

Promotion

Bir atama (**assignment**) işleminde sağ taraf değerinin (**rvalue**) tipi sol taraf değerinin (**lvalue**) tipinden küçükse sağ taraf değerinin tipi sol taraf değerinin tipine yükseltilir. Bu işleme **promotion** denir.

Ör:

```
#include <stdio.h>

int main(void)
{
    int loonies;
    double cash;

    printf("Loonies ? ");
    scanf("%d", &loonies);
    cash = loonies; // promotion
    printf("Cash is $%.2lf\n", cash);

    return 0;
}
```

Narrowing

Yine bir atama (**assignment**) işleminde sağ taraf değerinin tipi (**rvalue**) sol taraf değerinin tipinden (lvalue) büyükse sağ taraf değerinin tipi sol taraf değerinin tipine düşürülür. Bu işleme **narrowing** denir.

Örn:

```
#include <stdio.h>

int main(void)
{
    double cash;
    int loonies;

    printf("Cash ? ");
    scanf("%lf", &cash);
    loonies = cash; // truncation
    printf("%d loonies.\n", loonies);

    return 0;
}
```

Not:

Tiplerdeki yüksek tip ve düşük tip olayı tamamen hafızda kaplanan boyut ve tip önceliği ile ilgilidir.

long double	higher
double	...
float	...
long long	...
long	...
int	...
short	...
char	lower

Statement

Programlama dili tarafından anlamı olan ve işlenmesi gereken en küçük birime denir.

Expression

Bir veya daha fazla değişkenden, operatörden veya fonksiyondan meydana gelen ve programlama dili tarafından yorumlandığında bir değer döndüren ifadelerdir.

Not

Expression'lar birer statement'dir. Yani her expression bir statement'dir ama her statement bir expression değildir. Bir statement' in Expression olabilmesi için kriterimiz değer döndürmesidir (üretmesidir).

Precision

Ekrana basılacak sayısal verilerin basamaklarının hassasiyetidir. Yazılacak bir sayının kaç basamaklık yer kaplayacağı, boşta kalan basamakların yerine 0(sıfır) basılıp basılmayacağı, ondalık sayılarda noktadan sonra kaç basamak gösterileceği bilgisidir.

Ör:

```
000222 // 222 sayısı için ekranda 6 basamak yer ayrıldı. Sayı sağa yaslandı
022222 // ve boşta kalan kısımlara 0(sıfır) eklendi.

222 // 222 sayısı için ekranda 6 basamak yer ayrıldı. Sayı sağa yaslandı
22222 // ve boşta kalan kısımlara bir şey eklenmedi
```

Workshop #2

Ödevi Zamanında Verememe/Eksik Ödev Teslimi

- Bir Paket 10'lu kahve getirilecek.
- Ekrana 3 basamaklı palindromik sayıların sayısı(adedi) kadar “workshop 3’ü ilk “BEN” bitireceğim” yazacak. Elbette bu palindromik sayıların adedini sizin programınız hesaplayacak.

Giriş

Bu çalışmada kullanıcıdan sayısal veriler alacak, bu sayısal verileri uygun tipteki değişkenlerde saklayacak, bu verilerle parasal hesaplamalar yapacak, **modulus** operatörünü ve tip dönüşümü için **casting** operatörünü kullanacak bir c programı yazıp derleyip çalıştıracaksınız. (Kavramlara/kelimelere ve özellikle örnek çıktıdaki ifadelerle çok dikkat etmelisiniz. Araştırmamız gereken ifadeler olabilir)

Konular

- Veri tipleri
- Aritmetik ve karşılaştırma operatörleri
- Verilerle hesaplama işlemleri

Beklenti

- C operatörlerini ve ifadelerini(**Expression**) kullanarak basit hesaplamaları kodlayabilme
- scanf() ile kullanıcıdan veri alabilme
- Bir veri türünden diğerine değer atayabilme
- int ve float tipindeki sayıları ekrana basarken hassasiyet (**precision**) ayarı yapabilme
- Parasal değerlerde yuvarlama yapabilme

Part-1 (%10)

Zip dosyanızı bilgisayarınızda Cdersleri\Workshops\ klasörünüze çıkarın. WS02 klasörünü VSC editörünüz ile proje olarak açın.

