

BİLGİSAYAR PROGRAMLAMA

DÖNEM SONU SINAVI

Sınav Süresi:135 Dakika // Toplam 110 Puan (100 üzeri notlar 100'e yuvarlanır)

Adı ve Soyadı: (aşağı kısmı imzalamayı unutmayınız)

Ders Numarası: (Eğer ders numaranızı bilmiyorsanız, boş bırakınız)

Öğrenci Kimlik Numarası:

SINAV KURALLARI

- Sınav boyunca okul kimliğinizi fotoğrafınızın olduğu kısım üste gelecek şekilde masanın üzerinde bulundurun.
- Cep telefonlarınızı kapatın ve gözetmene teslim edin. Sınavda herhangi bir nedenle cep telefonu ile yakalanan öğrenci hakkında disiplin soruşturması açılabilir.
- Sınav esnasında gözetmenin izni olmaksızın sınav salonundan ayrılamazsınız. Sağlık raporuyla belgelenmiş sağlık sorunları haricince lavabo vb. taleplerle öğrencinin sınavı geçici süreli terk etmesine izin verilememektedir.
- Sınavın ilk yarım saati sınav salonundan ayrılamazsınız.
- Sınav esnasında kitap, defter, kâğıt veya herhangi bir elektronik cihaz kullanamazsınız.
- Sınav esnasında konuşmak, başkalarının kâğıdına bakmak, kâğıdını başkalarının görebileceği şekilde kullanmak yasaktır ve yakalandığı durumda kopya olarak değerlendirilebilir.
- Sınav başlamadan önce sıralarınızı mutlaka temizleyin. Sıralarında kopya sayılabilecek yazılar bulunan öğrenciler kopya çekmiş sayılabilir.

DÜRÜSTLÜK BEYANI

- Bu sınavda hiçbir şekilde kopya çekme girişiminde bulunmayacak ve hak ettiğim notu alacağım.
- Kopya çekme girişiminde bulunan herhangi bir kimse görürsem gözetmenimi çağıracağım ve durumu bildireceğim.

Yukarıda yazan kuralları okuduğumu ve dürüstlük beyanını kabul ettiğimi bildiririm.

İmza:

1) (5 puan + 5 Bonus Puan) Verilen örnek ekran çıktılarını sağlayacak şekilde istenen kodu yazınız. Bu soruda parçalı puanlama uygulanmaktadır. Ufak hatalar yapmanıza rağmen tam ya da tama yakın puan alabilirsiniz.

- **Mümkün olduğunca sade** komut/ifadeler kullanmanız beklenmektedir. Gereksiz komut/ifadeler **eklemeyiniz**.
- Kodun sadece sizden **istenen** kısmını yazınız.

Aşağıdaki örnek ekran çıktılarındaki gibi çalışan bir kod yazmanız istenmektedir.

- Kod çalışmaya başladığında kullanıcıdan bir giriş alır. Bu giriş, birden fazla sözcükten oluşabilir. Kullanıcı ardından enter'a basar. Kullanıcının her zaman küçük harflerle giriş yapacağını varsayınız. Kullanıcının geçersiz girişler yapmayacağını varsayınız.
- Eğer giriş tam olarak *"duzenli calismak"* ise kod *"Dogru bildiniz."* der ve "return value" olarak 0 dönerek sonlanır.
- Giriş *"duzenli calismak"* olmadığı sürece, kod *"Bilemediniz. Isin sirri: "* yazdırır ve kullanıcıdan tekrar değer ister.

```
Isin sirri: duzenli calismak
Dogru bildiniz.
-----
Process exited after 2.387 seconds with return value 0
```

ÖRNEK EKRAN ÇIKTISI 1

Kod çalıştıktan sonra kullanıcı *"duzenli calismak"* girişi yapıp enter'a basmıştır.

```
Isin sirri: duzensiz calismak
Bilemediniz. Isin sirri: son ana birakmak
Bilemediniz. Isin sirri: ?
Bilemediniz. Isin sirri: duzenli calismak
Dogru bildiniz.
-----
Process exited after 43.27 seconds with return value 0
```

ÖRNEK EKRAN ÇIKTISI 2

Kod çalıştıktan sonra kullanıcı sırasıyla *"duzensiz calismak"*, *"son ana birakmak"*, *"?"* ve *"duzenli calismak"* girişleri yapıp, her girişten sonra da enter'a basmıştır.

BONUS(5 puan): Kullanıcı kodun başında deneme hakkı girer (her zaman pozitif bir sayı gireceğini varsayınız) ve kod **ÖRNEK EKRAN ÇIKTISI 3**'te gözüktüğü şekilde kullanıcıya deneme hakları verir. Kullanıcı sürekli yanlış giriş yaparak deneme hakkının sonuna gelirse *"Elendiniz."* yazısı çıkar ve kod sonlanır. Bonus özelliğini içeren ayrı bir kod yazmanıza gerek yoktur. Sorunun tamamı için tek kod yazmanız yeterlidir. Kodunuzun ekran çıktısının, **ÖRNEK EKRAN ÇIKTISI 3**'teki girişler girildiğinde örnektekiyle aynı şekilde olmasına dikkat ediniz.

```
Deneme hakkı:2
Isin sirri: ruyada kod yazabilmek
Bilemediniz. 1 hakkınız kaldı.
Isin sirri: greyfurt suyu icmek
Elendiniz.
-----
Process exited after 71 seconds with return value 0
```

ÖRNEK EKRAN ÇIKTISI 3

Kod çalıştıktan sonra kullanıcı sırasıyla *"2"*, *"ruyada kod yazabilmek"*, *"greyfurt suyu icmek"* girişleri yapıp, her girişten sonra da enter'a basmıştır. Sonuncu yanlış giriş dışında, her yanlış girişten sonra kalan deneme hakkının belirtilmesine dikkat ediniz. Sonuncuda *"Bilemediniz. 0 hakkınız kaldı."* yazdırılmaz, *"Elendiniz."* yazısı çıkar ve kod sonlanır.

ÖNEMLİ UYARILAR:

- Sorunun başında da belirtildiği üzere **sade komutlar/ifadeler** kullanmanız beklenmektedir. Örneğin, 1 printf ile yapılabilecek kısmı 10 printf ile yaparsanız kodunuz doğru çalışsa bile puan kaybı yaşayabilirsiniz.
- Kod içerisinde **boşluk** bırakmanın kritik olduğu kısımlardaki boşluklarınız **açık olmalı**. Lütfen bu tür boşlukları okla göstererek *"burada boşluk var"* şeklinde belirtiniz.
- Kodlarda derleme imkanı olmadığı için rahatlıkla ; **gibi detaylar** unutulabilmektedir. Ayrıca, **kötü/hızlı yazımdan** kaynaklı (t gibi gözüken f harfleri, l gibi gözüken i'ler, . gibi gözüken ,ler...) hatalar da olabilmektedir. Soruyu çözmeyi bitirdiğinizde ya da sınavın sonunda **soru metnini, kodunuzun çalışmasını ve kodun okunurluğunu** dikkatlice ve özenle **kontrol ediniz**.



CEVABINIZI EN ŞİRİN YAZINIZLA VE DİKKATLİCE
BİR SONRAKİ ŞİRİN SAYFAYA YAZINIZ.

Lütfen karakterlerinizin (boşluk karakteri dâhil) **açık, net, eksiksiz** olduğunu kontrol ediniz.

`printf("Ornek);` komutunda " eksik olması gibi,

`<=` yerine `≤` yazılması gibi durumların cevabınızda **olmamasına** özen gösteriniz.

Ancak harflerin yazımı ile ilgili ufak yazım hataları (f yerine F yazmak gibi) , soruda özellikle ölçülenler arasında değilse, puan kaybına yol açmaz.

2) (Her boşluk 1 puan - en iyi 15 cevap değerlendirmeye alınır - toplam 15 puan)

Usta Şirin, ŞirinApp isimli bir mesajlaşma yazılımı icat etmek üzeredir. İcadının ilk denemelerini Şirin Baba ile yapacaktır. Ancak ŞirinApp'in kodlamasındaki bir mantık hatası yüzünden, Gargamel de yazışmaları takip edebilmekte ve hatta kendisi de mesajlar gönderebilmektedir. Bunu düşünerek, aşağıdaki mesajların kimden geldiğini ve mesaj içeriklerindeki boşlukları dikkatlice doldurunuz. Şirin Baba'nın her zaman doğru cevaplar verdiğini, Gargamel'in ise her zaman yanlış cevaplar verdiğini varsayınız.

**BU SORULARIN HER BİRİNDEN SADECE TAM YA DA 0 PUAN ALIRSINIZ.**

y yerine Y yazmak gibi yazım hataları, soruda özellikle ölçülenler arasında değilse, puan kaybına yol açmaz.

