

ANKARA ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ
BLM2067 ARASINAV
120 Dakika



ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ



SINAVLAR VE ÖDEVLER İÇİN ŞEREF SÖZÜ

Bir Ankara Üniversitesi öğrencisi olarak;

- -Bu ödevde/sınavda yardım almadığımı ya da hiç kimseye yardım etmediğimi,
- -Başkasına ait olan bir çalışmayı kendi çalışmam olarak sunmadığımı,
- -Sınav/ödev sorularının çözümü için hiç kimseden (öğrenci, öğretim üyesi ya da arkadaş) yardım istemediğimi,
- -Problemin çözümünü bulmak için interneti ya da çevrimiçi ya da basılı herhangi bir belgeyi kullanmadığımı beyan ederim.

Yukarıdaki ifadelere uymadığının tespit edilmesi durumunda sınavdan/ödevden sıfır alacağımı ve hakkımda **Ankara Üniversitesi Öğrenci Disiplin Yönetmeliği** çerçevesinde soruşturma açılacağını biliyorum.

HONOR CODE FOR EXAMS and ASSIGMENTS

As an Ankara University student, I agree that;

- I have neither given nor received unauthorized assistance on this exam or assignment.
- I have not represented the work of another as my work.
- I have not asked someone else (student, teacher, and friends) to help with this assignment or exam questions.
- I have not used the internet or any online or printed document to find problem solutions

I understand that failure to comply with the statements above will result in receiving a zero from this exam/assignment and being reported for academic dishonesty by the **disciplinary policies of Ankara University**.

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 ->next != NULL) { cur = cur ->next; } cur ->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 ->next; cur ->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 ->next!=NULL) { if (cur ->data == cur ->next->data) { struct node* nextNext = cur ->next->next; free(cur ->next); cur ->next = nextNext; } else { cur = cur ->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.)

