

**2024-2025 BAHAR DÖNEMİ**  
**ALGORİTMALAR VE PROGRAMLAMA II**  
**14. HAFTA**  
**GENEL TEKRAR ve UYGULAMA 10**

## 1. Soru: Özyinelemeli Fonksiyon ile Çarpma İşlemi

Kullanıcıdan iki adet negatif olmayan tamsayı alarak, bu iki sayının çarpımını **özyineleme-rekürsif (recursive)** yöntemi kullanarak hesaplayan bir C programı yazınız. Bu işlemi sadece **toplama işlemi (+)** kullanılarak yapılmalısınız. Aşağıdaki bazı yönergeler sunulmuştur. Çözümünüzün bu kısıtlamalara ve isteklere uyduğundan emin olunuz:

- Kullanıcıdan iki sayı alınız.
- Çarpım sonucunu ekranda gösteriniz
- \*, /, % gibi operatörleri kullanmadan çözmeye çalışınız.
- Fonksiyonun bir durma (base case) ve tekrar (recursive call) durumu olmalıdır
- Yalnızca toplama işlemi (+) ile işlem yapılmalıdır
- Girdi değerleri 0 ile 100 arasında olmalıdır

Yukarıdaki kısıtlamalara ve isteklere göre kullanıcıdan alınan değerleri gözeterek çarpım işleminin sonucu hesaplayan ve ekrana yazdıran C programını yazınız.

## 2. Soru: Bit Manipülasyonu ile Tek Bir Biti Kontrol Etme

Kullanıcıdan bir pozitif tamsayı ve bir bit konumu (0–31 arasında) alın. Bu program kullanıcının seçtiği bitin **0 mı yoksa 1 mi** olduğunu bit manipülasyonu kullanarak belirlemelidir. Aşağıdaki işlemler yapılmalıdır:

- Kullanıcıdan bir pozitif tamsayı girmesini isteyin ve bu sayıyı kontrol edin.
- Kullanıcıdan kontrol edilmek istenen bit pozisyonunu alın (0 ile 31 arası olduğundan emin olun)
- Bit manipülasyonu (>> ve & operatörleri) kullanarak istenen bitin değerini tespit edin.
- Bitin 0 mı yoksa 1 mi olduğunu kullanıcıya ekrana bir mesaj ile bildirin.

### 3. Soru: Araç Kiralama Takip Sistemi Uygulaması

Bir araç kiralama şirketinin filosunda farklı türde araçlar bulunmaktadır. Her araç aşağıdaki bilgileri taşımaktadır:

- Araç numarası (tam sayı)
- Marka (en fazla 20 karakterlik karakter dizisi)
- Model (en fazla 20 karakterlik karakter dizisi)
- Araç türü (Sedan, Hatchback, SUV, Kamyon)
- Müsaitlik durumu (1: müsait, 0: kirada)
- Günlük kiralama ücreti (ondalıklı sayı)

Bu bilgiler ışığında aşağıdaki adımları takip ederek bir C programı yazınız:

- enum kullanarak araç türlerini tanımlayınız.  
SEDAN, HB, SUV ve KAMYON değerlerini içeren bir AracTuru adlı enum tipi tanımlayınız.
- typedef ile birlikte struct tanımlayınız.  
Yukarıdaki bilgileri kapsayan bir struct yapısı oluşturunuz ve bu yapıya Arac adını veriniz.
- Enum değerini metin olarak döndüren bir fonksiyon yazınız.  
AracTuru türündeki bir enum değerini alıp, onu anlamlı bir karakter dizisine çeviren bir fonksiyon (aracTurulsmiGetir) yazınız.
- Bir aracı ekrana yazdıran bir fonksiyon yazınız.  
Bir Arac değişkenini parametre olarak alarak, araç bilgilerini düzenli bir biçimde ekrana yazan aracBilgisiYazdir adlı bir fonksiyon yazınız.
- Ana programda 4 araçlık bir dizi oluşturunuz.
- En az bir tanesi kirada (müsait değil) olacak şekilde 4 araçlık bir Arac dizisi oluşturunuz ve her aracı aracBilgisiYazdir fonksiyonu ile ekrana yazdırınız.