Usta Şirin: Şirin Baba, Şirin Soru Bankası'na çalışırken fark ettim. Yazışmalarımız oraya sızmış, bilgin olsun.

Şirin Baba: Şirin App'te küresel değişken kullanmışsındır. Biz yine başlayalım yazışmaya.

Usta Şirin: O zamanlar acemiydim. Tamam, başlıyorum öyleyse... Sana sorularım var Şirin Baba.

Şirin Baba: Dönem boyu düzenli çalıştıysan ne mutlu sana Usta Şirinim. Sor bakalım.

Usta Şirin: *int* türünde *x* değişkenim var. Kodumda "*x* dört basamaklı ise" koşul ifadesine ihtiyacım var. Aradığım ifade şöyle midir:

$$if(1000 \leq x < 10000)$$

_____ : Kullandığın " $\leq$ " yerine C dilinde kullanılan " $\leq$ " ifadesini kullanırsan işini görür.

Usta Şirin: Yapılar konusundaki $\rightarrow$ işaretini bir türlü anlamadım, tam olarak nedir?

_____ : *ptr* isimli bir işaretçin olsun ve bu işaretçi bir yapıya işaret ediyor olsun. (**ptr*). yazmak yerine *ptr* $\rightarrow$ yazabiliyorsun.

Usta Şirin: *union* ne işe yarar?

_____ : *union(yapi1, yapi2)* şeklinde kullandığında iki yapının birleşimini alıp *yapi1*'e kaydeder.

Usta Şirin: *int x*; şeklinde tanımlı değişkenimi uygun şekilde tanımlı *fonk1* fonksiyonuna *fonk1(x)* şeklinde gönderince, fonksiyon onun üzerinde kalıcı değişiklik **yapamıyor**. Ama *int x[1]*; şeklinde tanımlı değişkenimi uygun şekilde tanımlı *fonk2* fonksiyonuna *fonk2(x)* şeklinde gönderince, fonksiyon onun üzerinde kalıcı değişiklik **yapabiliyor**. Bunun nedeni nedir?

_____ : Diziler fonksiyona adres ile giderler. Bu nedenle *fonk2* fonksiyonu, değişkenin RAM'dekini yerini bilebilir ve değerini kalıcı olarak değiştirebilir.

Usta Şirin: *int x[1];* şeklinde tanımlı değişkenin ismi adres mi oluyor?

Şirin Baba: Evet, C dilinde bu şekilde bir dizi tanımlarsan, çağırma aşamasında *fonk2()* süslüleri içerisinde *x* ifadesini yazmak & _____ yazmaya denktir.

Usta Şirin: Şöyle bir kod gördüm(aşağıda paylaşıyorum). Bu kod hata veya uyarı vermeksizin çalışabilir mi?

```
int main()
{
    int x = 56;
    return 0;
}
```

_____: Hayır, bu kod çalışsa bile derleyici uyarı verecektir. Çünkü başına *stdio.h* kütüphanesi eklenmemiştir.

Usta Şirin: *int main(int argc, char* argv[])* ifadesindeki parantez içi ifadeler nedir?

_____: Kod komut ekranından çalıştırıldığında koda parametre yollayabilmene imkan verir.

Usta Şirin: *FILE* dosya;* diye bir değişken tanımladım ve *fopen* ile açtığım dosyadan döngü içerisinde *fscanf* ile değer çekiyorum. Ancak bu dosyada değer kalıp kalmadığını nasıl anlarım? Döngüden ne zaman çıkmam gerektiğine onunla karar vereceğim.

Şirin Baba: Koduna ekleyeceğin _____(*dosya*) ifadesi 0 dönüyorsa dosyanın sonuna henüz gelinmemiştir. Onu kontrol ederek, kodunda döngüden çıkmaya karar verebilirsin.

Usta Şirin: *typedef* ile şöyle bir yapı tanımladım.

```
typedef struct
{
    int x;
}building;
```

Şimdi *main*'in içerisinde

```
building.x = 7;
printf("%lf", building.x);
```

yazdığımda kod hata veriyor. Bunun nedeni nedir?

_____: *building.x* ifadesi *int* türünde olduğu için *%lf* değil de *%d* yazarsan sorun çözülür.

Usta Şirin: C dili nesne tabanlı bir dil midir?

Şirin Baba: Hayır, değildir ancak içindeki _____ kavramı 'nesne'ye çok benzemektedir.

Usta Şirin: Kullanıcıdan üç tane isim alacağım, bu isimlerin her biri sadece "Sirine", "Gargamel" ya da "Guclu" olabilecek. Şirin kullanıcı bunların haricinde bir isim girişi yapmayacak. Bu 3 ismi içinde tutabilecek *sirinler* isimli tek bir değişkenim olsun istiyorum. Bu değişkeni nasıl tanımlarım?

Şirin Baba: *char sirinler* _____; şeklinde tanımlarsan işini görecektir.

Usta Şirin: *int sirin[3][10];* şeklinde tanımlı *sirin* dizisi kaç tane *int* değerini içinde tutabilir?

Şirin Baba: _____

Usta Şirin: *int *y = &x;* ifadesi ne iş yapar?

_____ : *int *y;* ve *y = &x;* ifadeleri peş peşe geldiğinde ne iş yapıyorsa, senin yazdığın ifade de o işi yapar.

Usta Şirin: Kod her çalıştırıldığında farklı bir rakam yazdırsın istiyorum.

_____ : En başa gerekli kütüphaneleri ekler ve *main*'in içinde de *printf("%d", rand()%10);* şeklinde kullanırsan her çalıştırdığında farklı bir rakam yazdırır. Ancak ara sıra tesadüfen aynı rakam da çıkabilir.

Usta Şirin: *malloc* fonksiyonunun kullanımı nasıldır?

_____ : Bellekten yer ayırt edip ilk baytının adresini sana dönen bir fonksiyon. 100'lük bir *int* dizisi açacaksan, *malloc(400)* yazıp döndüğü değeri bir *int* işaretçisine kaydedebilirsin.

Usta Şirin: Azman programlama öğrenmeye başlamış. Geçen gün ormana mesaj bırakmış, "*strcat ne işe yarıyor? Kedi dizgisi mi üretiyor? Biri bana açıklasın, miyav*" şeklinde. Yardımcı olabilir miyiz?

_____ : Tabii ki Şirinim, *strcat(dizgi1, "Azman");* şeklinde kullanarak "*Azman*" ifadesinin peşine *dizgi1*'in içeriğini ekletebiliyorsun.

Gargamel: Bu arada, şirinim, bunu Azman'a doğrudan iletirsen çok sevinir. Selamımı söylersin.

3) (Her boşluk 2 puan - en iyi 10 cevap değerlendirmeye alınır - toplam 20 puan)

Aşağıda bazı kodlar ve ekran çıktıları verilmiştir. Boşlukları doldurunuz.

• **BU BOŞLUKLARIN HER BİRİNDEN SADECE, DOĞRU CEVABI TAM OLARAK VERMENİZ DURUMUNDA PUAN ALABİLİRSİNİZ. BİR BOŞLUK YANLIŞSA SADECE O BOŞLUKTAN PUAN KAYBEDERSİNİZ.**

- / ve \ farkı, ' ve " farkı, ≤ yerine <= gibi detaylar da dâhil olmak üzere cevabınızın tam olarak doğruluğuna emin olunuz. r yerine R yazmak gibi durumlar, soruda özellikle ölçülenler arasında değilse, puan kaybına yol açmaz.
- Mümkün olduğunca sade komut/ifadeler kullanmanız beklenmektedir. Gereksiz komut/ifadeler eklemeyiniz.
- Sadece boşluklara yazdıklarınız değerlendirmeye alınır. Boşluklara cevap dışında hiçbir şey yazmayınız.
- Kod çalıştırıldığında derleyici herhangi bir uyarı ya da hata vermeyecek şekilde boşlukları doldurunuz.

