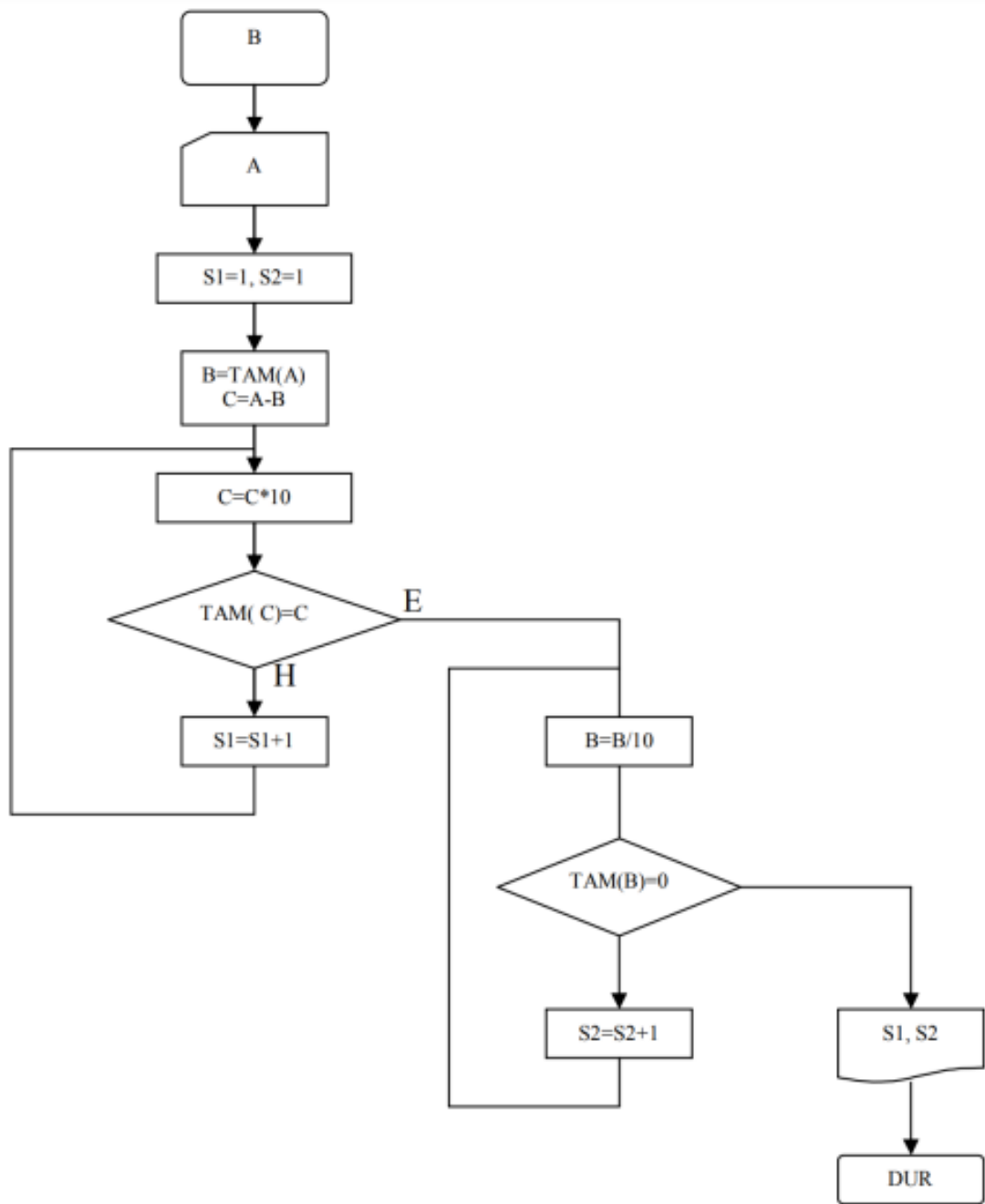


1. Rasgele girilen bir rasyonel sayısının ondalıklı kısmının ve tam kısmının hane sayısını bulan algoritmanın akış şemasını çizin ve C kodunu yazın.