### 4. Soru: Öğrenci Bilgilerini Dosyaya Yazma ve Okuma (Sıralı Erişimli)

Aşağıdaki işlemleri yapan bir C programı yazınız:

1. Kullanıcıdan 2 öğrenci için aşağıdaki bilgileri alınız:
  - Öğrenci numarası (tam sayı)
  - Öğrencinin adı (karakter dizisi)
2. Alınan bu bilgileri "ogrenci.txt" adlı bir metin dosyasına satır satır yazınız.  
Her satırda bir öğrencinin bilgisi olmalıdır.

Örnek satır biçimi:  
1234 Ali

3. Dosyaya yazma işlemi bittikten sonra, dosyayı tekrar açıp tüm öğrenci bilgilerini dosyadan okuyup ekrana yazdırınız.

**Not: Dosya işlemlerinde hata kontrolü yapılması zorunludur.**

## 5. Soru: Rasgele Erişimli Dosyada Ürün Fiyatı Güncelleme

Aşağıda tanımı verilen Product yapısı kullanılarak, ikili dosya işlemleriyle ürün fiyatı ekleme ve güncelleme işlemlerini gerçekleştiren bir C programı yazınız:

```
typedef struct {  
    int id;      // Ürün numarası  
    float price; // Ürün fiyatı  
} Product;
```

Programın gerçekleştirmesi beklenen işlemler aşağıdaki gibidir:

1. Kullanıcıdan alınan 3 adet ürün bilgisi (ürün numarası ve fiyatı), "**products.dat**" adlı bir **ikili (binary)** dosyaya yazılmalıdır. Bu işlemde **fwrite** fonksiyonu kullanılmalıdır.
2. Daha sonra kullanıcıdan bir ürün numarası alınarak, ilgili ürün dosya üzerinden **fread** ile aranmalıdır.
3. Ürün bulunduğu takdirde, kullanıcıdan yeni fiyat bilgisi alınmalı ve bu fiyat, **fseek** fonksiyonu kullanılarak doğrudan dosya üzerinde **rasgele erişim** yöntemiyle güncellenmelidir.
4. Programın sonunda, dosyada yer alan tüm ürün bilgileri okunarak ekrana yazdırılmalıdır.

## 6. Soru: Tek Yönlü Bağlı Liste ile Kitap Fiyatı Güncelleme

Aşağıda tanımlanan Book yapısını kullanarak, kitap bilgilerini **tek yönlü bağlı liste** üzerinde yöneten bir C programı yazınız:

```
typedef struct Book {  
    int id;      // Kitap no  
    char title[50]; // kitap adı  
    float price; // fiyatı  
    struct Book *next;  
} Book;
```

Program aşağıdaki işlemleri gerçekleştirmelidir:

1. Kullanıcıdan alınan kitap bilgilerini (**id**, **title**, **price**) listeye sırayla eklemelidir.
2. Kullanıcıdan bir kitap ID'si alarak ilgili kitabın fiyatını güncellemelidir.
3. Tüm kitap bilgilerini ekrana yazdırmalıdır.

# CEVAPLAR

## 1. Sorunun çözümü:

```
#include <stdio.h>

// Recursive function to multiply two numbers using addition
int multiply(int a, int b) {
    if (b == 0)
        return 0;
    return a + multiply(a, b - 1);
}

int main() {
    int num1, num2;

    // Take user input
    printf("Enter first number: ");
    scanf("%d", &num1);

    printf("Enter second number: ");
    scanf("%d", &num2);

    // Call recursive function
    int result = multiply(num1, num2);

    // Print the result
    printf("Product of %d and %d is %d\n", num1, num2, result);

    return 0;
}
```