KOD	ÖRNEK EKRAN ÇIKTISI
Kullanıcıdan değerler alarak, örnek ekran çıktılarındaki gibi cümle yazdıran kod <pre>#include <stdio.h> typedef struct { int gun; char ay[100]; int yil; }date; void duzelt(date*); void yazdir(date); int main() { _____; printf("Konser ne zaman?\n"); printf("Gun:"); scanf("%d", &tarih.yil); printf("Ay:"); scanf("%s", tarih._____); printf("Yil:"); scanf("%d", &tarih.gun); duzelt(_____); yazdir(tarih); return 0; } void duzelt(date* ptr) { int temp; _____; ptr->gun = ptr->yil; ptr->yil = temp; } void yazdir(date tarih) { printf("%d yilinin %d %s gunu bulusmak uzere.", tarih.yil, tarih.gun, tarih.ay); }</pre>	Konser ne zaman? Gun:28 Ay:Kasim Yil:2018 2018 yilinin 28 Kasim gunu bulusmak uzere. ----- Process exited after 4.995 seconds with return value 0 Devam etmek için bir tuşa basın . . . Örnek ekran çıktısı -1 Kod çalıştırıldıktan sonra sırasıyla "28", "Kasim", "2018" girişleri yapılmış ve her girişten sonra enter'a basılmıştır. Konser ne zaman? Gun:12 Ay:Temmuz Yil:2019 2019 yilinin 12 Temmuz gunu bulusmak uzere. ----- Process exited after 23.46 seconds with return value 0 Devam etmek için bir tuşa basın . . . Örnek ekran çıktısı -2 Kod çalıştırıldıktan sonra sırasıyla "12", "Temmuz", "2019" girişleri yapılmış ve her girişten sonra enter'a basılmıştır. Gereksiz komut/ifadeler eklemeyiniz. Bu tür durumlarda kodunuz çıktı açısından isteneni yapıyor olsa bile puan alamayabilirsiniz. Kod çalıştırıldığında derleyici herhangi bir uyarı ya da hata vermemelidir.

KOD - İstatistikte ortalama(int. average, mean) gibi **ortanca**(ing. median) değer de kullanılır. Bir sayı dizisinin **ortancası**, sayılar artan (ya da azalan) sırada sıralandıktan sonra ortada kalan değerdir. 1 2 4 dizisinin **ortancası** 2'dir. Dizi eleman sayısı çift sayıda olursa, ortada iki eleman kalır. Bu durumda **ortanca**, bu iki elemanın ortalamasıdır. 1 2 3 5 dizisinin **ortanca** değeri (2+3) değerinin yarısı olan 2,5'tur.

Kullanıcının girdiği doğal sayı değerleriyle oluşan dizinin **ortancasını** veren kod. Kullanıcı, 10'dan az sayıda değer girdiğinde değerlerin sonuna -1 ekler ve enter'a basar. Kullanıcı, 10 ya da daha fazla değer girdiğinde, kod sadece ilk 10 değeri değerlendirmeye almak üzere tasarlıdır; bu durumlarda kullanıcının sayıların sonuna -1 eklemesine gerek yoktur.

ÖRNEK EKRAN ÇIKTISI

```
#include <stdio.h>
void siralat(int dizi[], int i);
int main()
{
    int dizi[10]= {}, i, j;
    printf("Sayılar:");
    for(i=0; i<10; i++)
    {
        scanf("%d", &dizi[i]);

        if(dizi[i]==-1) _____;
    }

    siralat(_____);
    printf("Kucukten buyuge:");

    for(j=0; j<_____ ; j++)
        printf("%d ", dizi[j]);

    printf("\nOrtanca: ");
    printf("%.1lf", (dizi[(i-1)/2]+_____/2.0 ));
    return 0;
}

void siralat(int dizi[], int i)
{
    int a,b, bittiMi=0;
    for(a=1; bittiMi==0; a++)
    {
        bittiMi=141;
        for(b=0; b!=i-a; b++)
        {
            if(_____)
            {
                _____;
                dizi[b] = dizi[b+1];
                dizi[b+1] = temp;
                _____;
            }
        }
    }
}
```

Sayılar:5 6 985414 -1
Kucukten buyuge:5 6 985414
Ortanca: 6.0

Örnek ekran çıktısı -1
Kod çalıştıktan sonra kullanıcı 3 değer girmiş, peşine -1 eklemiş ve enter'a basmıştır. Ortanca değer olan 6 değeri yazdırılmıştır.

Sayılar:9 8 7 6 5 4 3 2 1 0 50 27
Kucukten buyuge:0 1 2 3 4 5 6 7 8 9
Ortanca: 4.5

Örnek ekran çıktısı -2
Kod çalıştıktan sonra kullanıcı 12 değer girmiştir. Sonuna -1 eklemesine gerek kalmamıştır. Çünkü kod, 10'dan fazla değer girilirse ilk 10 tanesini dikkate almaktadır. Girilen ilk 10 eleman küçükten büyüğe sıralandığında ortada kalan değerler 4 ve 5 olduğu için, bunların ortalaması olan 4,5 ortanca değer olarak yazdırılmıştır.

Sayılar:3 5 5 5 6 7 8 9 -1
Kucukten buyuge:3 5 5 5 6 7 8 9
Ortanca: 5.5

Örnek ekran çıktısı -3
Kod çalıştıktan sonra kullanıcı 8 değer girmiş, peşine -1 eklemiş ve enter'a basmıştır. Girilen elemanlar küçükten büyüğe sıralandığında ortada kalan değerler 5 ve 6 olduğu için, bunların ortalaması olan 5,5 ortanca değer olarak yazdırılmıştır.

Sayılar:50 1000 80 9000 -1
Kucukten buyuge:50 80 1000 9000
Ortanca: 540.0

Örnek ekran çıktısı -4
Kod çalıştıktan sonra kullanıcı 4 değer girmiş, peşine -1 eklemiş ve enter'a basmıştır. Dizilimden sonra ortada kalan 80 ve 1000'in ortalaması olan 540 değeri ortanca olarak yazdırılmıştır.

Gereksiz komut/ifadeler eklemeyiniz. Bu tür durumlarda kodunuz çıktı açısından isteneni yapıyor olsa bile puan alamayabilirsiniz.

4) (Her çıktı 3 puan - en iyi 20 cevap değerlendirmeye alınır - toplam 60 puan)

Aşağıdaki kodların altındaki kutulara ekran çıktılarını çok dikkatlice yazınız.

• **BU SORULARDAN SADECE, DOĞRU CEVABI TAM OLARAK VERMENİZ DURUMUNDA PUAN ALABİLİRSİNİZ.**

Örneğin, cevap 14.3 ise 14.30 yazan öğrenci o sorudan puan alamaz! Ancak .(nokta) yerine ,(virgül) koymak ya da C yerine c yazmak gibi durumlar, soruda özellikle ölçülenler arasında değilse , puan kaybına yol açmaz.

• Kutular içine ekran çıktısını içeren cevabınız dışında hiçbir şey yazmayınız.

• Kutulardaki her hücre ekrandaki bir karakteri temsil eder. Bir hücreye birden fazla karakter yazmayınız.

• Cevabınızın kutuya sığmadığı durumlar olursa, ilgili kısma ek hücreler çizerek cevabınızı belirtebilirsiniz.

• Bazı sorularda çıktının bir kısmı cevap kutusunun içine yazdırılmış olabilir, bu tip sorularda çıktının geriye kalan kısmını siz tamamlayınız. Bu sorularda kullanıcının klavyeden yaptığı giriş değerleri çıktı kutuları içinde *koyu italik* olarak gösterilir.

• (Mantık hatası haricinde) Hata içerdiğini düşündüğünüz sorularda ne tür bir hata olduğunu aşağıdaki seçeneklerden seçerek cevap kutusunun içine (örn. "HATA1" şeklinde) yazınız VE nedenini cevap kutusunun altına ya da soruya yakın başka boş bir alana (örn. "Çünkü koddaki ... eksik." şeklinde) belirtiniz. YUKARIDA BELİRTİLDİĞİ ÜZERE, HATA TESPİTİ DURUMUNDA **HEM HATA NUMARASINI HEM DE NEDENİNİ** YAZMANIZ İSTENMEKTEDİR. PUAN ALABİLMENİZ İÇİN İKİSİNDEN BİRİ EKSİK OLMAMALIDIR.

HATA1:Kod hiç derlenemeyecektir.

HATA2:Kod derlenecek ama çalışması esnasında hata verecektir.

HATA3:Kod derlenecek ama çalışması esnasında sonsuz döngüye girecektir.