Kullanıcıdan bir rasyonel sayı girişi alın.
Rasyonel sayının tam kısmını ve ondalık kısmını hesaplayın.
Tam kısmın hane sayısını bulmak için bir while döngüsü kullanarak her basamağı sayın.
Ondalık kısmın hane sayısını bulmak için bir başka while döngüsü kullanarak ondalık kısmı genişletin.
Hesaplanan hane sayılarını ekrana yazdırın.



```
#include <stdio.h>

int main() {
    double rasyonel_sayi;
    int tam_kisim_hane_sayisi = 0;
    int ondalik_kisim_hane_sayisi = 0;

    printf("Bir rasyonel sayı girin: ");
    scanf("%lf", &rasyonel_sayi);

    // Rasyonel sayının tam kısmını ve ondalık kısmını hesapla
    int tam_kisim = (int)rasyonel_sayi; // Tam kısmı al
    double ondalik_kisim = rasyonel_sayi - tam_kisim; // Ondalık kısmı al

    // Tam kısmın hane sayısını bul
    while (tam_kisim != 0) {
        tam_kisim /= 10;
        tam_kisim_hane_sayisi++;
    }

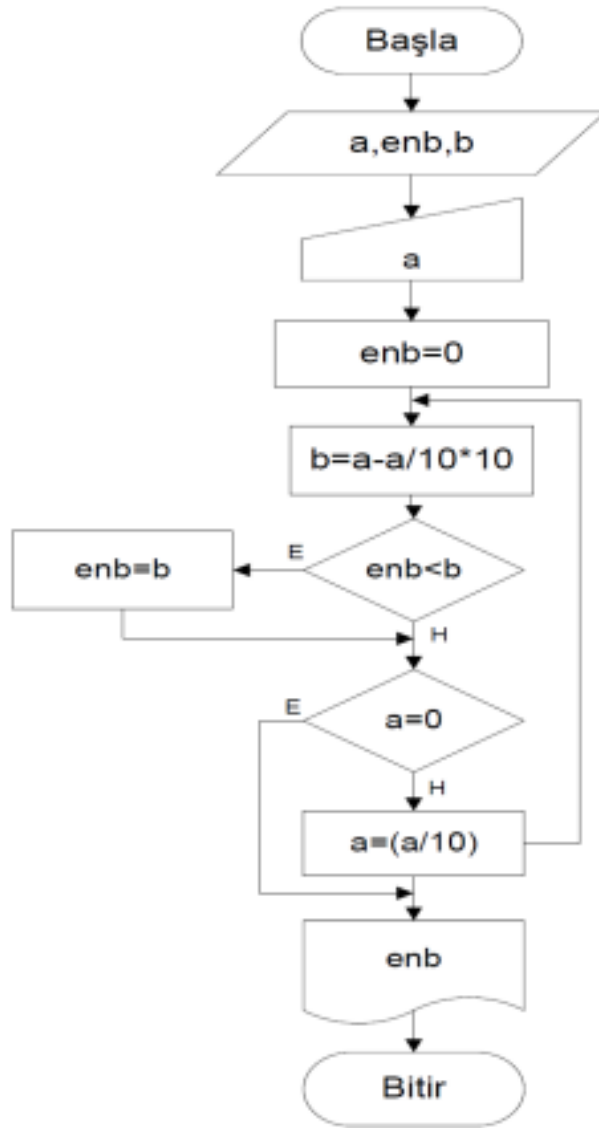
    // Ondalık kısmın hane sayısını bul
    while (ondalik_kisim - (int)ondalik_kisim != 0) {
        ondalik_kisim *= 10;
        ondalik_kisim_hane_sayisi++;
    }

    printf("Tam Kısım Hane Sayısı: %d\n", tam_kisim_hane_sayisi);
    printf("Ondalık Kısım Hane Sayısı: %d\n", ondalik_kisim_hane_sayisi);

    return 0;
}
```

2) Girilen bir tam sayının hanelerindeki en büyük sayıyı bulan algoritmanın akış şemasını çizin ve C kodunu yazın.

