



ERCIYES ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ BİLGİSAYAR
MÜHENDİSLİĞİ BÖLÜMÜ PROGRAMMING LABORATORY NOTLARI



1. HAFTA NOTU (C Programlama)

*Bu not dersin 1. haftasında yapacağınız uygulamaya hazırlanmanızı
sağlayacak olup temel kavramları anlatan bilgileri içermektedir.*

DERSİN ÖĞRETİM ÜYESİ

Prof. Dr. Bahriye AKAY

DENEYİ YAPTIRAN ÖĞRETİM ELEMANI

Arş. Gör. Hayri ÖZTÜRK

2026
KAYSERİ

1-) Deneyde Kullanılacak Kütüphaneler Ve Fonksiyonlar Hakkında Kısa Bilgi

1.1 stdio.h

stdio.h (Standard Input Output), C dilinin en temel ve en yaygın kullanılan kütüphanelerinden biridir. Bu kütüphane, standart giriş (input) ve standart çıkış (output) işlemlerini gerçekleştirmek için gerekli fonksiyonları içerir. Konsol uygulamalarında kullanıcı ile etkileşim büyük ölçüde bu kütüphane aracılığıyla sağlanır. Bu kütüphanede yer alan en temel fonksiyonlardan biri printf() fonksiyonudur. printf, formatlı çıktı üretmek için kullanılır ve ekrana metin, sayı veya karakter yazdırılmasını sağlar. Bu fonksiyonun en önemli özelliği, biçimlendirme belirteçleri aracılığıyla farklı veri türlerini kontrollü bir şekilde çıktı olarak verebilmesidir. Örneğin %d tamsayıları, %c karakterleri, %s karakter dizilerini, %f ise ondalıklı sayıları temsil eder. Bu sayede hem veri türüne uygun çıktı üretimi sağlanır hem de okunabilir ve düzenli bir çıktı formatı elde edilir. Bu kütüphanede, scanf() fonksiyonu da yer almaktadır. scanf, standart girişten veri almak için kullanılır ve format belirteçleri aracılığıyla kullanıcıdan alınan verinin türünü tanımlar.

1.2 stdlib.h

stdlib.h (Standard Library), genel amaçlı yardımcı fonksiyonları içeren bir standart C kütüphanesidir. Bellek yönetimi, rastgele sayı üretimi, dönüşüm fonksiyonları ve sistem çağrıları gibi çeşitli işlevleri kapsar. Bu yönüyle stdlib.h, programın işletim sistemiyle ve çalışma ortamıyla etkileşimini kolaylaştıran bir altyapı sunar. Bu kütüphanede yer alan system() fonksiyonu, program içerisinden işletim sistemine komut gönderilmesini sağlar. Örneğin system("cls") komutu, Windows işletim sisteminde konsol ekranını temizlemek amacıyla kullanılır. Bu tür bir kullanım, programın doğrudan işletim sistemi komut yorumlayıcısı ile iletişim kurmasına imkân tanır.

1.3 conio.h

conio.h (Console Input Output), standart C kütüphanesinin bir parçası değildir ve genellikle belirli derleyiciler (örneğin Windows tabanlı bazı C derleyicileri) tarafından sağlanır. Bu kütüphane, konsol tabanlı giriş-çıkış işlemlerine yönelik ek fonksiyonlar içerir ve özellikle etkileşimli uygulamalarda kullanılır. Bu kütüphanenin en bilinen fonksiyonlarından biri getch() fonksiyonudur. getch(), kullanıcıdan tek bir karakterlik giriş alır ve bu işlemi gerçekleştirmek için Enter tuşuna basılmasını beklemes. Bu özellik, gerçek zamanlı tepki gerektiren uygulamalarda büyük avantaj sağlar. Kullanıcı tuşa bastığı anda program akışı devam eder; böylece daha dinamik ve akıcı bir etkileşim elde edilir. Standart C kütüphanesinin parçası olmaması nedeniyle conio.h taşınabilirlik açısından sınırlamalar içerebilir. Bununla birlikte, konsol tabanlı ve etkileşim odaklı uygulamalarda hızlı giriş kontrolü sağladığı için öğretim amaçlı projelerde yaygın olarak kullanılmaktadır.

1.4 time.h

time.h kütüphanesi, zaman ölçümü ve tarih işlemleri için kullanılan standart C başlık dosyalarından biridir. Programın çalışma süresini ölçmek, zaman damgası almak veya iki zaman değeri arasındaki farkı hesaplamak gibi işlemler bu kütüphane aracılığıyla gerçekleştirilir. time(NULL) fonksiyonu, sistem saatinden mevcut zamanı alır ve bunu saniye cinsinden bir time_t değeri olarak döndürür. İki farklı zaman değeri arasındaki fark alınarak geçen süre hesaplanabilir. Bu yöntem, programın başlangıç anı ile mevcut zaman arasındaki farkın bulunmasını ve böylece çalışma süresinin ölçülmesini mümkün kılar.

