node** head) {     struct node* r = NULL;     struct node* cur = *head;     struct node* n;     while (cur!= NULL) {         n = cur -&gt;next;         cur -&gt;next = r;         r = cur;         cur = n;     }     *head = r; }</pre>                                                                                                                                     |
| (c) | <pre>void fonksiyon3(struct node* head) {     struct node* cur = head;     if (cur == NULL) return;     while(cur -&gt;next!=NULL) {         if (cur -&gt;data == cur -&gt;next-&gt;data) {             struct node* nextNext = cur -&gt;next-&gt;next;             free(cur -&gt;next);             cur -&gt;next = nextNext;         }         else {             cur = cur -&gt;next;         }     } }</pre> |

2. a) (20 p) Aşağıdaki infix ifadesini yığıt (stack) kullanarak postfix ifadesine dönüştürün.  
(Convert the following infix expression to postfix one using stacks.)

$$80 + (((8 - 6) * 3) - 30 / 5) * 4$$

- b) (15 p) Yığıtları kullanarak aşağıdaki postfix ifadesinin sonucunu hesaplayın.  
(Evaluate (find the result) the following postfix expression using stacks.)

$$50 \ 2 \ 6 \ 18 \ 12 \ - \ / \ 35 \ + \ - \ +$$

3. (35 p) Aşağıda verilen yapı için bir düğüm yapısı tanımlayınız.  
(Define a node structure for the structure given below.)

```
struct node{  
    ...  
};
```

Aşağıdaki yapıyı oluşturacak C kodunu yazınız. Düğümler arasındaki bağlantıları gerçekleştirirken, değer atamalarını ve bellek kontrollerini de yapmalısınız. Bütün programı yazmanıza gerek yok. 1. Elemanın değerini 1, 2. Elemanın değerini 2, n. elemanın değerini n olarak atayabilirsiniz. Programınızda n değerinin kullanıcı tarafından girilmiş olduğunu varsayın.

(Write a C code that will create the following structure. While making connections between nodes, you must also perform value assignments and memory checks. You don't need to write the whole program. You can assign, 1 to the value of the 1st element, 2 to the value of the 2nd element, n to the value if the nth element. Assume that the value n is user-entered in your program.)

