

ÖDEV-1

Öğrenci No: 233301076 Ad-Soyad : Mahmut Esat Kolay

NÖ ☒ İÖ ☐

Bir bağlı listede aynı değerli olan düğümleri (nodes) gruplayan bir fonksiyon yazınız. Yazacağınız kod listedeki tüm yinelenen değerleri ardışık olarak yan yana olacak şekilde yeniden düzenlemelidir. Mesela listede dört kez "3" sayısının tekrar ettiğini varsayalım. Bu durumda ilk karşılaşılan "3" sayısının hemen yanına tekrar eden diğer "3" sayıları gelmelidir. Böylece liste ilk durumuyla aynı sırada olacaktır.

Yazacağınız fonksiyon 2 parametrelidir: Bağlı listenin başını temsil eden bir *Node işaretçisine ait referans* ve bir araya getirilecek maksimum sayı için *max_tekrar*. Aynı değere sahip diğer değerler ise silinmelidir.

Bağlı listenin elemanları öğrenci numaranızdan oluşacaktır. Örneğin **213301130** öğrenci numarasına sahip bir öğrencinin bağlı listedeki elemanları şu şekilde olmalıdır:

{ 2, 1, 3, 3, 0, 1, 1, 3, 0 }

Yazdığımız gruplama fonksiyonu, örneğin: *grupla(head, 10)* çağırıldıktan sonra şu şekilde olmalıdır:

{ 2, 1, 1, 1, 3, 3, 3, 0, 0 }

Yukarıda *max_tekrar* parametresine verdiğimiz 10 değeri fazla olduğu için hiçbir elemanı silinmedi. Eğer fonksiyonu *grupla(head, 2)* şeklinde çağırıyorduk son durum şu şekilde olacaktı:

{ 2, 1, 1, 3, 3, 0, 0 }



- Bağlı listenin elemanlarının öğrenci numaralarınızdan oluşması gerekmektedir.
- Düğümün data alanlarını değiştirmeden düğümü taşıyarak yapmalısınız.
- Yeni bir düğüm eklemeyiniz.
- Listeden bir düğüm sildiğiniz zaman hafızayı boşaltınız.

SORU:

1. Düğüm elemanlarını kullanıcıdan alarak bağlı listeye ekleyiniz.
 2. Yukarıdaki talimatları gerçekleştiren **grupla** fonksiyonunu yazınız.
 3. Bağlı listenin durumunu **grupla** fonksiyonu öncesi ve sonrasını gösterecek şekilde **yazdır** fonksiyonu ile yazdırınız.
 4. Yazmış olduğunuz **grupla** fonksiyonunun çalışma zamanı analizini (Big-O) **aşamalı ve detaylı** bir şekilde yapınız.
- Sadece C/C++ dilinde kodlayınız.
 - Kodları öğrencino_odev1.c şeklinde yükleyiniz. Kod çıktısı ve Çalışma zamanı analizini word dosyasında yapıp pdf formatında yükleyiniz.
 - Toplamda sisteme iki dosya yüklemelisiniz.

ÖDEV-1

Öğrenci No: 233301076 Ad-Soyad : Mahmut Esat Kolay

NÖ ☒ İÖ ☐

KOD ÇIKTISI (EKRAN GÖRÜNTÜSÜ):

```
Öğrenci numaranızı girin: 233301076
ilk hali

2 3 3 3 0 1 0 7 6

Gruplama sonrası
2 3 0 1 7 6

kodu xcode da yazdim 22-24-29-41-49-55-70-72-77 breaking point noktaları
Program ended with exit code: 0
```

ÇALIŞMA ZAMANI ANALİZİ:

```
createNode fonksiyonu:
Zaman Karmaşıklığı: O(1)
Alan Karmaşıklığı: O(1)
Node* createNode(int data) {
    Node* newNode = (Node*)malloc(sizeof(Node)); -->1
    newNode->data = data; -->1
    newNode->next = NULL; -->1
    return newNode; -->1
}

grup fonksiyonu:
Zaman Karmaşıklığı: O(n^2). döngü içleri n kadar döner ++ gittiği için
Alan Karmaşıklığı: O(1)
void grup(Node* head) {
    Node* current = head; -->1
    while (current != NULL) { -->n.
        Node* runner = current; -->1
        while (runner->next != NULL) { -->n.
            if (runner->next->data == current->data) {
                Node* temp = runner->next;
                runner->next = runner->next->next;
                free(temp);
            } else {
                runner = runner->next;
            }
        }
        current = current->next; -->1
    }
}
```

```
add fonksiyonu:
Zaman Karmaşıklığı: O(n) - tek döngü var n olur doğrusal olduğu için
Alan Karmaşıklığı: O(1)
void add(Node** head, int data) {
    Node* newNode = createNode(data); -->1
    if (*head == NULL) { -->1
        *head = newNode; -->1
        return; -->1
    }
    Node* temp = *head; -->1
    while (temp->next != NULL) { -->n
        temp = temp->next;
    }
    temp->next = newNode; -->1
}

printList fonksiyonu:
Zaman Karmaşıklığı: O(n)
Alan Karmaşıklığı: O(1)
void printList(Node* head) {
    Node* temp = head; -->1
    while (temp != NULL) { -->n
        printf("%d ", temp->data);
        temp = temp->next;
    }
    printf("\n"); -->1
}

Öğrenci numarasını bağlı listeye dönüştürme işlemi
Zaman Karmaşıklığı: O(n) - doğrusal döngü var
Alan Karmaşıklığı: O(n) - Girilen öğrenci numarası karakter sayısına göre
bellek kullanımı artar
for (int i = 0; ogr_no[i] != '\0'; i++) { for için n kadar döner
    int sayac = ogr_no[i] - '0';
    add(&head, sayac);
}
```