Yukarıdaki ifadelerle uymayan hatalarda (örneğin mantık hatası içerdiğini düşündüğünüz sorularda) hatasız sorularda olduğu gibi sadece ekran çıktısını yazınız, HATA# vs. yazmayınız.

```
#include <stdio.h>

void ressam(int x);

int main()
{
    ressam(1);
    return 0;
}

void ressam(int x)
{
    if(x==5)
        return;
    ressam(x+1);
    printf("%d",x);
    return;
}
```


```
#include <stdio.h>

int main()
{
    int c=4;
    int* d;
    d=&c;
    printf("X:");
    scanf("%d", d);
    printf("Y:");
    printf("%d", c**d+*d);
    return 0;
}
```

X	:	9							

/*Kod çalıştıktan sonra kullanıcı "9"(çift tımlar dahil değil) yazmış ve enter'a basmıştır. enter nedeniyle kodun bir aşağı satıra inerek devam etmesi beklenir. Çıktının bir kısmı verilmiştir, geriye kalanını sizin tamamlamanız istenmektedir.*/

```
#include <stdio.h>

int main()
{
    int i,j,k=0;
    for(i=0; i<3; i++)
    {
        for(j=0;j<k;j++)
            printf("w");
        for(j=i; j<i+2;j++)
            printf("%d",j);
        printf("\n");
        k++;
    }
    return 0;
}
```


```
#include <stdio.h>

enum booleanTuru {true, false};

int main()
{
    enum booleanTuru p = !!!true;
    printf("%d", p);
    p = true && (false || !true);
    printf("%d", p);
    return 0;
}
```


```
#include <stdio.h>

int schtroumpf(int puan);

int main()
{
    int puan = 0;
    do
    {
        schtroumpf(puan);
    }while(puan!=0);
    printf("SCORE%d",puan);
    return 0;
}

int schtroumpf(int x)
{
    x+=100;
    return x;
}
```


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

int main()
{
    // a'nin ASCII'si 97'dir.
    char guclu[8]= {'a','s','c','i'};
    strcpy(guclu,"98");
    printf("%s", guclu);
    return 0;
}
```


```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
void tembel(int []);

int main()
{
    srand(time(NULL));
    int dizi[10]={};
    tembel(dizi);
    printf("%d+2", dizi[0]+2); //dikkat!
    if(dizi[5]<30)
        printf("A");
    else if(dizi[5]<=10)
        printf("B");
    else
        printf("C");
    return 0;
}

void tembel(int dizi[])
{
    int i=1;
    while(i!=10)
    {
        dizi[i]=(rand()+80)%10+1;
        i++;
    }
}
```


```
#include <stdio.h>
#include <string.h>
int main()
{
    char name[100] = "le schroumpf grognon";
    name[2]='\0';
    if(strcmp(name,"le"))
        printf("%s", name);
    else
        printf("%s", name+1);
    return 0;
}
```


```
#include <stdio.h>
double somurtkan(double);
int main()
{
    double suslu[3]={3.14,2.22,1.41};
    somurtkan(suslu[0]);
    suslu[1] = somurtkan(suslu[1]);
    printf("%.2lf\n",suslu[0]+suslu[1]);
    return 0;
}

double somurtkan(double x)
{
    x = 7.20;
    printf("%.2lf\n",x);
    return x;
}
```


```
#include <stdio.h>
double somurtkan(double*);
int main()
{
    double suslu[3]={3.14,2.22,1.41};
    somurtkan(&suslu[0]);
    suslu[1] = somurtkan(&suslu[1]);
    printf("%.2lf\n",suslu[0]+suslu[1]);
    return 0;
}

double somurtkan(double* ptr)
{
    *ptr = 3.00;
    printf("%.2lf\n", *ptr );
    return (*ptr)+1.0;
}
```



```
#include <stdio.h>
void ressam(int x);
int main(){
    ressam(3);
    return 0;
}
void ressam(int x){
    int i;
    if(x==1)
    {
        printf("***");
        printf(" "); //1 bosluk
        printf("***");
        return;
    }
    for(i=1; i<=x; i++)
        printf("***");
    printf("\n");
    ressam(x-2);
    printf("\n");
    for(i=x; i>=1; i--)
        printf("***");
    return;
}
```


```
#include <stdio.h>
#include <string.h>
typedef union{ //dikkat! struct degil
    char sirinlsim[4];
    int boy;
}smurf;
int main(){
    smurf gozluklu;
    gozluklu.boy=141;
    printf("%d",gozluklu.boy);
    strcpy(gozluklu.sirinlsim, "ABC");
    if(gozluklu.boy+1 == 142)
        printf("%d\n", gozluklu.boy+101);
    printf("%d", strlen(gozluklu.sirinlsim));
    return 0;
}
```


```
#include <stdio.h>
int t(int[], int);
int main(int argc, char* argv[])
{
    int dizi[4] = {1,4,7};
    printf("%d", t(dizi,argc));
    return 0;
}
int t(int dizi[], int x)
{
    if(x<=2) return dizi[x];
    return t(dizi,x-1)+t(dizi,x-2)+t(dizi,x-3);
}
```

Komut Ekranı

gcc kod.c -o kod.exe

kod azman ve gargamel

(Kullanıcı komut ekranına, verilen iki satır komutu girmiş ve her komuttan sonra enter'a basmıştır.)

```
#include <stdio.h>
#include <conio.h>
void degerlendir(int);
int main()
{
    printf("Sayi:");
    int a1 = getch();
    //getche kullanilmadi!
    degerlendir(a1);
    printf("C");
    return 0;
}
void degerlendir(int x)
{
    if(x!=7)
        printf("D");
    else
        printf("E");
}
```

S	a	y	i	:					

/*Kod çalıştıktan sonra kullanıcı "7" tuşuna basmıştır. Çıktının bir kısmı verilmiştir, geriye kalanını sizin tamamlamanız istenmektedir.*/

```

#include <stdio.h>
#define X 3
#define Y 6
int main()
{
    int dizi[2][X][Y]={ { { 1,0,0,6,8,3},{2324,0,8},{0,2,3,11,2} } , {} };
    int i,j;
    for(i=0; i<X; i++)
    {
        for(j=0;j<Y;j++)
            dizi[1][2-i][j]=dizi[0][i][j];
    }

    for(i=0; i<X; i++)
    {
        for(j=0;j<Y;j++)
        {
            if(dizi[1][i][j]!=0)
            {
                printf("%d",dizi[1][i][j]);
                printf(" "); //1 boşluk
            }
        }
        printf("\n");
    }
    //cevabinizi iyice kontrol ediniz.
    return 0;
}

```


```

#include <stdio.h>
int main(int argc, char* argv[])
{
    printf("%d",argc); // %dden sonra \n yok!
    printf("%s", (argc%2==1) ? argv[2]:argv[1]);
    return 0;
}

```

Komut Ekranı

gcc kod.c -o kod.exe
kod.exe sirinler tatilde

(Kullanıcı komut ekranına, verilen iki satır komutu girmiş ve her komuttan sonra enter'a basmıştır.)

```
#include <stdio.h>
int main(int argc, char* argv[])
{
    int i;
    for(i=0; i<argc-1; i++)
    {
        switch(i)
        {
            case 1:
                printf("%s\n", argv[2]);
                break;
            case 2:
                printf("%s\n", argv[0]);
                break;
            default:
                printf("%s\n", argv[1]);
        }
    }
    return 0;
}
```

Komut Ekranı

gcc kod.c -o kod.exe									
kod.exe sirinler tatilde 2									

(Kullanıcı komut ekranına, verilen iki satır komutu girmiş ve her komuttan sonra enter'a basmıştır.)

