

Conditional Expression

Koşul ifadesi bir kontrol yapısıdır. Bunun için üçlü operatör (**Ternary Operator**) kullanılır. Üç operant aldığı için bu isimle bilinir. If else yapılarının kısa halidir. Kullanımı aşağıdaki gibidir.

```
condition ? operand : operand
```

Eğer ilk operant olan **condition** kısmı doğru ise ikinci operant(soru işaretinden sonraki), condition kısmı yanlış ise üçüncü operant (iki noktadan sonraki) değer üretilir.

Ör:

```
#include <stdio.h>

int main(void)
{
    int minutes;
    char s;

    printf("How many minutes left ? ");
    scanf("%d", &minutes);

    s = minutes > 1 ? 's' : ' ';    // Conditional Expression

    printf("%d minute%c left\n", minutes, s);

    return 0;
}
```

Dangling Else

Blok parantez kullanılmayan iç içe if-else yapılarında (**Nested if-else**) bir belirsizlik ortaya çıkar. Aşağıdaki kod parçalarını dikkatlice inceleyin.

```
// Problem: Ambiguity
if (grade < 50)
    if (sup == 1)
        printf("Sup\n");
else // <-- Does this belong to 'if (grade < 50)' OR 'if (sup == 1)'?
    printf("Pass\n");
```

else hangi **if** 'e ait? Veya aşağıdaki durumu inceleyin:

```
// Interpretation #2:
if (grade < 50)
    if (sup == 1) // <-- now the formatting suggests this 'if'
        printf("Sup\n");
    else
        printf("Pass\n");
```

C dili sarkan **else** 'i (**dangling else**) her zaman içteki **if** 'e ekler. Girintilemenin nasıl olacağına bakmaz. İlk örnekteki durumda eğer dıştaki **if** 'e **else** 'i dahil etmek istiyorsak aşağıdaki gibi blok parantezleri kullanırız.

```
if (grade < 50)
{
    if (sup == 1)
    {
        printf("Sup\n");
    }
}
else
{
    printf("Pass\n");
}
```

Workshop #4

Ödevi Zamanında Verememe/Eksik Ödev Teslimi

- Bir Paket 10'lu kahve getirilecek.
- Mikro işlemci mimarisi çeşitlerini ve temel olarak ne olduklarını birkaç cümlecik olarak anlatacak.

Giriş

Bu çalışmada, kullanıcıdan veriler alıp bu verileri uygun değişkenlerde saklayacak Verilen probleme göre gerekli sıralama, seçim, yinleme rutinlerini kodlayacak Ve istenen çıktıyı görüntüleyecek bir C programı kodlayacak, derleyecek ve yürüteceksiniz.

Konular

- Mantık
- Kontrol yapıları
- Döngüler

Beklenti

- Kontrol yapılarını kullanarak seçim yapıp karar verebilecek kodları yazabileceksiniz.
 - Döngü yapılarını kullanarak yinlenebilir kodlar yazabileceksiniz.
 - Bir mantıksal bloğu başka bir mantıksal bloğa yerleştirebileceksiniz.
-

Part-1 (%10)

Zip dosyanızı bilgisayarınızda Cdersleri\Workshops\ klasörünüze çıkarın. WS04 klasörünü VSC editörünüz ile proje olarak açın.

1. “**Part-1 Output Example**” kısmında programın çıktılarını, programın nasıl çalışacağını anlamak için dikkatlice incleyin.
2. Kodlarınızı proje klasörünüzdeki p1 klasöründe bulunan “**w4p1.c**” dosyasına yazın.
3. Kodunuzda ikiden fazla **integer** değişken ve birden fazla **char** değişken bulunamaz.
4. Tek bir **scanf** çağrısında kullanıcıdan iki ayrı veri alacaksınız. İlk değer istenen döngü türünü temsil edeceği için **char** tipinde olacak ve ikinci değer yineleme sayısını (**iteration** object) tutacak. Tek karakter alırken programımız tanımsız davranışlar (**unexpected behavior**) sergileyebilir. Şimdilik bu durumdan kaçınmak için aşağıdaki **scanf** formatını kullanabilirsiniz.

```
scanf(" %c%d", ...
```

5. Program, döngü tipini belirleyecek olan karakter için yalnızca büyük harf (**uppercase**) kabul edecektir (D, W, F, Q). Diğer bütün değerler hata göstermelidir. Uygun yineleme yapısı D=do..while, W=while, F=for, Q=quit.
6. İstenen yineleme sayısı 3 ila 20 arasında (20 dahil) olmalıdır. Diğer bütün değerlerde kodunuz hata göstermeli. Tek istisna programdan çıkmayı simgeleyen 0 (sıfır) değeridir.

