

BİLGİSAYAR PROGRAMLAMA**ARA SINAV I**

Sınav Süresi:120 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 telefonuyla 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 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) (10 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 örnek ekran çıktılarındaki görüldüğü gibi kullanıcıdan **bir sözcük** alacaktır.
 - Kullanıcı **tek sözcüklük bir metin** girip enter'a basacaktır.
 - Kullanıcı maksimum **69 harfli bir sözcük** girecektir. Maksimum (69) harfli giriş yapıldığında kodunuz bu veriyi saklamak için **gereksiz yere büyük değişken** kullanmamalıdır; kullanıcı daha az harfli (örn. 8 harfli) bir giriş yaptığında değişkeninizin bir kısmının kullanılmıyor olması normaldir.
 - Girişteki harflerin, **İngiliz alfabesindeki 52 harf** ([a-z] arası 26 küçük harf ve [A-Z] arası 26 BÜYÜK harf) arasından seçileceğini ve bu 52 harften başka harfin girişte (ğ, ş gibi) olmayacağını varsayınız. Kullanıcının her zaman geçerli girişler yapacağını (sayı, boşluk, cümle vs. girmeyeceğini) varsayınız.
- Kod, bu sözcükteki **BÜYÜK harf ([A-Z] arası harf) sayısını** örnek ekran çıktılarındaki gibi ekrana yazdıracaktır.
- Kod, bu sözcükteki küçük harflerin karşılık gelen BÜYÜK harflerle (örn. 'd' ise 'D') , BÜYÜK harflerinse karşılık gelen küçük harflerle (örn. 'R' ise 'r') **değiştirilmiş şeklini**, örnek ekran çıktılarındaki gibi, ekrana yazdıracaktır.
- Kodunuzun **return değeri 0** olmalıdır.

```
Giris: gArGaMeL
Buyuk harf sayisi: 5
Yeni sekli: GaRgAmel
-----
Process exited after 23.67 seconds with return value 0
Devam etmek için bir tuşa basın . . .
```

ÖRNEK EKRAN ÇIKTISI 1

Kod çalıştıktan sonra kullanıcı "gArGaMeL" girişi yapıp enter'a basmıştır. "A-G-M-E-L" olarak 5 tane BÜYÜK harf olduğu için bu belirtilmiştir. Sonra da yeni şekli 'g' yerine 'G', 'A' yerine 'a', 'r' yerine 'R', 'G' yerine 'g'... şeklinde ekrana yazdırılmıştır.

```
Giris: CEKOSLAVakyalilastiraMAdiklariMIZdanMIScasina
Buyuk harf sayisi: 15
Yeni sekli: cekOslavAKYALILASTIRAmADIKLARIMizDANmisCASINA
-----
Process exited after 2.416 seconds with return value 0
Devam etmek için bir tuşa basın . . .
```

ÖRNEK EKRAN ÇIKTISI 2

Kod çalıştıktan sonra kullanıcı uzun bir sözcük girişi yapıp enter'a basmıştır. "C-E-K-S-L-A-V-M-A-M-I-Z-M-I-S" olarak 15 tane BÜYÜK harf olduğu için bu belirtilmiştir. Sonra da yeni şekli 'C' yerine 'c', 'E' yerine 'e', 'K' yerine 'k', 'o' yerine 'O'... şeklinde ekrana yazdırılmıştır.

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.
- Benzer şekilde; **kütüphane eklenmesi** gibi olgulara dikkat etmeden yazdığınız kodlarda da kodunuz doğru çalışsa bile puan kaybı yaşanabilir. Örneğin, kodunuzda gereksiz yere eklenmiş kütüphane bulunmasın.
- Kod içerisinde **boşluk** bırakmanın kritik olduğu kısımlardaki boşluklarınız **açık olmalı**. Gerekirse ilgili satıra yorum ekleyerek bunu belirtiniz.
- Bu soruyu çözebilmek için **ASCII tablosunu** ezbere bilmeye gerek yoktur.
- C dilinin **char** ve **int** arasındaki **tür dönüşümleri**ni arka planda başarılı şekilde yapabildiğini varsayabilirsiniz.
- Kodunuzu en baştan **girintilere** özen göstererek yazarsanız, algoritmanızı daha **okunur** kılarak olası hatalardan ({ } unutmak gibi) kolaylıkla kaçınırsınız. Kodlarda derleme imkanı olmadığı için rahatlıkla ; gibi **detaylar** unutulmaktadır. Ayrıca, **kötü/hızlı yazımdan** kaynaklı (t gibi gözüken f harfleri, ı 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**.