## 2. Sorunun çözümü:

```
#include <stdio.h>

int main() {
    unsigned int sayi;
    int pozisyon;

    printf("Bir pozitif tamsayı giriniz: ");
    scanf("%u", &sayi);

    printf("Kontrol etmek istediğiniz bit pozisyonunu girin (0-31): ");
    scanf("%d", &pozisyon);
}
```

```

    if (pozisyon < 0 || pozisyon > 31) {
        printf("Geçersiz bit pozisyonu! 0 ile 31 arasında olmalıdır.\n");
        return 1;
    }

    int bitDegeri = (sayi >> pozisyon) & 1;

    printf("Bit %d konumundaki değer: %d\n", pozisyon, bitDegeri);

    return 0;
}

```

### 3. Sorunun çözümü:

```

#include <stdio.h>
#include <string.h>

// Enum: Araç türleri
typedef enum {
    SEDAN,
    HB,
    SUV,
    KAMYON
} AracTuru;

// Enum'dan string'e dönüşüm için
const char* aracTuruIsmiGetir(AracTuru tur) {
    switch (tur) {
        case SEDAN: return "Sedan";
        case HB: return "Hatchback";
        case SUV: return "SUV";
        case KAMYON: return "Kamyon";
        default: return "Bilinmiyor";
    }
}

// Struct: Araç bilgisi
typedef struct {
    int aracNo;
    char marka[20];
    char model[20];
    AracTuru tur;
    int musaitMi;
    float gunlukUcret;
} Arac;

// Fonksiyon: Araç bilgilerini yazdırır
void aracBilgisiYazdir(Arac a) {
    printf("Araç No: %d\n", a.aracNo);
    printf("Marka: %s\n", a.marka);
    printf("Model: %s\n", a.model);
}

```

```

        printf("Tür: %s\n", aracTuruIsmiGetir(a.tur));
        printf("Günlük Ücret: %.2f TL\n", a.gunlukUcret);
        printf("Durum: %s\n", a.musaitMi ? "Müsait" : "Kirada");
        printf("-----\n");
    }

// Araç sayısı
#define ARAC_SAYISI 4

int main() {
    // Araç dizisi tanımlama
    Arac aracler[ARAC_SAYISI] = {
        {1, "Toyota", "Corolla", SEDAN, 1, 750.0},
        {2, "Ford", "Focus", HB, 1, 680.0},
        {3, "Nissan", "X-Trail", SUV, 0, 920.0},
        {4, "Isuzu", "NPR", KAMYON, 1, 1200.0}
    };

    printf("=== Araç Listesi ===\n");
    for (int i = 0; i < ARAC_SAYISI; i++) {
        aracBilgisiYazdir(araclar[i]);
    }

    return 0;
}

```

#### 4. Sorunun çözümü:

```

#include <stdio.h>
#include <stdlib.h>

int main() {
    FILE *dosya;
    int i;
    int numara;
    char ad[20];

    // Yazma işlemi: Dosyayı "w" modunda aç
    dosya = fopen("ogrenci.txt", "w");
    if (dosya == NULL) {
        printf("Dosya açılamadı!\n");
        return 1;
    }

    // 2 öğrenci bilgisi al ve dosyaya yaz
    for (i = 0; i < 2; i++) {
        printf("%d. öğrencinin numarasını girin: ", i + 1);
        scanf("%d", &numara);
        printf("%d. öğrencinin adını girin: ", i + 1);
        scanf("%s", ad);

        fprintf(dosya, "%d %s\n", numara, ad);
    }
}

```

```

}

fclose(dosya); // Dosyayı kapat

// Okuma işlemi: Dosyayı "r" modunda aç
dosya = fopen("ogrenci.txt", "r");
if (dosya == NULL) {
    printf("Dosya okunamıyor!\n");
    return 1;
}

printf("\n--- Dosyadaki Öğrenci Bilgileri ---\n");

// Dosyadan oku ve ekrana yaz
while (fscanf(dosya, "%d %s", &numara, ad) == 2) {
    printf("Numara: %d, Ad: %s\n", numara, ad);
}

fclose(dosya); // Dosyayı kapat

return 0;
}

