

1 Aşağıdaki C programı, kullanıcıdan "a" ve "b" değerlerini ondalık olarak girmesini isteyecek ve bu değerleri toplayarak sonucu 2 ondalık basamakla ekrana yazdıracak bir programdır:

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

int main() {
 float a, b;

 printf("Lütfen 'a' değerini giriniz: ");
 scanf("%f", &a);

 printf("Lütfen 'b' değerini giriniz: ");
 scanf("%f", &b);

 printf("Toplam: %0.2f", a + b);

 return 0;
}
```
```

Bu program kullanıcıdan sırasıyla "a" ve "b" değerlerini girmesini isteyecek, bu değerleri toplayacak ve sonucu 2 ondalık basamakla ekrana yazdıracaktır.

2 Bu C programı, ekrana "Merhaba" ve "Nasılsın?" yazdırmak için kullanılır. İşte programdaki her komutun anlamı:

1. `// Ekrana Merhaba Nasılsın? yazan program``: Bu bir yorum satırıdır ve programın açıklamasını içerir. Programın çalışmasına etki etmez, sadece açıklayıcıdır.
2. `#include <stdio.h>`: Bu satır, C programında kullanılan standart giriş/çıkış fonksiyonlarını ve veri türlerini içeren `stdio.h` başlık dosyasını programa dahil eder.
3. `int main()`: Bu satır, `main` adında bir işlevin başladığını belirtir. `int` veri türü, bu işlevin bir tamsayı değeri döndüreceğini gösterir.
4. `{`: Bu süslü parantez açma işareti, `main` işlevinin başladığını ve kod bloğunun başladığını işaretler.
5. `printf("Merhaba \n");`: Bu komut, "Merhaba" yazısını ekrana yazdırmak için `printf` fonksiyonunu kullanır. `\n` karakteri yeni bir satıra geçmeyi sağlar, böylece "Merhaba" ve "Nasılsın?" farklı satırlarda görünür.
6. `printf("Nasilsin?");`: Bu komut, "Nasilsin?" yazısını ekrana yazdırmak için `printf` fonksiyonunu kullanır.
7. `return 0;`: Bu satır, `main` işleminin başarıyla tamamlandığını ve işlevden çıkıldığını belirtir. Genellikle `0` değeri, programın başarıyla çalıştığını gösterir.
8. `}`: Bu süslü parantez kapatma işareti, `main` işlevinin ve kod bloğunun sona erdiğini belirtir.

Bu program, "Merhaba" ve "Nasılsın?" yazılarını ekrana yazdırarak çalışır ve başarıyla sonlandığında 0 değerini döndürür.

Özet

3 Yanlış değişken isimleri şunlardır:

- c. maas\$ (Değişken ismi özel karakter (\$) içeremez.)
- g. maas_\$ (Değişken ismi özel karakter (\$) içeremez.)

- j. le (Bu isim, dilin rezerve edilmiş bir kelimesidir ve değişken ismi olarak kullanılamaz.)
k. 23_maas (Değişken ismi rakamla başlayamaz.)

Diğer seçenekler geçerli değişken isimleri olabilir.

4 Tamsayı olarak tanımlanmış "toplam" değişkenini doğru bir şekilde ifade etmek için aşağıdaki C dilindeki deklarasyonu kullanabilirsiniz:

```
``c
int toplam = 0;
``
```

Bu deklarasyon, "toplam" adında bir tamsayı (int) değişkenini tanımlar ve ilk değeri 0 olarak atar.

5 İşte kullanıcıdan iki tamsayı alarak toplamını bulan bir C programı:

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

int main() {
    int sayi1, sayi2, toplam;

    printf("İlk tamsayıyı girin: ");
    scanf("%d", &sayi1);

    printf("İkinci tamsayıyı girin: ");
    scanf("%d", &sayi2);

    toplam = sayi1 + sayi2;

    printf("Toplam: %d\n", toplam);

    return 0;
}
``
```

Bu program kullanıcıdan iki tamsayı alır, bu sayıların toplamını hesaplar ve sonucu ekranda gösterir.

6 Doğru cevap:

a. Toplam:integer=0;

7 Aşağıdaki komutlar sırayla yürütüldüğünde, a, b ve c'nin son değerleri şu şekildedir:

- İlk olarak, a'ya 8 değeri atanır: $a = 8$
- Sonra, b'ye 15 değeri atanır: $b = 15$
- c ifadesi hesaplanır:
 - a/b ifadesi: $\frac{8}{15}$
 - b/a ifadesi: $\frac{15}{8}$
 - $b/a * 2$ ifadesi: $(\frac{15}{8}) * 2$
 - $a/b + b/a * 2$ ifadesi: $(\frac{8}{15}) + (\frac{15}{8}) * 2$
 - $c = 0.533333 + 3.75 = 4.283333$ (yaklaşık olarak)
- b değeri b/2 ile güncellenir: $b = \frac{15}{2} = 7.5$ (Ancak bu bir tam sayı olmalıdır. Bu nedenle genellikle bütün sayıya dönüştürülür. Burada, b'yi 7 olarak kabul edelim.)
- a değeri a + 3 ile güncellenir: $a = 8 + 3 = 11$

Sonuç olarak, $a = 11$, $b = 7$ ve $c \approx 4.283333$ olur (veya daha yaklaşık bir değer).

8 Verilen program parçasının çıktısı şu şekildedir:

```
""  
-27.38  
-28.4  
25  
30  
""
```

Açıklamalar:

1. İlk ``printf`` ifadesi, ``y`` değişkenini 6 haneli, 2 ondalık basamağı olan bir kayan noktalı sayı olarak formatlar ve -27.3789'u görüntüler.
2. İkinci ``printf`` ifadesi, ``--y`` ifadesini hesaplar (yani y değerini azaltır), ardından 7 haneli, 1 ondalık basamağı olan bir kayan noktalı sayı olarak formatlar ve -28.4'ü görüntüler.
3. Üçüncü ``printf`` ifadesi, ``a`` değişkenini 11'e böler ve sonucu tam sayı olarak formatlar, böylece $285 / 11$ işleminin sonucu olan 25'i görüntüler.
4. Son ``printf`` ifadesi, ``a`` değişkenine 5 ekler ($a+=5$), ardından 2 haneli bir tam sayı olarak formatlar ve sonucu 30'u görüntüler.