İskenderun Teknik Üniversitesi

Fakülte: Mühendislik ve Doğa Bilimleri *Fakültesi*
Bölüm: ***Bilgisayar Mühendisliği Bölümü***
Ders: Algoritmalar ve Programlama
Dönem: *2021 – 2022 (Güz)*
Öğretim üyesi: *Dr. Öğr. Üyesi Mehmet SARIGÜL*

6.
Hafta

Yapılar

- Sıralı yapılar:
  - Varsayılan olarak sırayla yürütülen programlar
- Seçim yapıları:
  - if, if… else ve geçiş yap
- Tekrar yapıları:
  - while, do ... while and for

Programlarda tekrar blokları

- döngü
  - bir programdaki bir grup adımı tekrarlayan bir kontrol yapısı
- döngü gövdesi
  - döngüde tekrarlanan ifadeler

- Sayaç kontrollü yineleme
  - Karşı kontrollü yinelemeye bazen kesin yineleme denir çünkü döngünün tam olarak kaç kez yürütüleceğini önceden biliyoruz.
- Sentinel kontrollü yineleme
  - Nöbetçi kontrollü yineleme, döngünün kaç kez yürütüleceği önceden bilinmediği için bazen belirsiz yineleme olarak adlandırılır.

Programlarda tekrar blokları

- Sayaç kontrollü yinelemede, yinelemelerin sayısını saymak için bir kontrol değişkeni kullanılır.
- Kontrol değişkeni, talimat grubu her gerçekleştirildiğinde (genellikle 1 oranında) artırılır.
- Kontrol değişkeninin değeri doğru sayıda yineleme gerçekleştirildiğini gösterdiğinde, döngü sona erer ve yineleme ifadesinden sonraki yürütme ifadesi ile devam eder.

Programlarda tekrar blokları

- Sentinel değerleri, aşağıdaki durumlarda yinelemeyi kontrol etmek için kullanılır:
  - Kesin yineleme sayısı önceden bilinmemektedir ve
  - Döngü, döngü her gerçekleştirildiğinde verileri alan ifadeleri içerir.
- Sentinel değeri "verinin sonunu" gösterir.
- Sentinel, tüm düzenli veri öğeleri programa sağlandıktan sonra girilir.
- Sentinel, normal veri öğelerinden farklı olmalıdır.

Programlarda tekrar blokları

Sayma Döngüleri

- Sayaç kontrollü yineleme şunları gerektirir:
  - Bir kontrol değişkeninin (veya döngü sayacının) adı.
  - Kontrol değişkeninin başlangıç değeri.
  - Döngü boyunca her seferinde kontrol değişkeninin değiştirildiği artış (veya azalma).
  - Kontrol değişkeninin son değerini test eden koşul (yani döngünün devam edip etmeyeceği).

- while (tekrar koşulu)
- {
- ifade;
- }

while döngüsü

- count =0;
- while(count<N)
- {
- printf("*\n");
- count = count + 1;
- }

while yazım kuralları

- 1. Döngü kontrol değişkenini başlatın.
- 2. Çıkış koşulu karşılanmadığı sürece
- 3. İşlemeye devam edin

Genel Koşullu Döngü

```
int count = 1;
while(count<=10)
{
        printf("%d",count);
        count++;
}
```

- döngü kontrol değişkeninin başlatılması
- döngü tekrarlama koşulunun testi
- döngü kontrol değişkeninin değiştirilmesi (güncellenmesi)

- for döngüsü, bu üç bileşenin her biri için belirlenmiş bir yer sağlar

Döngü Kontrol Bileşenleri

for döngüsü

- for (başlangıç ifadesi; döngü tekrar koşulu; güncelleme ifadesi)
- ifade;

- for (count=0; count< N ; count += 1)
- printf("*");

- for (int count=1; count<=10; count++)
- printf("%d",count);

- For ifadesi yürütülmeye başladığında, kontrol değişkeni sayacı 1 olarak başlatılır.
- Daha sonra döngü devam koşulu sayacı <= 10 kontrol edilir.
- Count başlangıç değeri 1 olduğu için koşul karşılanmıştır, bu nedenle cout ifadesi counter değerini yani 1'i yazdırır.
- Kontrol değişkeni sayacı daha sonra ifade sayacı ++ tarafından artırılır ve döngü, döngü sürdürme testi ile yeniden başlar.

for döngüsü

- Kontrol değişkeni artık 2'ye eşit olduğundan, son değer aşılmaz, bu nedenle program cout ifadesini yeniden gerçekleştirir.
- Bu işlem, kontrol değişkeni sayacı 11 olan son değerine yükseltilene kadar devam eder - bu, döngü devam testinin başarısız olmasına neden olur ve yineleme sona erer.
- Program, for ifadesinden sonraki ilk ifadeyi gerçekleştirerek devam eder (bu durumda programın sonu).

for döngüsü

- For ifadesi bir döngü için gereken "herşeyi yapar" - bir kontrol değişkeniyle karşı kontrollü yineleme için gereken öğelerin her birini belirttiğine dikkat edin.