1. Kullanıcıdan bir tam sayı alın.
2. Eğer sayı negatifse, işlemi pozitifeye dönüştürmek için negatif işaretini kaldırın, çünkü en büyük hane işarete bağlı değil.
3. Sayının her basamağını sırayla inceleyin. Bunun için bir `while` döngüsü kullanırız. Sayı sıfıra eşit olana kadar her seferinde son basamağı alırız.
4. Her basamağı alırken, en büyük hane ile karşılaştırırız. Eğer o ana kadar bulunan en büyük haneden daha büyükse, en büyük hane değerini güncelleriz.
5. Döngü sona erdiğinde, en büyük hane değerini ekrana yazdırırız.



```
#include <stdio.h>

int main() {
    int sayi, enBuyukHane = 0;

    printf("Bir tam sayi girin: ");
    scanf("%d", &sayi);

    // Negatif sayıyı pozitif'e dönüştür
    if (sayi < 0) {
        sayi = -sayi;
    }

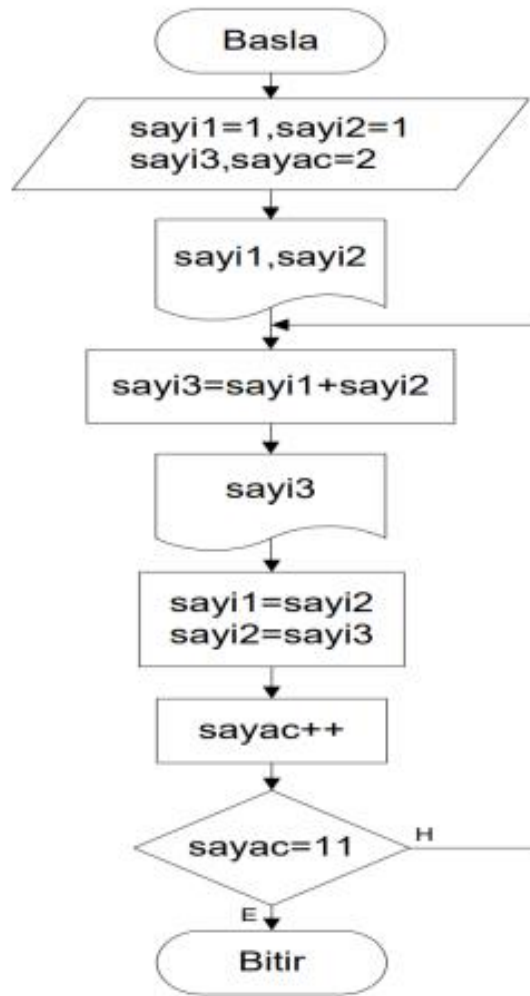
    // En büyük hane bulma işlemi
    while (sayi > 0) {
        int hane = sayi % 10; // Son basamağı al
        if (hane > enBuyukHane) {
            enBuyukHane = hane;
        }
        sayi /= 10; // Son basamağı çıkar
    }

    printf("En büyük hane: %d\n", enBuyukHane);

    return 0;
}
```