Not

- Derlenen dosyaya Windows sistemlerde “**prg_w4p1.exe**”, Linux sistemlerde “**prg_w4p1**” adı verilecek.
- Derleyici parametrelerinden **-Wall** ve **-Werror** kullanılacak.

Part-1 Oputput Example (SARI vurgulanan kısımlar kullanıcı girdileridir)

```
+-----+
Loop application STARTED
+-----+

D = do/while | W = while | F = for | Q = quit
Enter loop type and the number of times to iterate (Quit=Q0): R10
ERROR: Invalid entered value(s)!

D = do/while | W = while | F = for | Q = quit
Enter loop type and the number of times to iterate (Quit=Q0): D2
ERROR: The number of iterations must be between 3-20 inclusive!

D = do/while | W = while | F = for | Q = quit
Enter loop type and the number of times to iterate (Quit=Q0): D21
ERROR: The number of iterations must be between 3-20 inclusive!

D = do/while | W = while | F = for | Q = quit
Enter loop type and the number of times to iterate (Quit=Q0): W2
ERROR: The number of iterations must be between 3-20 inclusive!

D = do/while | W = while | F = for | Q = quit
Enter loop type and the number of times to iterate (Quit=Q0): W21
ERROR: The number of iterations must be between 3-20 inclusive!
```

D = do/while | W = while | F = for | Q = quit
Enter loop type and the number of times to iterate (Quit=Q0): F2
ERROR: The number of iterations must be between 3-20 inclusive!

D = do/while | W = while | F = for | Q = quit
Enter loop type and the number of times to iterate (Quit=Q0): F21
ERROR: The number of iterations must be between 3-20 inclusive!

D = do/while | W = while | F = for | Q = quit
Enter loop type and the number of times to iterate (Quit=Q0): Q-1
ERROR: To quit, the number of iterations should be 0!

D = do/while | W = while | F = for | Q = quit
Enter loop type and the number of times to iterate (Quit=Q0): Q1
ERROR: To quit, the number of iterations should be 0!

D = do/while | W = while | F = for | Q = quit
Enter loop type and the number of times to iterate (Quit=Q0): D3
DO-WHILE: DDD

D = do/while | W = while | F = for | Q = quit
Enter loop type and the number of times to iterate (Quit=Q0): W3
WHILE : WWW

D = do/while | W = while | F = for | Q = quit
Enter loop type and the number of times to iterate (Quit=Q0): F3
FOR : FFF

D = do/while | W = while | F = for | Q = quit
Enter loop type and the number of times to iterate (Quit=Q0): W20
WHILE : WWWWWWWWWWWWWWWWWWW

D = do/while | W = while | F = for | Q = quit
Enter loop type and the number of times to iterate (Quit=Q0): F20
FOR : FFFFFFFFFFFFFFFFFFFFFF

D = do/while | W = while | F = for | Q = quit
Enter loop type and the number of times to iterate (Quit=Q0): D20
DO-WHILE: DDDDDDDDDDDDDDDDDDD

D = do/while | W = while | F = for | Q = quit
Enter loop type and the number of times to iterate (Quit=Q0): d0
ERROR: Invalid entered value(s)!

D = do/while | W = while | F = for | Q = quit
Enter loop type and the number of times to iterate (Quit=Q0): w0
ERROR: Invalid entered value(s)!

D = do/while | W = while | F = for | Q = quit
Enter loop type and the number of times to iterate (Quit=Q0): f0
ERROR: Invalid entered value(s)!

D = do/while | W = while | F = for | Q = quit
Enter loop type and the number of times to iterate (Quit=Q0): Q0

```
+-----+  
Loop application ENDED  
+-----+
```

Part-2 (%40)

1. Programın nasıl çalıştığına dair fikir edinmek için “**Part-2 Output Example**” kısmını dikkatlice inceleyin.
2. Çözümünüzü p2 klasöründe yer alan **w4p2.c** dosyasına kodlayın.
3. Yapmanız gereken ilk şey, yapılacak alışverişte ihtiyacımız olan elma, portakal, armut, domates ve lahana sayısını belirlemek.
Sıfır (0) değeri verilmesi, öğenin gerekli olmadığı anlamına gelir. Ancak negatif değere izin verilmez. Böyle bir durumda hata mesajı görüntülenmelidir.
4. Ürün alma aşamasında seçilen ürün sayısını kullanıcıdan alırken sadece pozitif değere sahip olan (ihtiyaç miktarı belli olan) ürünler için soru sorulmalıdır.
5. Alınan ürün miktarı ihtiyaç duyulan ürün miktarından fazlaysa veya az ise kullanıcıya, uygun hata mesajlarını göstermelisiniz. Örnek çıktıyı iyi inceleyin
6. Alınan ürün miktarı sıfır(0) veya negatif olamaz. Uygun hata mesajı gösterilmeli.
7. Eğer ürün, ihtiyacımız olan miktar kadar alınırsa bir sonraki ihtiyacımız olan ürün için alıma geçilebilir.
8. İhtiyaç duyulan bütün ürünleri aldıktan sonra programınız alışverişe devam edip etmeyeceğini kullanıcıya sormalı. Kullanıcı cevap olarak sıfır (0) girene kadar program çalışmalı.