// BIL141 DERSİ SINAVINDAKİ 1.SORU KODUM



Cevabınızı en
ŞİRİN YAZINIZLA,
girintilere özen
göstererek ve
dikkatlice bu şirin
sayfaya yazınız.

Kısa ve sade bir
şirin **SÖZDE KOD**
ile algoritmanızı
oturttukten sonra
koda başlarsanız
daha başarılı
olabilirsiniz.

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

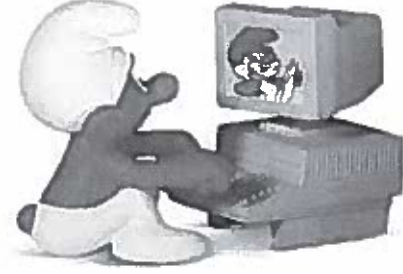
`printf("Ornek);` komutunda " eksik olması gibi,

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

Ancak sınav kağıdına yazdığınız 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: Merhaba Şirin Baba. Babalar günün kutlu olsun. Şirin sıcak bir ay... yine ben... ve yine programlama dersi ile ilgili sorularım.

Şirin Baba: Sağ ol şirinim. Ben de az önce düzenli ders çalışmayan seni andım. İçimden "Usta'nın düzenli çalışacağı yok, dersin hocası değişmişken karambolde belki şirin vasat bir notla dersi geçiverir." diyordum.

Usta Şirin: Yok Şirin Baba, o şekilde bir değişim değilmiş; baksana her şey aynı, yine burada seninle sınav kağıdının ortasında soru-cevap yapıyoruz. Ama zaten ben bu dönem düzenli çalışıp şirin bir not ile dersi geçeceğim. Bu ders şirin geleceğim için bana gerekli.

Şirin Baba: Peki, Şirinim. Başarılar sana sınavında. Ne sorular getirdin bakalım?

Usta Şirin: Geçen gün Şirine'ye "beni İnşiringam'ına ekler misin?" dedim, "Bilmem ki, bir kod yaz, kod karar versin." dedi. Gerekli kütüphaneleri, komutları vs. ekledikten sonra %50 ihtimalle "Ekle" diyen şu kodu yazmak istedim ama ???????? yazdığım yere ne yazmam gerek?

```
if( rand() ??????? )
    printf("Ekle");
else
    printf("Uzak dur");
```

Şirin Baba: ???????? kısmını silip, oraya _____ yazman yeterli olur.

Usta Şirin: Yaptık ve "Ekle" çıktı!!! Şimdi teşekkür olarak ormandan beşli böğürtlen paketleri göndereceğim. [50-80] aralığında (50 ve 80 dahil) 5'in katı olan sayı üreten ifade nasıldır?

Şirin Baba: Şirin bilgi yarışması ödevini yapmadın mı yoksa? $(P + \text{rand}() \% Q) * R$ şeklinde bir ifade olabilir. Ama P, Q, R değişkenlerinin sayı değerlerini sen bulmalısın. Ben sadece sana şu ipucunu vereyim. P, Q ve R sayılarını

çarpınca _____ eder.

Usta Şirin: Kullanıcı klavyeden bir harfe bastığında, bastığı harf ekranda **gözükmesin**, peşinden enter'a basmasına gerek kalmasın ve ben hemen girilen harfin **ASCII değerini** fonksiyonun döndüğü değer olarak alabileyim istiyorum. Hangi fonksiyonu önerirsin?

_____ : *conio.h* kütüphanesindeki *getche* fonksiyonu tam olarak aradığın fonksiyondur.

Usta Şirin: *int array[5];* dedikten sonra dizideki son elemana 21 sayısını kaydetmek istiyorum. Ancak *array[5] = 21;* yazdığımda hata veriyor. Doğrusu nedir?

_____ : Bu dizinin son elemanına kayıt yapabilmek için kodunda *array[5] = 21;* ifadesini kullanman gerekir.

Usta Şirin: Kullanıcıdan bir karakter almak istiyorum. *char karakter;* tanımını yaptım ve peşinden *scanf("%c", karakter);* şeklindeki kod parçasıyla aldığım değeri *karakter*'e kaydetmeye çalıştım. Ancak kodum çalıştıktan sonra, deneme amaçlı 1 girip enter'a basınca program çöküyor, neden?