- For öğesinin gövdesinde birden fazla ifade varsa, döngünün gövdesini tanımlamak için kaşlı ayraçlar gerekir. {}

Genel hatalar

- Döngü devam koşulu sayacının <= 10 olduğuna dikkat edin.
- Sayaç <10'u yanlış yazdıysanız, döngü yalnızca 9 kez çalıştırılır.
- Bu, tek tek hata adı verilen yaygın bir mantık hatasıdır.

Genel hatalar

- Bir while veya for ifadesi koşulunda nihai değeri kullanmak ve <= ilişkisel operatörünü kullanmak, tek tek hataları önlemeye yardımcı olabilir. Örneğin, 1'den 10'a kadar olan değerleri yazdırmak için kullanılan bir döngü için, döngü devam koşulu, sayaç <11 veya sayaç <10 yerine sayaç <= 10 olmalıdır.

- Genellikle, başlatma ve artış ifadeleri virgülle ayrılmış ifade listeleridir.
- Burada kullanıldığı şekliyle virgül, ifade listelerinin soldan sağa doğru değerlendirilmesini garanti eden aslında virgül operatörleridir.

Virgülle Ayrılmış İfade Listeleri

- Virgül operatörü genellikle for ifadesinde kullanılır.
- Birincil kullanımı, çoklu başlatma ve / veya çoklu artış ifadeleri kullanmanızı sağlamaktır.
- Örneğin, tek bir for deyiminde başlatılması ve artırılması gereken iki kontrol değişkeni olabilir.

Virgülle Ayrılmış İfade Listeleri

- int a = 1, b = 2;

- for (i=-1,j=0;i<5;j=i+1){
- …
- }

- For ifadesindeki üç ifade de isteğe bağlıdır.
- Koşul ifadesi atlanırsa, C koşulun doğru olduğunu varsayar ve böylece sonsuz bir döngü oluşturur.
- Kontrol değişkeni programın başka bir yerinde başlatılmışsa, başlatma ifadesini atlayabilirsiniz.
- For ifadesinin gövdesindeki ifadelerle hesaplanırsa veya herhangi bir artış gerekmiyorsa artış atlanabilir.

İsteğe bağlı ifadeler

- For ifadesindeki artış ifadesi, for ifadesinin gövdesinin sonunda bağımsız bir C++ ifadesi gibi davranır.
- Bu nedenle, aşağıdaki ifadeler eşdeğerdir.
  - sayaç = sayaç + 1
  - sayaç + = 1
  - ++sayaç
  - sayaç++
- Burada önceden artırılan veya sonradan artırılan değişken daha büyük bir ifadede görünmediğinden, her iki artırma biçimi de aynı etkiye sahiptir.
- For ifadesindeki iki noktalı virgül gereklidir.

- Başlatma, döngü devam koşulu ve artış aritmetik ifadeler içerebilir. Örneğin, x = 2 ve y = 10 ise, aşağıdaki ifadeler eşdeğerdir.
  - for (j = x; j <= 4 * x * y; j += y / x)
  - for (j = 2; j <= 80; j += 5)

- "Artış" negatif olabilir (bu durumda bu gerçekten bir azalmadır ve döngü aslında aşağı doğru sayar).
- Döngü sürdürme koşulu başlangıçta yanlışsa, döngü gövdesi çalışmaz. Bunun yerine, yürütme, for ifadesini izleyen ifadeyle devam eder.

for döngü örnekleri

- Aşağıdaki örnekler, bir for ifadesindeki kontrol değişkenini değiştirme yöntemlerini gösterir.

- Kontrol değişkenini 1'lik artışlarla 1'den 100'e kadar değiştirin.
  - `for (i = 1; i <= 100; i++ )`
- Kontrol değişkenini -1'lik artışlarla (1'lik düşüşler) 100'den 1'e değiştirin.
  `for (i = 100; i >= 1; i--)`
- 7'lik adımlarla kontrol değişkenini 7'den 77'ye değiştirin.
  `for (i = 7; i <= 77; i += 7)`
- Kontrol değişkenini -2'lik adımlarla 20'den 2'ye değiştirin.
  - `for (i = 20; i >= 2; i -= 2)`
- Kontrol değişkenini aşağıdaki değerler dizisi üzerinden değiştirin: 2, 5, 8, 11, 14, 17.
  - `for (j = 2; j <= 17; j += 3)`
- Kontrol değişkenini aşağıdaki değerler dizisi üzerinden değiştirin: 44, 33, 22, 11, 0.
  `for (j = 44; j >= 0; j -= 11)`

- 2'den 100'e kadar Tam Sayıları Toplayın

Uygulama

Ödevler

- Kullanıcı tarafından girilen bir tamsayının faktöriyelini bulmak için bir C++ programı yazın.
- Kullanıcı tarafından girilen bir tam sayıdaki basamak sayısını saymak için bir C++ programı yazın.

Ders Sonu