```
#include <stdio.h>
typedef struct
{
    char sirinlsim[100];
    double boy;
}smurf;
int main()
{
    smurf asci;
    asci = (smurf){.boy=0.000012, .sirinlsim = "Asci Sirin"};
    printf("%.2lf\n", asci.boy);
    printf("%s\n", asci.sirinlsim);
    return 0;
}
```


GERİBİLDİRİM SORULARI (EKSTRA 5 PUAN)

4 PUAN: Eğer aşağıdaki maddeleri cevaplarsanız & 1 PUAN: Girilen sayı alınan nota yakın olursa

1. Aşağıya bir sayı yazınız. Yazdığınız sayının sınavdan alacağınız nota maksimum 5,0 puan yakın olması durumunda ekstra 1 puan alırsınız. (Bu kısımdan alınabilecek 1 puanı dâhil etmeden sayı girişinde bulununuz. Mantıklı olan aşağıdaki boşluğu [5,0-95,0) aralığında bir değerle doldurmaktır.)

Sayı = _____ ± 5,0

2. Bu dersten almayı beklediğiniz harf notu nedir? AA BA BB CB CC DC DD FF

3. Bu sınava yeterince çalışabildiğinizi düşünüyor musunuz?

4. Kuramsal ve uygulamalı dersler hakkında neler düşünüyorsunuz?

5. Ödevler ve kısa sınavlar hakkında neler düşünüyorsunuz?

6. Geri bildirimde bulunmak istediğiniz diğer konular nelerdir?

BİLGİSAYAR PROGRAMLAMA

DÖNEM SONU SINAVI

Sınav Süresi:135 Dakika // Toplam 110 Puan (100 üzeri notlar 100'e yuvarlanır)

Adı ve Soyadı:

(aşağı kısmı imzalamayı unutmayınız)

Ders Numarası:

(Eğer ders numaranızı bilmiyorsanız, boş bırakınız)

Öğrenci Kimlik Numarası:

SINAV KURALLARI

- Sınav boyunca okul kimliğinizi fotoğrafınızın olduğu kısım üste gelecek şekilde masanın üzerinde bulundurun.
- Cep telefonlarınızı kapatın ve gözetmene teslim edin. Sınavda herhangi bir nedenle cep telefonu ile yakalanan öğrenci hakkında disiplin soruşturması açılabilir.
- Sınav esnasında gözetmenin izni olmaksızın sınav salonundan ayrılamazsınız. Sağlık raporuyla belgelenmiş sağlık sorunları haricince lavabo vb. taleplerle öğrencinin sınavı geçici süreli terk etmesine izin verilememektedir.
- Sınavın ilk yarım saati sınav salonundan ayrılamazsınız.
- Sınav esnasında kitap, defter, kâğıt veya herhangi bir elektronik cihaz kullanamazsınız.
- Sınav esnasında konuşmak, başkalarının kâğıdına bakmak, kâğıdını başkalarının görebileceği şekilde kullanmak yasaktır ve yakalandığı durumda kopya olarak değerlendirilebilir.
- Sınav başlamadan önce sıralarınızı mutlaka temizleyin. Sıralarında kopya sayılabilecek yazılar bulunan öğrenciler kopya çekmiş sayılabilir.

DÜRÜSTLÜK BEYANI

- 1) Bu sınavda hiçbir şekilde kopya çekme girişiminde bulunmayacak ve hak ettiğim notu alacağım.
- 2) Kopya çekme girişiminde bulunan herhangi bir kimse görürsem gözetmenimi çağıracağım ve durumu bildireceğim.

Yukarıda yazan kuralları okuduğumu ve dürüstlük beyanını kabul ettiğimi bildiririm.

İmza:

1) (5 puan + 5 Bonus Puan) Verilen örnek ekran çıktıları sağlayacak şekilde istenen kodu yazınız. Bu soruda parçalı puanlama uygulanmaktadır. Ufak hatalar yapmanıza rağmen tam ya da tama yakın puan alabilirsiniz.

- **Mümkün olduğunca sade** komut/ifadeler kullanmanız beklenmektedir. Gereksiz komut/ifadeler **eklemeyiniz**.
- Kodun sadece sizden **istenen** kısmını yazınız.

Aşağıdaki örnek ekran çıktılarındaki gibi çalışan bir kod yazmanız istenmektedir.

- Kod çalışmaya başladığında kullanıcıdan bir giriş alır. Bu giriş, birden fazla sözcükten oluşabilir. Kullanıcı ardından enter'a basar. Kullanıcının her zaman küçük harflerle giriş yapacağını varsayınız. Kullanıcının geçersiz girişler yapmayacağını varsayınız.
- Eğer giriş tam olarak "duzenli calismak" ise kod "Dogru bildiniz." der ve "return value" olarak 0 dönerek sonlanır.
- Giriş "duzenli calismak" olmadığı sürece, kod "Bilemediniz. Isin sirri: " yazdırır ve kullanıcıdan tekrar değer ister.

```
Isin sirri: duzenli calismak
Dogru bildiniz.
-----
Process exited after 2.387 seconds with return value 0
```

ÖRNEK EKRAN ÇIKTISI 1

Kod çalıştıktan sonra kullanıcı "duzenli calismak" girişi yapıp enter'a basmıştır.

```
Isin sirri: duzensiz calismak
Bilemediniz. Isin sirri: son ana birakmak
Bilemediniz. Isin sirri: ?
Bilemediniz. Isin sirri: duzenli calismak
Dogru bildiniz.
-----
Process exited after 43.27 seconds with return value 0
```

ÖRNEK EKRAN ÇIKTISI 2

Kod çalıştıktan sonra kullanıcı sırasıyla "duzensiz calismak", "son ana birakmak", "?" ve "duzenli calismak" girişleri yapıp, her girişten sonra da enter'a basmıştır.

BONUS(5 puan): Kullanıcı kodun başında deneme hakkı girer (her zaman pozitif bir sayı gireceğini varsayınız) ve kod ÖRNEK EKRAN ÇIKTISI 3'te gözüktüğü şekilde kullanıcıya deneme hakları verir. Kullanıcı sürekli yanlış giriş yaparak deneme hakkının sonuna gelirse "Elendiniz." yazısı çıkar ve kod sonlanır. Bonus özelliğini içeren ayrı bir kod yazmanıza gerek yoktur. Sorunun tamamı için tek kod yazmanız yeterlidir. Kodunuzun ekran çıktısının, ÖRNEK EKRAN ÇIKTISI 3'teki girişler girildiğinde örnektekiyle aynı şekilde olmasına dikkat ediniz.

```
Deneme hakkı:2
Isin sirri: ruyada kod yazabilmek
Bilemediniz. 1 hakkınız kaldı.
Isin sirri: greyfurt suyu icmek
Elendiniz.
-----
Process exited after 71 seconds with return value 0
```

ÖRNEK EKRAN ÇIKTISI 3

Kod çalıştıktan sonra kullanıcı sırasıyla "2", "ruyada kod yazabilmek", "greyfurt suyu icmek" girişleri yapıp, her girişten sonra da enter'a basmıştır. Sonuncu yanlış giriş dışında, her yanlış girişten sonra kalan deneme hakkının belirtilmesine dikkat ediniz. Sonuncuda "Bilemediniz. 0 hakkınız kaldı." yazdırılmaz, "Elendiniz." yazısı çıkar ve kod sonlanır.

ÖNEMLİ UYARILAR:

- Sorunun başında da belirtildiği üzere **sade komutlar/ifadeler** kullanmanız beklenmektedir. Örneğin, 1 printf ile yapılabilecek kısmı 10 printf ile yaparsanız kodunuz doğru çalışsa bile puan kaybı yaşayabilirsiniz.
- Kod içerisinde **boşluk** bırakmanın kritik olduğu kısımlardaki boşluklarınız **açık olmalı**. Lütfen bu tür boşlukları okla göstererek "burada boşluk var" şeklinde belirtiniz.
- Kodlarda derleme imkanı olmadığı için rahatlıkla ; gibi detaylar unutulabilmektedir. Ayrıca, kötü/hızlı yazımdan kaynaklı (t gibi gözüken f harfleri, l gibi gözüken i'ler, . gibi gözüken 'ler...) hatalar da olabilmektedir. Soruyu çözmeyi bitirdiğinizde ya da sınavın sonunda **soru metnini, kodunuzun çalışmasını ve kodun okunurluğunu** dikkatlice ve özenle **kontrol ediniz**.



CEVABINIZI EN ŞİRİN YAZINIZLA VE DİKKATLİCE
BİR SONRAKİ ŞİRİN SAYFAYA YAZINIZ.

// BİL141 DERSİ SINAVINDAKİ 1.SORU KODUM