_____ : *%c* karakter almak içindir ama sen denemek için rakam girmişsin. Rakam almak için *%d* kullanmalıydın.

Usta Şirin: Kodumda *main* içerisinde *char sirine[100];* şeklinde bir dizgi tanımladım. *ilanEt* isimli aşağıdaki gibi bir fonksiyonum var. Fonksiyonumun prototipini de *void ilanEt(char);* olarak *main*'in başına koydum.

```
void ilanEt( char sirine )
{
    printf("Usta kalp %c ", sirine);
    sirine = 'S';
    return;
}
```

Şirin Baba, sana sorum şu şekilde: *ilanEt* fonksiyonunu *main*'den *ilanEt (sirine[0]);* şeklinde çağırırsam hatasız şekilde çalışır mı?

_____ : Çalışmaz, çünkü sen *char* değişkeni göndereceğin yerde *main*'deki *sirine* dizgisini gönderiyorsun.

Usta Şirin: Peki, doğru şekilde çağırıldığında fonksiyon içerisinde yapılan 'S' değer ataması *main*'de de geçerli olur mu?

_____ : Hayır, gönderdiğin değişkenin fonksiyonda kopyası çıkarıldığı için değer atamasının *main*'e etkisi olmaz.

Usta Şirin: Daldan dala atlıyorum ama kusura bakma. *main* içerisinde şöyle bir kod yazdım.

```
FILE* araba= fopen("veri.txt", "w");
fprintf(araba, "12");
// ????????
araba= fopen("veri.txt", "r");
int x = -1;
fscanf(araba, "%d", &x);
printf("%d", x);
```

Çalıştırdığımda ekrana 12 yazmasını beklerken -1 yazdırıyor. Nasıl düzelir acaba?

Şirin Baba: // ???????? ifadesini silip, onun yerine _____; yazarsan düzelir.

Usta Şirin: Uygun kütüphaneleri vs. ekledikten sonra *main* içerisine

```
if( sayi > 10 )
    printf("İki basamaklı");
else ( sayi < 10 )
    printf("Tek basamaklı");
```

yazdım. Kodum, ekrana *sayi*'nin basamak adedini yazsın istiyorum. *sayi* değişkeni [1-99] arasında değer içeren bir değişkendir. Bu kod parçası doğru çalışır mı?

_____: İki basamaklı olan 10'u da dahil etmek üzere >10 ifadesi yerine >=10 ifadesi yazarsan kod istediğin gibi doğru çalışacaktır.

Usta Şirin: Herhangi bir fonksiyon içinde tanımlanmamış değişkenlere ne denir? Baş harfi k idi ama gerisini hatırlayamadım.

Şirin Baba: Fonksiyonların dışında tanımlanmış bu değişkenlere C dilinde ve Türkçe olarak *k* _____ *değişken* denir. ŞirinC'de *gargamel değişken* olarak da geçer.

Usta Şirin: Unutkan Şirin gelmiş, *istGelin* dizisindeki [1][1] indisli elemanın değerini unuttum diyor. Dizinin tanım satırı şu şekilde: *int istGelin[2][3] = { { 74, 233, 667 } , { 93, 46, 142 } };*

Şirin Baba: Bu tanım ve değer atama işlemine göre *istGelin[1][1]*, _____ değerine sahiptir.

Unutmamak için çok çalışmak gerekiyor.

Usta Şirin: *Blok içerisinde atanan blok içinde kalır*, diye hatırlıyorum. Uygun şekilde kütüphaneler vs. eklendiğinde, main içerisine yerleştirilen aşağıdaki kod parçası ekrana 7 yazdırması lazım, değil mi?

```
int azman = 7;
{
    azman = 8;
    int azman;
    azman = 9;
}
printf("%d", azman);
```

_____ : Hayır, öyle bir şey yoktur; doğrusu "*Blok içinde tanımlanan blok içinde kalır*" şeklindedir.

Bu kod ekrana 8 yazdıracaktır.

Usta Şirin: Peki, *kabarcık sıralama* algoritmasını 64 elemanlı bir diziye uyguluyorum. Baştan sona doğru, çiftler halinde sayıları incelediğim her tura bir *geçiş* diyorum. En az kaç *geçiş* sonrasında "dizi kesinlikle sıralanmıştır" diyebiliriz?

Şirin Baba: 64 elemanlı bir diziyi kesinlikle sıralı hale getirmek için en az _____ *geçiş* gerekir.

Usta Şirin: *scanf("%d", &sayi);* şeklindeki bir kod parçasını içeren bir kodum var. Kod çalışırken bu satıra gelindiğinde, klavyeden "a" harfini yazıp enter'a basarsam ne olur?

_____ : Harflerin de C dilinde sayısal karşılıkları olduğu için, "a" harfinin ASCII değeri *sayi* değişkenine kaydedilir.