1. “**Part-1 Output Example**” kısmında programın çıktılarına, programın nasıl çalışacağını anlamak için dikkatlice bakın.
2. Kodlarınızı proje klasörünüzdeki p1 klasöründe bulunan “**w2p1.c**” dosyasına verilenleri değiştirmeden yazın (verilen değişkenler değiştirilmeyecek vs.).
3. Üç(3) farklı gömlek bedeninin fiyatlarını saklamak için gerekli değişkenleri oluşturun. Değişken isimleri amacına uygun seçilmeli ve İngilizce olmalı, değişken türü saklanacak veri türüne uygun olmalıdır. Bu üç değişkenin haricinde ihtiyaç duyacağınız(aşağıdaki maddelerde istenen) tüm değişkenler int tipinde tanımlanacak.
4. Gömlek fiyat listesini görüntüleyin. Parasal değerleri gösterirken ondalık kısmında ve tam kısmında hassasiyet (**precision**) ayarlarına dikkat edin.
5. Parasal verileri saklarken ve bu veriler ile işlemler yaparken yuvarlamalardan dolayı bir takım sorunlar yaşarız. Örneğin virgülden sonra iki kısım gösterilir ve yuvarlamalar yapılır. Birim fiyatı üzerinden çok adetli gömlek alımını düşündüğümüzde veri kaybı ciddi boyutta olabilir.

İpucu

- Veri kaybı olmaması için parasal hesaplarda küsurat 0.5 ve üstüdeyse **yukarı**, değilse **aşağı** yuvarlayın.
- Parasal hesaplanan değeri int türden ve Cent(kuruş gibi) olarak saklayın. Bu durum, bu veriyi ekrana basarken yine \$(Dolar'a) çevireceğiniz anlamına gelir ve yuvarlama işlemini kolaylaştırır. Örneğin \$17.02 (17 dolar 2 cent) miktarını 1702 cent olarak saklamak gibi.

6. Seçilen gömlek türünden ve belirlenen adetten hesaplama yapıldığında vergisiz fiyatı ayrı bir değişkende (**sub-total**) saklanacak.
7. Vergiyi (**taxes**) hesaplarken **sub-total**'den faydalanarak hesaplama yapıp ayrı bir değişkende saklayacaksınız.
8. Ayrıca vergisiz fiyat ile verginin toplamını saklayan ayrı bir değişkene yine ihtiyacınız olacak.
9. Ve alışveriş ayrıntılarını gösteren (ekrana basan) kodları yazacaksınız.

Not

Parasal çıktılarda noktadan sonraki ve önceki hassasiyet(**precision**) değerine dikkat edin. (1.maddeyi dikkate alın)

Not

- Derlenen dosyaya Windows sistemlerde "**prg_w2p1.exe**", Linux ve MacOS sistemlerde "**prg_w2p1**" adı verilecek.
- Derleyici olarak gcc ve parametrelerden **-Wall** ve **-Werror** kullanılacak.

Part-1 Oputput Example (Not: **Sarı** alanlar kullanıcı veri girişini temsil eder)

```
Set Shirt Prices
=====
Enter the price for a SMALL shirt: $17.96
Enter the price for a MEDIUM shirt: $26.96
Enter the price for a LARGE shirt: $35.97

Shirt Store Price List
=====
SMALL   : $17.96
MEDIUM : $26.96
LARGE   : $35.97

Patty's shirt size is 'S'
Number of shirts Patty is buying: 8

Patty's shopping cart...
Contains : 8 shirts
Sub-total: $143.6800
Taxes    : $ 18.6800
Total    : $162.3600
```

Part-2 (%40)

1. Part-1'deki ana fonksiyonunuzu kopyalayın ve p2 klasöründe yer alan **w2p2.c** dosyasına yapıştırın. **w2p2.c** dosyasında var olan başlangıç kodlarının ezilmediğine dikkat edin.
2. Programın nasıl çalıştığına dair fikir edinmek için "**Part-2 Output Example**" kısmını dikkatlice inceleyin
3. Örnek çıktıda gösterildiği gibi çalışacak bir çözüm üretmek için kodunuzu gerektiği gibi değiştirmeniz gerekecektir.
4. Satış verilerini tablo formatında görüntülemek için biraz daha gelişmiş string formatlama özelliklerini araştırmanız gerekecek.