```

#include <stdio.h> //strcmp istisnai olarak string.h gerektirmez.
int main()
{
    int hak;
    char giris[100], kontrol[] = "duzenli calismak";
    printf("Deneme hakki:");
    scanf("%d", &hak);
    fflush(stdin);
    printf("Isin sirri: ");
    gets(giris);
    hak--;
    while( strcmp(giris, kontrol) != 0 && hak > 0)
    {
        printf("Bilemediniz. %d hakkiniz kaldi.\n", hak);
        gets(giris); printf("Isin sirri: ");
        hak--;
    }
    if( strcmp(giris, kontrol) != 0 )
        printf("Elendiniz.");
    else
        printf("Dogru bildiniz.");
    return 0;
}

```

→ Bonus'lu kısımlar

Lütfen karakterlerinizin (boşluk karakteri dâhil) açık, net, eksiksiz olduğunu kontrol ediniz.

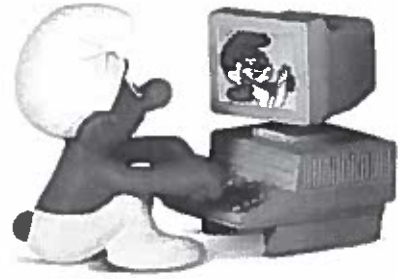
printf("Ornek); komutunda " eksik olması gibi,

<= yerine ≤ yazılması gibi durumların cevabınızda olmamasına özen gösteriniz.

Ancak harflerin yazımı ile ilgili ufak yazım hataları (f yerine F yazmak gibi), soruda özellikle ölçülenler arasında değilse, puan kaybına yol açmaz.

2) (Her boşluk 1 puan - en iyi 15 cevap değerlendirmeye alınır - toplam 15 puan)

Usta Şirin, ŞirinApp isimli bir mesajlaşma yazılımı icat etmek üzeredir. İcadının ilk denemelerini Şirin Baba ile yapacaktır. Ancak ŞirinApp'in kodlamasındaki bir mantık hatası yüzünden, Gargamel de yazışmaları takip edebilmekte ve hatta kendisi de mesajlar gönderebilmektedir. Bunu düşünerek, aşağıdaki mesajların kimden geldiğini ve mesaj içeriklerindeki boşlukları dikkatlice doldurunuz. Şirin Baba'nın her zaman doğru cevaplar verdiğini, Gargamel'in ise her zaman yanlış cevaplar verdiğini varsayınız.



BU SORULARIN HER BİRİNDEN SADECE TAM YA DA 0 PUAN ALIRSINIZ.

y yerine Y yazmak gibi yazım hataları, soruda özellikle ölçülenler arasında değilse, puan kaybına yol açmaz.

Usta Şirin: Şirin Baba, Şirin Soru Bankası'na çalışırken fark ettim. Yazışmalarımız oraya sızmış, bilgin olsun.

Şirin Baba: Şirin App'te küresel değişken kullanmışsındır. Biz yine başlayalım yazışmaya.

Usta Şirin: O zamanlar acemiydim. Tamam, başlıyorum öyleyse... Sana sorularım var Şirin Baba.

Şirin Baba: Dönem boyu düzenli çalıştıysan ne mutlu sana Usta Şirinim. Sor bakalım.

Usta Şirin: *int* türünde *x* değişkenim var. Kodumda "*x* dört basamaklı ise" koşul ifadesine ihtiyacım var. Aradığım ifade şöyle midir:

if($1000 \leq x < 10000$)

→ $1000 \leq x$ & $x < 10000$ olmalı.

Gargamel

: Kullandığın " $\leq$ " yerine C dilinde kullanılan " $<=$ " ifadesini kullanırsan işini görür.

Usta Şirin: Yapılar konusundaki $\rightarrow$ işaretini bir türlü anlamadım, tam olarak nedir?

Şirin Baba

: *ptr* isimli bir işaretçi olsun ve bu işaretçi bir yapıya işaret ediyor olsun. (**ptr*). yazmak yerine *ptr* $\rightarrow$ yazabiliyorsun.

Usta Şirin: *union* ne işe yarar?

Gargamel

: *union*(yapi1, yapı2) şeklinde kullandığında iki yapının birleşimini alıp *yapi1*'e kaydeder.

Usta Şirin: *int x*; şeklinde tanımlı değişkenimi uygun şekilde tanımlı fonk1 fonksiyonuna *fonk1(x)* şeklinde gönderince, fonksiyon onun üzerinde kalıcı değişiklik **yapamıyor**. Ama *int x[1]*; şeklinde tanımlı değişkenimi uygun şekilde tanımlı fonk2 fonksiyonuna *fonk2(x)* şeklinde gönderince, fonksiyon onun üzerinde kalıcı değişiklik **yapabiliyor**. Bunun nedeni nedir?

Şirin Baba

: Diziler fonksiyona adres ile giderler. Bu nedenle fonk2 fonksiyonu, değişkenin RAM'dekini yerini bilebilir ve değerini kalıcı olarak değiştirebilir.

Usta Şirin: `int x[1];` şeklinde tanımlı değişkenin ismi adres mi oluyor?

Şirin Baba: Evet, C dilinde bu şekilde bir dizi tanımlarsan, çağırma aşamasında `fonk2()` süslüleri içerisine `x` ifadesini

yazmak & `x[0]` yazmaya denktir.

Usta Şirin: Şöyle bir kod gördüm(aşağıda paylaşıyorum). Bu kod hata veya uyarı vermeksizin çalışabilir mi?

```
int main()
{
    int x = 56;
    return 0;
}
```

Gargamel: Hayır, bu kod çalışsa bile derleyici uyarı verecektir. Çünkü başına `stdio.h` kütüphanesi eklenmemiştir.

Usta Şirin: `int main(int argc, char* argv[])` ifadesindeki parantez içi ifadeler nedir?

Şirin Baba: Kod komut ekranından çalıştırıldığında koda parametre yollayabilmene imkan verir.

Usta Şirin: `FILE* dosya;` diye bir değişken tanımladım ve `fopen` ile açtığım dosyadan döngü içerisinde `fscanf` ile değer çekiyorum. Ancak bu dosyada değer kalıp kalmadığını nasıl anlarım? Döngüden ne zaman çıkmam gerektiğine onunla karar vereceğim.

Şirin Baba: Koduna ekleyeceğin `feof` (`dosya`) ifadesi 0 dönüyorsa dosyanın sonuna henüz gelinmemiştir. Onu kontrol ederek, kodunda döngüden çıkmaya karar verebilirsin.

Usta Şirin: `typedef` ile şöyle bir yapı tanımladım.