Usta Şirin: Demek ki *sayi*, şirin 97 değerini alır diyorsun. Tamamdır. İçinde "*tutku*" olan bir karakter dizisi istiyorum.

char dizi[] = { t, u, t, k, u }; şeklinde yapsam "*tutku*" ifadesindeki harfleri dizide saklamış olur muyum?

_____ : Bir tek harfleri tek tırnak içerisinde (örn. 't') alman eksik olmuş. Onu da yaparsan başarılı bir şekilde saklamış olursun.

Usta Şirin: Geçen gün, Uykucu Şirin gelmiş, "*Geçen uzanmış uyanmaya çalışırken aklıma takıldı: [15-X] (15 ve X dahil) aralığında bir sayı tutacak olayım, sen de bu sayıyı tahmin edecek ol. Ben, senin her tahmininden sonra "çık", "in" ya da "buldun" şeklinde seni yönlendireceğim. Senin 7 tahmin hakkın var ve optimum algoritma ile sayımı bulmaya çalışacaksın. Her durumda sayımı bulabilmen için X sayısı en çok kaç olabilir Usta?*" diye bana soruyor.

Şirin Baba: Optimum algoritma kullanılacaksa X en büyük _____ değerini alabilir Ustacığım. Şu *optimum* sözcüğünü de atalım artık dilimizden, onun özşirincesi *şirinmumdur*.

Usta Şirin: Tamam Şirin Baba. *fprintf(stdout, "Şirince kal.");*

3) (Her boşluk 2 puan - en iyi 16 cevap değerlendirmeye alınır - toplam 32 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, $\leq$ yerine $\leq$ 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.

SÖZDE KOD	SÖZDE EKRAN ÇIKTISI
1. başla 2. _____ 3. sayi=2 4. Eğer sayi > 300 ise _____ 5. Eğer sayi mod 10 = 0 ise sayac=sayac+1 6. sayi=sayi+2 7. Git 4 8. Yaz sayac 9. Dur	30 Sözde kodun ekran görüntüsü yukarıdaki şekilde bir sayıdan oluşmaktadır. Soruda boşlukların dışını doldurmayınız. Sadece boşluklara yazdıklarınız değerlendirmeye alınır. Bir boşluğa bir komut yazınız. Gereksiz komut/ifadeler eklemeyiniz. Bu tür durumlarda kodunuz çıktı açısından isteneni yapıyor olsa bile puan alamayabilirsiniz.

KOD - Çalıştığında ekrana "1" yazdıran kod	EKRAN ÇIKTISI
<pre>#include <stdio.h> // iyi ki Odev3'u fonksiyonlu yaptım char harftenRakama (char); int main() { char harf1 = _____; printf("%c", harftenRakama(harf1)); return 0; } char harftenRakama (char _____) { _____(x) { case 'b': return '1'; default: return '0'; } }</pre>	<pre>1 ----- Process exited after 0.006962 seconds with return value 0 Devam etmek için bir tuşa basın . . .</pre> ÖRNEK EKRAN ÇIKTISI Kod çalıştırıldıktan sonra herhangi bir kullanıcı girişi olmamıştır ve görüldüğü gibi bir ekran çıktısı sağlanmıştır. 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. Soruda boşlukların dışını doldurmayınız. Sadece boşluklara yazdıklarınız değerlendirmeye alınır.

KOD - Kullanıcıdan bir para değeri alan, bu değer 20'den küçük olduğu sürece tekrar değer girişi isteyen, kendisi de [25, 50] arasında 5'in katı olan bir para değeri üreten ve örnek çıktılarıdaki gibi çalışan kod

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main()
{
    _____ ( time(NULL) );

    int tl;
    printf("Ne yiyelim kanka?\n\t\tKac liran var?\n");
    scanf("%d", &tl);

    _____ ( _____ (tl>=20) )
    {
        printf("\t\tCok az.\n");
        printf("\t\tTum ceplerine iyice bak.\n");
        printf("\t\tKac liran var?\n");
    }

    int ben = 25+ rand()_____;
    printf("\n\t\tBende de %d lira var.\n", ben );
    if(tl+ben>=30*2)
        printf("Iki iskender soyleyelim mi?");

    _____
        printf("Ben ac degilim zaten.");
    printf("\n\t\tBana uyar kanka.");
    return 0;
}
```

EKRAN ÇIKTISI

```
Ne yiyelim kanka?
                Kac liran var?
27
                Bende de 45 lira var.
Iki iskender soyleyelim mi?
                Bana uyar kanka.
```

ÖRNEK EKRAN ÇIKTISI

Kod çalıştırıldıktan kullanıcı "27" yazıp enter'a basmıştır. Kod [25, 50] aralığında ve 5'in katı olan 45 değerini üretmiştir.