Not

- Derlenen dosyaya Windows sistemlerde “**prg_w4p2.exe**”, Linux sistemlerde “**prg_w4p2**” adı verilecek.
- Derleyici parametrelerinden **-Wall** ve **-Werror** kullanılacak.

Part-2 Output Example (SARI alanlar kullanıcı veri girişini temsil eder)

```
Grocery Shopping  
=====  
How many APPLES do you need? : -1  
ERROR: Value must be 0 or more.  
How many APPLES do you need? : 10  
  
How many ORANGES do you need? : -1  
ERROR: Value must be 0 or more.  
How many ORANGES do you need? : 0  
  
How many PEARS do you need? : -1  
ERROR: Value must be 0 or more.  
How many PEARS do you need? : 5  
  
How many TOMATOES do you need? : -1  
ERROR: Value must be 0 or more.  
How many TOMATOES do you need? : 0
```

How many CABBAGES do you need? : -1

ERROR: Value must be 0 or more.

How many CABBAGES do you need? : 15

Time to pick the products!

Pick some APPLES... how many did you pick? : 40

You picked too many... only 10 more APPLE(S) are needed.

Pick some APPLES... how many did you pick? : -2

ERROR: You must pick at least 1!

Pick some APPLES... how many did you pick? : 0

ERROR: You must pick at least 1!

Pick some APPLES... how many did you pick? : 5

Looks like we still need some APPLES...

Pick some APPLES... how many did you pick? : 40

You picked too many... only 5 more APPLE(S) are needed.

Pick some APPLES... how many did you pick? : 5

Great, that's the apples done!

Pick some PEARS... how many did you pick? : 40

You picked too many... only 5 more PEAR(S) are needed.

Pick some PEARS... how many did you pick? : -2

ERROR: You must pick at least 1!

Pick some PEARS... how many did you pick? : 0

ERROR: You must pick at least 1!

Pick some PEARS... how many did you pick? : 4

Looks like we still need some PEARS...

Pick some PEARS... how many did you pick? : 40

You picked too many... only 1 more PEAR(S) are needed.

Pick some PEARS... how many did you pick? : 1

Great, that's the pears done!

Pick some CABBAGES... how many did you pick? : 40

You picked too many... only 15 more CABBAGE(S) are needed.

Pick some CABBAGES... how many did you pick? : -2

ERROR: You must pick at least 1!

Pick some CABBAGES... how many did you pick? : 0

ERROR: You must pick at least 1!

Pick some CABBAGES... how many did you pick? : 12

Looks like we still need some CABBAGES...

Pick some CABBAGES... how many did you pick? : 40

You picked too many... only 3 more CABBAGE(S) are needed.

Pick some CABBAGES... how many did you pick? : 3

Great, that's the cabbages done!

All the items are picked!

Do another shopping? (0=NO): 1

Grocery Shopping

=====

How many APPLES do you need? : 0

How many ORANGES do you need? : 20

How many PEARS do you need? : 0

How many TOMATOES do you need? : 15

How many CABBAGES do you need? : 0

Time to pick the products!

Pick some ORANGES... how many did you pick? : 40

You picked too many... only 20 more ORANGE(S) are needed.

Pick some ORANGES... how many did you pick? : -2

ERROR: You must pick at least 1!

Pick some ORANGES... how many did you pick? : 0

ERROR: You must pick at least 1!

Pick some ORANGES... how many did you pick? : 5

Looks like we still need some ORANGES...

Pick some ORANGES... how many did you pick? : 40

You picked too many... only 15 more ORANGE(S) are needed.

Pick some ORANGES... how many did you pick? : 15

Great, that's the oranges done!

Pick some TOMATOES... how many did you pick? : 40

You picked too many... only 15 more TOMATO(ES) are needed.

Pick some TOMATOES... how many did you pick? : -2

ERROR: You must pick at least 1!

Pick some TOMATOES... how many did you pick? : 0

ERROR: You must pick at least 1!

Pick some TOMATOES... how many did you pick? : 12

Looks like we still need some TOMATOES...

Pick some TOMATOES... how many did you pick? : 40

You picked too many... only 3 more TOMATO(ES) are needed.

Pick some TOMATOES... how many did you pick? : 3

Great, that's the tomatoes done!

All the items are picked!

Do another shopping? (0=NO): 0

Your tasks are done for today - enjoy your free time!