```
typedef struct
```

```
{
```

```
    int x;
```

```
}building;
```

Şimdi `main`'in içerisinde

```
building.x = 7;
```

```
printf("%lf", building.x);
```

yazdığımda kod hata veriyor. Bunun nedeni nedir?

Gargamel: `building.x` ifadesi `int` türünde olduğu için `%lf` değil de `%d` yazarsan sorun çözülür.

Usta Şirin: C dili nesne tabanlı bir dil midir?

Şirin Baba: Hayır, değildir ancak içindeki yapı kavramı 'nesne'ye çok benzemektedir. *ya da struct*

Usta Şirin: Kullanıcıdan üç tane isim alacağım, bu isimlerin her biri sadece "Sirine", "Gargamel" ya da "Guclu" olabilecek. Şirin kullanıcı bunların haricinde bir isim girişi yapmayacak. Bu 3 ismi içinde tutabilecek sirinler isimli tek bir değişkenim olsun istiyorum. Bu değişkeni nasıl tanımlarım?

Şirin Baba: *9'dan büyük bir sayı da olur.*
char sirinler [3][9]; şeklinde tanımlarsan işini görecektir. *8 harfli '10' minimum 9 gerekli.*

Usta Şirin: `int sirin[3][10];` şeklinde tanımlı *sirin* dizisi kaç tane *int* değerini içinde tutabilir?

Şirin Baba: 30

Usta Şirin: `int *y = &x;` ifadesi ne iş yapar?

Şirin Baba: `int *y;` ve `y = &x;` ifadeleri peş peşe geldiğinde ne iş yapıyorsa, senin yazdığın ifade de o işi yapar.

Usta Şirin: Kod her çalıştırıldığında farklı bir rakam yazdırsın istiyorum.

Gargamel: *rand(time(NULL)) şart.*
En başa gerekli kütüphaneleri ekler ve *main*'in içinde de `printf("%d", rand()%10);` şeklinde kullanırsan her çalıştırdığında farklı bir rakam yazdırır. Ancak ara sıra tesadüfen aynı rakam da çıkabilir.
bu durumda hep aynı 'seed' alır ve hep aynı rakam çıkar. Deneyiniz.

Usta Şirin: *malloc* fonksiyonunun kullanımı nasıldır?

Şirin Baba: Bellekten yer ayırt edip ilk baytının adresini sana dönen bir fonksiyon. 100'lük bir *int* dizisi açacaksan, `malloc(400)` yazıp döndüğü değeri bir *int* işaretçisine kaydedebilirsin.

Usta Şirin: Azman programlama öğrenmeye başlamış. Geçen gün ormana mesaj bırakmış, "strcat ne işe yarıyor? Kedi dizgisi mi üretiyor? Biri bana açıklasın, miyav" şeklinde. Yardımcı olabilir miyiz?

Gargamel: Tabii ki Şirinim, `strcat(dizgi1, "Azman");` şeklinde kullanarak "Azman" ifadesinin peşine *dizgi1*'in içeriğini ekleyebiliyorsun. *dizgi1'in peşine "Azman" ekler.*

Gargamel: Bu arada, şirinim, bunu Azman'a doğrudan iletirsen çok sevinir. Selamımı söylersin.

3) (Her boşluk 2 puan - en iyi 10 cevap değerlendirmeye alınır - toplam 20 puan)

Aşağıda bazı kodlar ve ekran çıktıları verilmiştir. Boşlukları doldurunuz.

• BU BOŞLUKLARIN HER BİRİNDEN SADECE, DOĞRU CEVABI TAM OLARAK VERMENİZ DURUMUNDA PUAN ALABİLİRSİNİZ. BİR BOŞLUK YANLIŞSA SADECE O BOŞLUKTAN PUAN KAYBEDERSİNİZ.

- / ve \ farkı, ' ve " farkı, ≤ yerine <= gibi detaylar da dâhil olmak üzere cevabınızın tam olarak doğruluğuna emin olunuz. r yerine R yazmak gibi durumlar, soruda özellikle ölçülenler arasında değilse, puan kaybına yol açmaz.
- Mümkün olduğunca sade komut/ifadeler kullanmanız beklenmektedir. Gereksiz komut/ifadeler eklemeyiniz.
- Sadece boşluklara yazdıklarınız değerlendirmeye alınır. Boşluklara cevap dışında hiçbir şey yazmayınız.
- Kod çalıştırıldığında derleyici herhangi bir uyarı ya da hata vermeyecek şekilde boşlukları doldurunuz.

KOD	ÖRNEK EKRAK ÇIKTISI
Kullanıcıdan değerler alarak, örnek ekran çıktılarındaki gibi cümle yazdıran kod <pre>#include <stdio.h> typedef struct { int gun; char ay[100]; int yil; }date; void duzelt(date*); void yazdir(date); int main() { <u>date tarih;</u> printf("Konser ne zaman?\n"); printf("Gun:"); scanf("%d", &tarih.yil); printf("Ay:"); scanf("%s", tarih.<u>ay</u>); printf("Yil:"); scanf("%d", &tarih.gun); duzelt(<u>tarih</u>); yazdir(tarih); return 0; } void duzelt(date* ptr) { int temp; <u>temp = ptr->gun</u>; ptr->gun = ptr->yil; ptr->yil = temp; } void yazdir(date tarih) { printf("%d yilinin %d %s gunu bulusmak uzere.", tarih.yil, tarih.gun, tarih.ay); }</pre>	Konser ne zaman? Gun:28 Ay:Kasim Yil:2018 2018 yilinin 28 Kasim gunu bulusmak uzere. ----- Process exited after 4.995 seconds with ret Devam etmek için bir tuşa basın . . . Örnek ekran çıktısı -1 Kod çalıştırıldıktan sonra sırasıyla "28", "Kasim", "2018" girişleri yapılmış ve her girişten sonra enter'a basılmıştır. Konser ne zaman? Gun:12 Ay:Temmuz Yil:2019 2019 yilinin 12 Temmuz gunu bulusmak uzere. ----- Process exited after 23.46 seconds with ret Devam etmek için bir tuşa basın . . . Örnek ekran çıktısı -2 Kod çalıştırıldıktan sonra sırasıyla "12", "Temmuz", "2019" girişleri yapılmış ve her girişten sonra enter'a basılmıştır. Gereksiz komut/ifadeler eklemeyiniz. Bu tür durumlarda kodunuz çıktı açısından isteneni yapıyor olsa bile puan alamayabilirsiniz. Kod çalıştırıldığında derleyici herhangi bir uyarı ya da hata vermemelidir.

KOD - İstatistikte ortalama(int. average, mean) gibi **ortanca**(ing. median) değer de kullanılır. Bir sayı dizisinin **ortancası**, sayılar artan (ya da azalan) sırada sıralandıktan sonra **ortada kalan** değerdir. 1 2 4 dizisinin **ortancası** 2'dir. Dizi eleman sayısı çift sayıda olursa, ortada iki eleman kalır. Bu durumda **ortanca**, bu iki elemanın ortalamasıdır. 1 2 3 5 dizisinin **ortanca** değeri (2+3) değerinin yarısı olan 2,5'tur.

Kullanıcının girdiği doğal sayı değerleriyle oluşan dizinin **ortancasını** veren kod. Kullanıcı, 10'dan az sayıda değer girdiğinde değerlerin sonuna -1 ekler ve enter'a basar. Kullanıcı, 10 ya da daha fazla değer girdiğinde, kod sadece ilk 10 değeri değerlendirmeye almak üzere tasarlıdır; bu durumlarda kullanıcının sayıların sonuna -1 eklemesine gerek yoktur.

ÖRNEK EKRAN ÇIKTISI

```
#include <stdio.h>
void siralat(int dizi[], int i);
int main()
{
    int dizi[10] = {}, i, j;
    printf("Sayılar:");
    for(i=0; i<10; i++)
    {
        scanf("%d", &dizi[i]);
        if(dizi[i]==-1) break;
    }
    siralat(dizi, i);
    printf("Kucukten buyuge:");

    for(j=0; j<i; j++)
        printf("%d ", dizi[j]);

    printf("\nOrtanca: ");
    printf("%.1f", (dizi[(i-1)/2] + dizi[i/2])/2.0);
    return 0;
}

void siralat(int dizi[], int i)
{
    int a, b, bittiMi=0;
    for(a=1; bittiMi==0; a++)
    {
        bittiMi=141;
        for(b=0; b!=i-a; b++)
        {
            if(dizi[b] > dizi[b+1])
            {
                int temp = dizi[b];
                dizi[b] = dizi[b+1];
                dizi[b+1] = temp;
                bittiMi = 0;
            }
        }
    }
}
```

*Tek sayıda eleman varsa ikisi de aynıdır.
Örn. i 3 ise, dizi[2/2] = dizi[3/2]*

Sayılar:5 6 985414 -1
Kucukten buyuge:5 6 985414
Ortanca: 6.0

Örnek ekran çıktısı -1

Kod çalıştıktan sonra kullanıcı 3 değer girmiş, peşine -1 eklemiş ve enter'a basmıştır. Ortanca değeri olan 6 değeri yazdırılmıştır.

Sayılar:9 8 7 6 5 4 3 2 1 0 50 27
Kucukten buyuge:0 1 2 3 4 5 6 7 8 9
Ortanca: 4.5

Örnek ekran çıktısı -2

Kod çalıştıktan sonra kullanıcı 12 değer girmiştir. Sonuna -1 eklemesine gerek kalmamıştır. Çünkü kod, 10'dan fazla değer girilirse ilk 10 tanesini dikkate almaktadır. Girilen ilk 10 eleman küçükten büyüğe sıralandığında ortada kalan değerler 4 ve 5 olduğu için, bunların ortalaması olan 4,5 ortanca değeri olarak yazdırılmıştır.

Sayılar:3 5 5 5 6 7 8 9 -1
Kucukten buyuge:3 5 5 5 6 7 8 9
Ortanca: 5.5

Örnek ekran çıktısı -3

Kod çalıştıktan sonra kullanıcı 8 değer girmiş, peşine -1 eklemiş ve enter'a basmıştır. Girilen elemanlar küçükten büyüğe sıralandığında ortada kalan değerler 5 ve 6 olduğu için, bunların ortalaması olan 5,5 ortanca değeri olarak yazdırılmıştır.

Sayılar:50 1000 80 9000 -1
Kucukten buyuge:50 80 1000 9000
Ortanca: 540.0

Örnek ekran çıktısı -4

Kod çalıştıktan sonra kullanıcı 4 değer girmiş, peşine -1 eklemiş ve enter'a basmıştır. Dizilimden sonra ortada kalan 80 ve 1000'in ortalaması olan 540 değeri ortanca olarak yazdırılmıştır.

Gereksiz komut/ıfadeler eklemeyiniz. Bu tür durumlarda kodunuz çıktı açısından isteneni yapıyor olsa bile puan alamayabilirsiniz.

#include <stdio.h>

int main()

{

int i,j,k=0;

for(i=0; i<3; i++)

{

for(j=0; j<k; j++)

printf("w");

for(j=i; j<i+2; j++)

printf("%d", j);

printf("\n");

k++;

}

return 0;

}

0	1								
w	1	2							
w	w	2	3						

#include <stdio.h>

enum booleanTuru {true, false};

int main()

{

enum booleanTuru p = !!!true;

printf("%d", p);

p = true && (false || !true);

printf("%d", p);

return 0;

}

1	0								

#include <stdio.h>

int schtroumpf(int puan);

int main()

{

int puan = 0;

do

{
puan = schtroumpf(puan);

}while(puan!=0);

printf("SCORE%d", puan);

return 0;

}

int schtroumpf(int x)

{

x+=100;

return x;

}

S	C	O	R	E	O				

#include <stdio.h>

#include <string.h>

int main()

{

// a'nin ASCII'si 97'dir.

char guclu[8] = {'a','s','c','i'};

strcpy(guclu, "98");

printf("%s", guclu);

return 0;

}

9	8								

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
void tembel(int []);

int main()
{
    srand(time(NULL));
    int dizi[10]={};
    tembel(dizi);
    printf("%d+2", dizi[0]+2); //dikkat!
    if(dizi[5]<30)
    {
        printf("A");
        else if(dizi[5]<=10)
        {
            printf("B");
        }
        else
        {
            printf("C");
        }
    }
    return 0;
}
```

```
void tembel(int dizi[])
{
    int i=1;
    while(i!=10)
    {
        dizi[i]=(rand()+80)%10+1;
        i++;
    }
}
```