3. Fibonacci serisinin ilk 10 terimini ekrana basan algoritmanın akış şemasını çizin ve C kodunu yazın.

1. İlk iki Fibonacci terimini ($a=0$, $b=1$) başlangıç değerleri olarak ayarla.
2. Sayacı (n) sıfırla.
3. Ekrana a değerini yazdır.
4. Sayacı (n) bir artır.
5. Geçici bir değişkende ($temp$) a değerini sakla.
6. a değerini b ile değiştir.
7. b değerini ($temp + b$) ile değiştir.
8. Ekrana a değerini yazdır.
9. Sayacı (n) bir artır.
10. Eğer sayacı (n) hedef terim sayısına (örneğin, 10) eşitse, akışı sonlandır.
11. Değilse, adıma 5'e geri dön ve Fibonacci dizisini bir sonraki terime devam et.




```
#include <stdio.h>

int main() {
    int sayi1 = 0, sayi2 = 1, terim_sayisi = 2; //

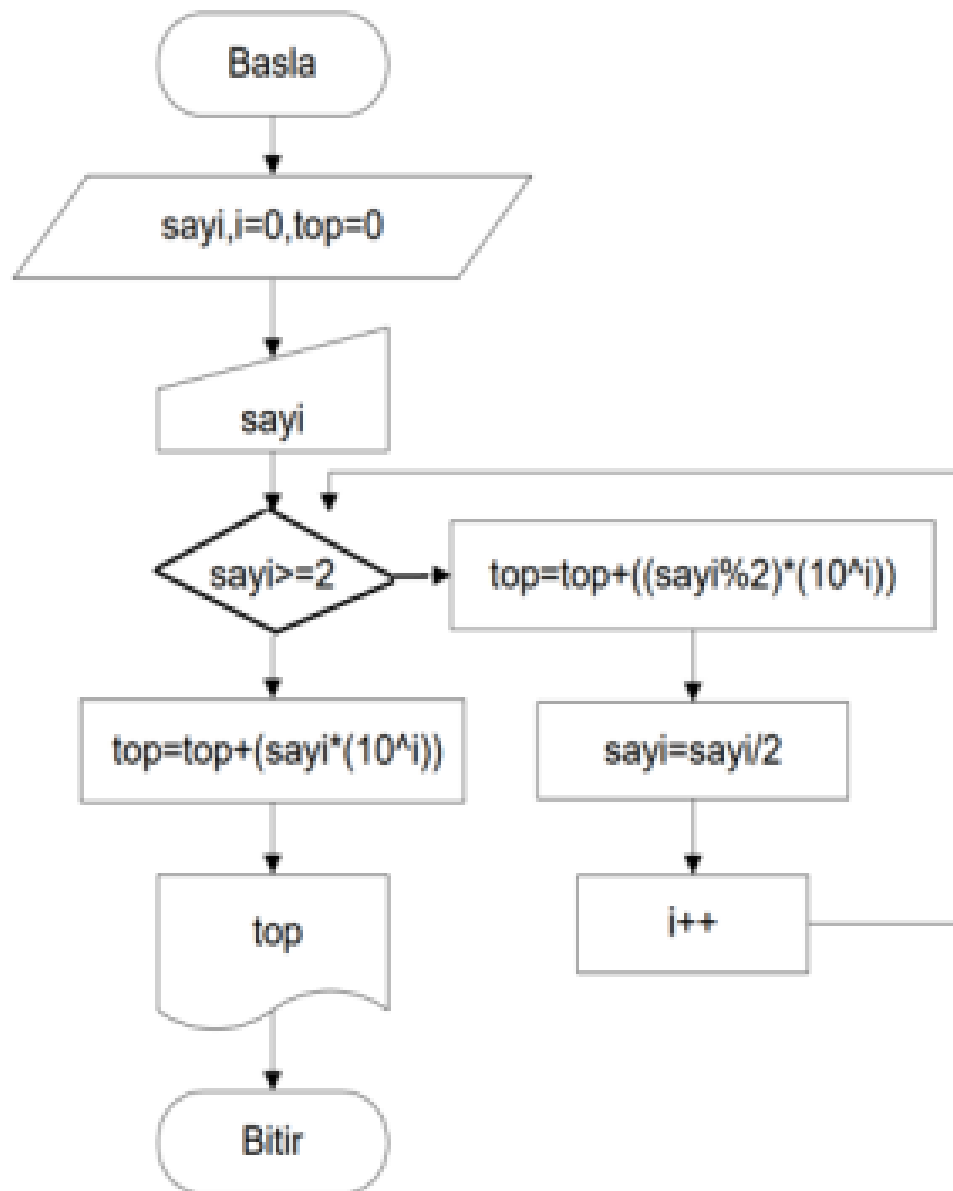
    printf("Fibonacci Serisi (İlk 10 Terim):\n");
    printf("%d\n%d\n", sayi1, sayi2);

    while (terim_sayisi < 10) {
        int fibonacci_terimi = sayi1 + sayi2;
        printf("%d\n", fibonacci_terimi);
        sayi1 = sayi2;
        sayi2 = fibonacci_terimi;
        terim_sayisi++;
    }

    return 0;
}
```

4. Girilen decimal (onluk) bir sayının binary (ikilik) bir sayıya dönüştüren programın algoritmanın akış şemasını çizin ve C kodunu yazın.