2-) DENEYDE KULLANILACAK PROGRAMLAMA YAPILARI HAKKINDA KISA BİLGİ

2.1 Veri Türleri ve Bellek Yapısı

2.1.1 Makrolar ve Sabit Tanımları

C programlama dilinde #define yönergesi kullanılarak makro sabitler tanımlanır. Bu yapı, derleme aşamasında metinsel yer değiştirme yapar; yani derleyici kodu işlerken makro adı yerine tanımlanan değeri yerleştirir. Bu nedenle makrolar çalışma zamanında bellekte ek bir alan kaplamazlar. Makro sabitlerin kullanımı, kodun okunabilirliğini ve sürdürülebilirliğini artırır. Özellikle sabit sayısal değerlerin doğrudan kod içine yazılması yerine anlamlı isimlerle tanımlanması, programın daha anlaşılır ve yönetilebilir olmasını sağlar. Boyut sabitleri, karakter temsilleri veya sembolik ifadeler gibi sabit değerler makrolar aracılığıyla tanımlanabilir. Makrolar genellikle global kapsamlıdır ve programın her yerinden erişilebilir; bu durum, ortak sabitlerin merkezi bir biçimde yönetilmesine imkân tanır.

2.1.2 Temel Veri Türleri

C dilinde her değişken belirli bir veri türü ile tanımlanır ve bu tür, değişkenin bellekte kaplayacağı alanı ve saklayabileceği veri tipini belirler. int veri türü tamsayı değerleri temsil eder ve sayısal hesaplamalarda kullanılır. char veri türü tek bir karakter saklamak için kullanılır ve genellikle sembolik temsiller veya karakter dizilerinin yapı taşıdır. time_t, zaman verisini temsil eden özel bir türdür ve sistem saatine ilişkin işlemlerde kullanılır. void ise özellikle fonksiyon tanımlarında kullanılır; fonksiyonun herhangi bir değer döndürmediğini ifade eder.

Veri türleri, belleğin düzenli ve kontrollü kullanılmasını sağlayan temel yapı taşlarıdır.

2.1.3 Diziler (Arrays)

Diziler, aynı veri türünden birden fazla değeri ardışık bellek alanlarında saklamaya yarayan veri yapılarıdır. İki boyutlu diziler ise satır ve sütun mantığıyla organize edilmiş matris yapıları oluşturur. Örneğin: char deneme[10][7] şeklindeki bir tanım, karakter türünde iki boyutlu bir veri yapısını ifade eder. C dili dizileri bellekte satır-satır biçiminde tutar; yani önce birinci satırın tüm elemanları, ardından ikinci satırın elemanları saklanır. Her satır, karakter dizisi (string) gibi davranabilir.

2.2 Kontrol Yapıları

2.2.1 Döngüler

Döngüler, belirli bir koşul sağlandığı sürece kod bloklarının tekrar çalıştırılmasını sağlar. while döngüsü, koşul doğru olduğu sürece çalışır ve genellikle belirsiz tekrar sayısına sahip durumlarda tercih edilir. for döngüsü ise sayaç kontrollü tekrarlar için uygundur ve başlangıç, koşul ve artış ifadelerini tek satırda barındırır. Özellikle iki boyutlu veri yapılarında iç içe döngüler kullanılarak satır ve sütun bazlı işlemler gerçekleştirilir.

2.2.2 Koşul Yapıları

Koşul yapıları, programın farklı durumlara göre farklı davranışlar sergilemesini sağlar. if yapısı, belirli bir mantıksal koşulun doğru olup olmadığını kontrol eder ve koşula bağlı olarak kod çalıştırır. switch yapısı ise tek bir değişkenin farklı sabit değerlere göre dallanmasını sağlar. Bu yapı, çoklu seçenek içeren durumlarda daha düzenli ve okunabilir bir kontrol mekanizması sunar.

Ders Katkıları

Bu uygulama kapsamında kullanılan yapılar, öğrencinin aşağıdaki teknik kazanımları edinmesini sağlar:

1. Algoritmik Düşünme

- Durum geçişlerinin modellenmesi
- Koşullu kontrol mekanizmalarının yapılandırılması
- Olay-temelli program akışının kavranması

2. Bellek ve Veri Yapıları Bilinci

- İki boyutlu dizilerin bellekteki temsilini anlama
- Global ve yerel değişken farkını kavrama
- Durum bilgisinin korunması

3. Modüler Programlama

- Fonksiyonel ayrıştırma
- Parametre kullanımı
- Kodun yeniden kullanılabilirliğini sağlama

4. Zaman ve Performans Kavramı

- Sistem zamanı ile çalışma
- Gerçek zamanlı ölçüm mantığı
- Süre hesaplama algoritması

5. Girdi-Çıktı Kontrolü

- Anlık klavye okuma teknikleri
- Formatlı çıktı üretimi
- Konsol uygulamalarında kullanıcı etkileşimi

6. Yazılım Mühendisliği Perspektifi

- Okunabilir kod yazımı
- Sabitlerin makro ile tanımlanması
- Mantıksal ayırım ve soyutlama