2	+	2	A						

```
#include <stdio.h>
#include <string.h>
int main()
{
    char name[100] = "le schroumpf grognon";
    name[2]='\0';
    if(strcmp(name,"le"))
    {
        printf("%s", name);
        else
        {
            printf("%s", name+1);
        }
    }
    return 0;
}
```

e									

```
#include <stdio.h>
double somurtkan(double);
int main()
{
    double suslu[3]={3.14,2.22,1.41};
    somurtkan(suslu[0]);
    suslu[1] = somurtkan(suslu[1]);
    printf("%.2lf\n",suslu[0]+suslu[1]);
    return 0;
}

double somurtkan(double x)
{
    x = 7.20;
    printf("%.2lf\n",x);
    return x;
}
```

7	.	2	0						
7	.	2	0						
1	0	.	3	4					

```
#include <stdio.h>
double somurtkan(double*);
int main()
{
    double suslu[3]={3.14,2.22,1.41};
    somurtkan(&suslu[0]);
    suslu[1] = somurtkan(&suslu[1]);
    printf("%.2lf\n",suslu[0]+suslu[1]);
    return 0;
}

double somurtkan(double* ptr)
{
    *ptr = 3.00;
    printf("%.2lf\n", *ptr);
    return (*ptr)+1.0;
}
```

3	.	0	0						
3	.	0	0						
7	.	0	0						

```
#include <stdio.h>
typedef struct
{
```

```
char isim[10][10][100];
```

```
}data;
```

```
int main()
{
```

```
data veri;
```

```
FILE* dosya = fopen("obur.txt", "r");
```

```
int a = 9, k = 4;
```

```
while(a>=4)
```

```
{
```

```
fscanf(dosya,"%s", veri.isim[a-2][k%3]);
```

```
k++;
```

```
--a;
```

```
}
```

```
printf("%s\n", veri.isim[7][1]);
```

```
printf("%s\n", veri.isim[5][0]);
```

```
return 0;
```

```
}
```

C	i	l	e	k	l	i					
2	9										

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

```
int main()
```

```
{
```

```
int sakar[5] = {4,9,1,6,2};
```

```
int* b = &sakar[2];
```

```
int **c;
```

```
c = &b;
```

```
printf("%d", sakar[2]);
```

```
(*c)--; → sakar[2]'i 1 azalt
```

```
printf("%d", sakar[2]);
```

```
(*b)++; → sakar[2]'i 1 arttır.
```

```
printf("%d", *b);
```

```
b++; → sakar[3]'e işaret et.
```

```
printf("%d", *b);
```

```
b++; → sakar[4]'e işaret et.
```

```
printf("%d", *b);
```

```
return 0;
```

```
}
```

1	0	1	6	2							

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

```
int main()
```

```
{
```

```
char x='a', y='x', z='t';
```

```
if( (int)z == (int) x )
```

```
printf("Bir");
```

```
else if(y == x)
```

```
printf("İki");
```

```
else if(0)
```

```
printf("Uc");
```

```
else if(1)
```

```
printf("Dort");
```

```
else
```

```
printf("Bes");
```

```
return 0;
```

```
}
```

D	o	r	t								

```
ascii[0] ←
```

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

```
int main()
```

```
{
```

```
FILE* dosya = fopen("obur.txt", "r");
```

```
int i, suslu[100];
```

```
char ascii[100][100];
```

```
for(i=0; i++)
```

```
{
```

```
fscanf(dosya, "%[^0123456789]s", ascii[i]);
```

```
fscanf(dosya, "%d", &suslu[i]);
```

```
printf("%c%d\n", ascii[i][0], suslu[i]);
```

```
if(suslu[i]<25) break;
```

```
}
```

```
return 0;
```

```
}
```

C	2	9									
S	1	9									

→ boy → 141
 depe
 Gope donustlu.

13

```

#include <stdio.h>
#define X 3
#define Y 6
int main()
{
    int dizi[2][X][Y]={ { {1,0,0,6,8,3},{2324,0,8},{0,2,3,11,2} } , {} };
    int i,j;
    for(i=0; i<X; i++)
    {
        for(j=0;j<Y;j++)
            dizi[1][2-i][j]=dizi[0][i][j];
    }
    for(i=0; i<X; i++)
    {
        for(j=0;j<Y;j++)
        {
            if(dizi[1][i][j]!=0)
            {
                printf("%d",dizi[1][i][j]);
                printf(" "); //1 boşluk
            }
        }
        printf("\n");
    }
    //cevabinizi iyice kontrol ediniz.
    return 0;
}

```

Algoritma

1	0	0	6	8	3
2	3	2	4	0	8
0	2	3	1	1	2

yelerini değiştiriyor.

ve

0 harici değerleri yazdırıyor.

2		3		1	1		2		
2	3	2	4		8				
1		6		8		3			

```

#include <stdio.h>
int main(int argc, char* argv[])
{
    printf("%d",argc); // %dden sonra \n yok!
    printf("%s", (argc%2==1) ? argv[2]:argv[1]);
    return 0;
}

```

3*2=1 ✓

argv[0] argv[1] argv[2]

Komut Ekranı

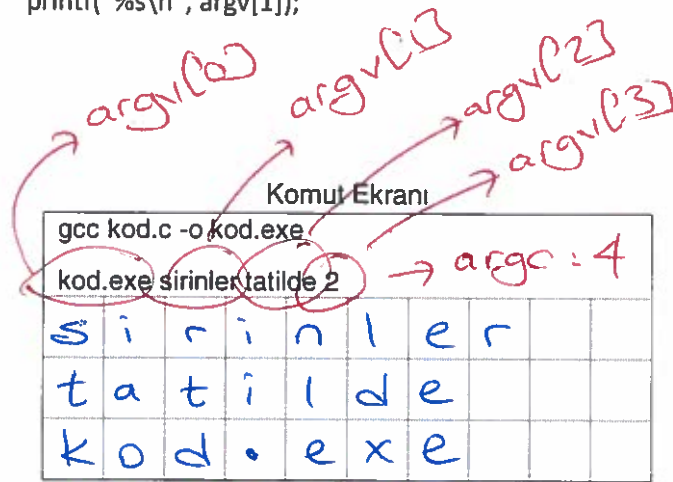
gcc kod.c -o kod.exe

kod.exe -sirinler tatilde] argc=3

3 t a t i l d e

(Kullanıcı komut ekranına, verilen iki satır komutu girmiş ve her komuttan sonra enter'a basmıştır.)


```
#include <stdio.h>
int main(int argc, char* argv[])
{
    int i;
    for(i=0; i<argc-1; i++)
    {
        switch(i)
        {
            case 1:
                printf("%s\n", argv[2]);
                break;
            case 2:
                printf("%s\n", argv[0]);
                break;
            default:
                printf("%s\n", argv[1]);
        }
    }
    return 0;
}
```



(Kullanıcı komut ekranına, verilen iki satır komutu girmiş ve her komuttan sonra enter'a basmıştır.)

```
#include <stdio.h>
typedef struct
{
    char sirinlsim[100];
    double boy;
}smurf;
int main()
{
    smurf asci;
    asci = (smurf){.boy=0.000012, .sirinlsim = "Asci Sirin"};
    printf("%.2lf\n", asci.boy);
    printf("%s\n", asci.sirinlsim);
    return 0;
}
```

0	.	0	0						
A	s	c	i		S	i	r	i	n

GERİBİLDİRİM SORULARI (EKSTRA 5 PUAN)

4 PUAN: Eğer aşağıdaki maddeleri cevaplarsanız & 1 PUAN: Girilen sayı alınan nota yakın olursa

1. Aşağıya bir sayı yazınız. Yazdığınız sayının sınavdan alacağınız nota maksimum 5,0 puan yakın olması durumunda ekstra 1 puan alırsınız. (Bu kısımdan alınabilecek 1 puanı dâhil etmeden sayı girişinde bulununuz. Mantıklı olan aşağıdaki boşluğu [5,0-95,0) aralığında bir değerle doldurmaktır.)

Sayı = _____ ± 5,0

2. Bu dersten almayı beklediğiniz harf notu nedir? AA BA BB CB CC DC DD FF

3. Bu sınava yeterince çalışabildiğinizi düşünüyor musunuz?

4. Kuramsal ve uygulamalı dersler hakkında neler düşünüyorsunuz?

5. Ödevler ve kısa sınavlar hakkında neler düşünüyorsunuz?

6. Geri bildirimde bulunmak istediğiniz diğer konular nelerdir?