```

## 5. Sorunun Çözümü:

```

#include <stdio.h>
#include <stdlib.h>

typedef struct {
    int id;
    float price;
} Product;

int main() {
    FILE *fp;
    Product pr;

    // 1. Ürün bilgilerini dosyaya yaz
    fp = fopen("products.dat", "w");
    if (fp == NULL) {
        perror("Dosya açılmadı");
        return 1;
    }

    for (int i = 0; i < 3; i++) {
        printf("%d. Ürün ID ve fiyatı: ", i + 1);
        scanf("%d%f", &pr.id, &pr.price);
        fwrite(&pr, sizeof(Product), 1, fp);
    }
    fclose(fp);
}

```

```

// 2. Güncellenecek ürün ID'sini al
int targetId;
printf("Güncellenecek ürün ID: ");
scanf("%d", &targetId);

// 3. Fiyatı güncelle
fp = fopen("products.dat", "rb+");
if (fp == NULL) {
    perror("Dosya açılmadı");
    return 1;
}

while (fread(&pr, sizeof(Product), 1, fp) == 1) {
    if (pr.id == targetId) {
        printf("Yeni fiyat: ");
        scanf("%f", &pr.price);
        fseek(fp, -sizeof(Product), SEEK_CUR);
        fwrite(&pr, sizeof(Product), 1, fp);
        break;
    }
}
fclose(fp);

// 4. Tüm ürünleri ekrana yazdır
fp = fopen("products.dat", "rb");
if (fp == NULL) {
    perror("Dosya açılmadı");
    return 1;
}

printf("\nTüm ürünler:\n");
while (fread(&pr, sizeof(Product), 1, fp) == 1) {
    printf("ID: %d, Fiyat: %.2f\n", pr.id, pr.price);
}
fclose(fp);

return 0;
}

```

## 6. Sorunun Çözümü:

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct Book {
    int id;
    char title[50];
    float price;
    struct Book *next;
} Book;

```



```

// Yeni düğüm oluştur
Book* createBook(int id, const char* title, float price) {
    Book *newBook = (Book*)malloc(sizeof(Book));
    newBook->id = id;
    strcpy(newBook->title, title);
    newBook->price = price;
    newBook->next = NULL;
    return newBook;
}

// Listenin sonuna kitap ekle
void append(Book **head, int id, const char* title, float price) {
    Book *newBook = createBook(id, title, price);
    if (*head == NULL) {
        *head = newBook;
    } else {
        Book *temp = *head;
        while (temp->next != NULL)
            temp = temp->next;
        temp->next = newBook;
    }
}

// Kitap fiyatını güncelle
void updatePrice(Book *head, int targetId) {
    while (head != NULL) {
        if (head->id == targetId) {
            printf("Yeni fiyat: ");
            scanf("%f", &head->price);
            return;
        }
        head = head->next;
    }
    printf("Kitap bulunamadı.\n");
}

// Kitap listesini yazdır
void printBooks(Book *head) {
    printf("\nKitap Listesi:\n");
    while (head != NULL) {
        printf("ID: %d | Ad: %s | Fiyat: %.2f\n", head->id, head->title, head->price);
        head = head->next;
    }
}

int main() {
    Book *head = NULL;
    int n, id, targetId;
    char title[50];
    float price;

    printf("Kaç kitap eklenecek? ");
    scanf("%d", &n);
}

```

```
    for (int i = 0; i < n; i++) {
        printf("\n%d. kitap ID: ", i + 1);
        scanf("%d", &id);
        printf("Kitap adı: ");
        scanf(" %[^\\n]", title);
        printf("Kitap fiyatı: ");
        scanf("%f", &price);
        append(&head, id, title, price);
    }

    printf("\nGüncellenecek kitap ID: ");
    scanf("%d", &targetId);
    updatePrice(head, targetId);

    printBooks(head);

    return 0;
}
```