Şimdilik aşağıdaki print formatını kullanabilirsiniz. Bu formatı kendinize göre düzenleyin.

```
printf("Patty %-4c %5.21f %3d %9.41f %9.41f %9.41f\n", ...
```

Benzer şekilde tablonun toplam satırı da gelişmiş formatlama özellikleri içerir.

```
printf("%33.41f %9.41f %9.41f\n\n", ...
```

5. Tablo şeklindeki gösterimlerde özellikle toplam miktarları gösterirken veri kaybı olmamasına dikkat edin. Eğer miktarı int türünden bir değişkende cent olarak sakladıysanız (part-1 koşullarını hatırlayın) bu miktarı ekrana basarken \$(dolar)'a çevirdiğinizde veri kaybı olmadan ve yuvarlama kuralına uyarak yapmak için bölme operatörü (division operator "/") ve modulus "%" operatörünü kullanmanız gerekecektir.

Not

- İlk tablo ara toplamı temel alır ve vergileri içermez
 - İkinci tabloda toplam yer alır ve vergiler dahildir
 - Parayı tabloda verilen madeni para türlerine(**coin**) dönüştüreceğinizi unutmayın. Ve her dönüşen madeni para türünden sonra bir sonraki dönüşümde kalandan öncekini çıkarmayı unutmayın
6. Part-2'nin ilk tablosu sadece başlangıç **bakiyesini (balance)** gösterir(vergisiz toplam tutar). Sonraki satırlarda madeni para birimlerine göre eldeki parayı paylaştıracaksınız (hangi madeni para türünden kaç tane olacak gibi). Aşağıda ilk satır ve diğer satırlar için print formatı verilmiştir. Kullanabilirsiniz.

```
printf("%22.41f\n", ...  
printf("Toonies %3d %9.41f\n", ...
```

7. Her tablodan sonra, gömlek başına hesaplanan ortalama fiyat gösterilecektir.

Not

- Derlenen dosyaya Windows sistemlerde "**prg_w2p2.exe**", Linux ve MacOS sistemlerde "**prg_w2p2**" adı verilecek.
- Derleyici olarak gcc ve parametrelerinden **-Wall** ve **-Werror** kullanılacak.

Part-2 Output Example (Not: **Sarı** alanlar kullanıcı veri girişini temsil eder)

```
Set Shirt Prices
=====
Enter the price for a SMALL shirt: $17.96
Enter the price for a MEDIUM shirt: $26.96
Enter the price for a LARGE shirt: $35.97

Shirt Store Price List
=====
SMALL   : $17.96
MEDIUM : $26.96
LARGE   : $35.97

Patty's shirt size is 'S'
Number of shirts Patty is buying: 6

Tommy's shirt size is 'L'
Number of shirts Tommy is buying: 3

Sally's shirt size is 'M'
Number of shirts Sally is buying: 4

Customer Size Price Qty Sub-Total      Tax      Total
-----
Patty    S    17.96  6  107.7600  14.0100  121.7700
Sally    M    26.96  4  107.8400  14.0200  121.8600
Tommy    L    35.97  3  107.9100  14.0300  121.9400
-----
                        323.5100  42.0600  365.5700

Daily retail sales represented by coins
=====

Sales EXCLUDING tax
Coin      Qty      Balance
-----
                        323.5100
Toonies   161      1.5100
Loonies    1      0.5100
Quarters   2      0.0100
Dimes      0      0.0100
Nickels    0      0.0100
Pennies    1      0.0000

Average cost/shirt: $24.8854

Sales INCLUDING tax
Coin      Qty      Balance
-----
                        365.5700
Toonies   182      1.5700
Loonies    1      0.5700
Quarters   2      0.0700
Dimes      0      0.0700
Nickels    1      0.0200
Pennies    2      0.0000

Average cost/shirt: $28.1208
```