1. Kullanıcıdan bir onluk sayı girmesini isteyin.
2. Kullanıcının girdiği onluk sayıyı bir değişkende saklayın.
3. Bir dizi veya başka bir değişken oluşturun, bu değişken sonuç ikilik sayısını saklamak için kullanılacaktır. İlk başta bu değişkeni boş olarak düşünün.
4. Onluk sayıyı ikilik sayıya çevirirken, sayıları bölerek ve kalanları kullanarak işlem yapacağız. Bir döngü kullanarak işlemi gerçekleştireceğiz.
5. Döngü başlatın ve şu adımları gerçekleştirin:
 - a. Onluk sayıyı 2'ye bölün ve kalanı hesaplayın. Kalan, ikilik sayının bir basamağıdır.
 - b. Kalanı sonuç ikilik sayısının başına ekleyin veya bir diziye ekleyin (tersten başa eklemek daha sonra işleri kolaylaştırır).
 - c. Onluk sayıyı bölüm sonucuna atayın (tam sayı bölmesi yaparak).
 - d. Onluk sayı 0 olduğunda döngüden çıkın.
6. Döngü bittiğinde, ikilik sayıyı düzgün sırayla alın ve ekrana yazdırın veya sonucu bir dizi içinde saklayın ve dizi içeriğini tersten başlayarak yazdırarak ikilik sayıyı elde edin.



```
#include <stdio.h>

int main() {
    int onluk, kalan, i = 0;
    int ikilik[32]; // İkilik sayıyı saklamak için bir dizi (32 bitlik)

    printf("Onluk (decimal) sayı giriniz: ");
    scanf("%d", &onluk);

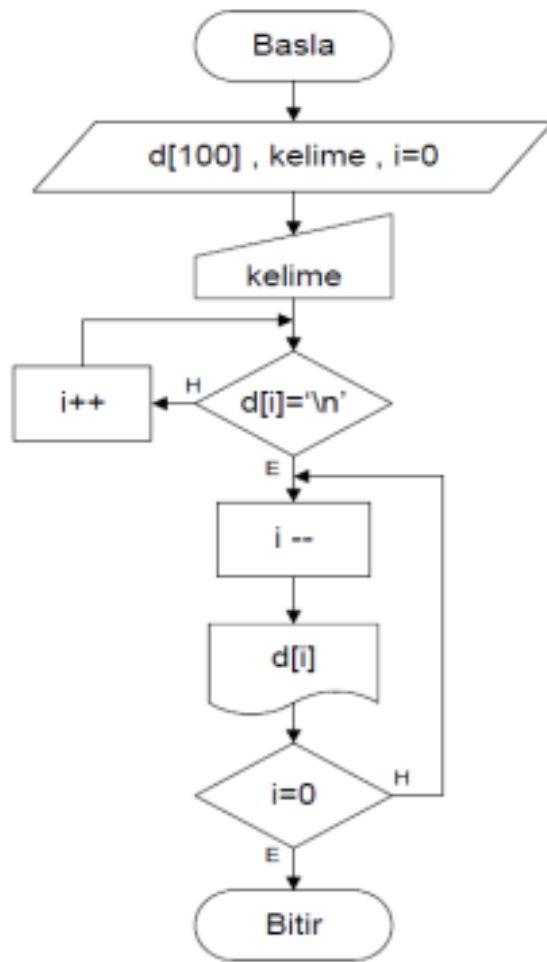
    while (onluk > 0) {
        kalan = onluk % 2;
        ikilik[i] = kalan;
        onluk = onluk / 2;
        i++;
    }

    printf("İkilik (binary) sayı: ");
    for (int j = i - 1; j >= 0; j--) {
        printf("%d", ikilik[j]);
    }

    return 0;
}
```

5. Girilen kelimeyi tersten yazdıran programın algoritmanın akış şemasını çizin ve C kodunu yazın.

1. Kullanıcıdan bir kelime girmesini isteyin.
2. Girilen kelimeyi bir değişkende saklayın.
3. Kelimenin uzunluğunu hesaplayın.
4. Bir döngü başlatın ve döngüyü kelimenin sonundan başlayarak başlangıcına doğru çalıştırın.
5. Her döngü adımında, kelimenin karakterlerini tersten yazdırın veya başka bir değişkende saklayın.
6. Döngü tamamlandığında, tersten yazılmış kelimeyi ekrana yazdırın.



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

int main() {
    char kelime[100]; // Kelimenin uzunluğunu kabul etme
    int uzunluk, i;

    printf("Bir kelime giriniz: ");
    scanf("%s", kelime);

    uzunluk = strlen(kelime);

    printf("Kelimenin ters hali: ");
    for (i = uzunluk - 1; i >= 0; i--) {
        printf("%c", kelime[i]);
    }

    return 0;
}
```

6. Girilen bir sayıyı roma rakamına çevirmenin C kodunu yazınız.


```
#include <stdio.h>
```

```
void printRoman(int num) {
```

```
    if (num <= 0 || num >= 4000) {
```

```
        printf("Geçersiz sayı\n");
```

```
        return;
```

```
    }
```

```
    // Roma rakamları ve değerleri
```

```
    int values[] = {1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1};
```

```
    char* numerals[] = {"M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "IV", "I"};
```

```
    for (int i = 0; num > 0; i++) {
```

```
        while (num >= values[i]) {
```

```
            printf("%s", numerals[i]);
```

```
            num -= values[i];
```

```
        }
```

```
    }
```

```
    printf("\n");
```

```
}
```

```
int main() {
```

```
    int num;
```

```
    printf("Bir sayı girin: ");
```

```
    scanf("%d", &num);
```

```
    printRoman(num);
```

```
    return 0;  
}
```