```
Ne yiyelim kanka?
                Kac liran var?
17
                Cok az.
                Tum ceplerine iyice bak.
                Kac liran var?
19
                Cok az.
                Tum ceplerine iyice bak.
                Kac liran var?
20
                Bende de 25 lira var.
Ben ac degilim zaten.
                Bana uyar kanka.
```

ÖRNEK EKRAN ÇIKTISI

Kod çalıştırıldıktan kullanıcı sırasıyla "17", "19" ve "20" yazıp her girişten sonra da enter'a basmıştır. İlk iki giriş 20'den küçük olduğu için kod tekrar giriş yapılmasını istemiştir. Kod [25, 50] aralığında ve 5'in katı olan 25 değerini üretmiştir.

Gereksiz komut/ıfadeler eklemeyiniz. Bu tür durumlarda kodunuz çıktı açısından isteneni yapıyor olsa bile puan alamayabilirsiniz. Kod çalıştırıldığında derleyici herhangi bir uyarı ya da hata vermemelidir. Soruda boşlukların dışını doldurmayınız. Sadece boşluklara yazdıklarınız değerlendirilmeye alınır.

KOD - Çalıştığında ekrana örnek çıktıdaki şekilde çıktı veren kod

```
#include <stdio.h>
/*burada fonk prototipinin dogru sekilde bulunduğunu varsayınız*/
int main()
{
    int a[4] = { 3 , 4 , 5 }, b[4]={ 7 , 9 , 1 , 2 };
    printf("% .21f-% .21f\n", fonk( a ), fonk( b ) );
    int i = 0;
    while( i<_____ )
    {
        printf("%d ", a[_____]);
    }
    return 0;
}

_____ fonk ( _____ )
{
    _____ = 9;

    return v[0]_____v[2];
}
```

EKRAN ÇIKTISI

```
0.60-7.00
3 9 5 0
-----
Process exited after 0.009524 seconds
with return value 0
```

ÖRNEK EKRAN ÇIKTISI

Kod çalıştırıldıktan sonra herhangi bir kullanıcı girişi olmamıştır ve görüldüğü gibi bir ekran çıktısı sağlanmıştır.

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

Kod çalıştırıldığında derleyici herhangi bir uyarı ya da hata vermemelidir.

Soruda boşlukların dışını doldurmayınız. Sadece boşluklara yazdıklarınız değerlendirilmeye alınır.

4) (Her çıktı 3 puan - en iyi 16 cevap değerlendirmeye alınır - toplam 48 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.

• Hata içerdiğini düşündüğünüz sorularda(mantık hatası haricindekilerde) 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 HER İKİSİ DE TAM OLMALIDIR.

HATA1:Kod hiç derlenemeyecektir.

HATA2:Kod derlenecek ama çalışması esnasında hata verecektir.

HATA3:Kod derlenecek ama çalışması esnasında sonsuz döngüye girecektir.

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

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

```
int main()
```

```
{
```

```
    int t = 1, b = 2;
```

```
    b *= t;
```

```
    printf("%d%d\n", t, b);
```

```
    scanf("%d", &t);
```

```
    printf("%d%d\n", t, b);
```

```
    return 0;
```

```
}
```

*/*Kod çalıştıktan sonra kullanıcıdan değer istendiğinde "18" girmiş ve enter'a basmıştır. Çıktının en son hali istenmektedir.*/*


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

```
int main()
```

```
{
```

```
    char dizgi[2][20] = {"Asci", "8lu"};
```

```
    printf("Sirin\t##\n");
```

```
    printf("%s\t\\\\\\n", dizgi[0]);
```

```
    printf("%s\t\\\"\\\"\\n", dizgi[1]);
```

```
    /* LUTFEN KARAKTERLERİNİZ  
    NET YAZILMIS OLSUN */
```

```
    return 0;
```

```
}
```

/ Çıktının en son hali istenmektedir. Çıktının bir kısmı verilmiştir, geriye kalanını sizin tamamlamanız istenmektedir.*/*

S	i	r	i	n				#	#

```
#include <stdio.h>
int main()
{
    int i, j;
    for( i = 1 ; i <= 2; i++ )
    {
        for( j = 1 ; j <= 4; j++ )
        {
            switch( (i+j) % 2 )
            {
                case 1:
                    printf("A");
                    break;
                case 0:
                    printf("B");
                    break;
                default:
                    printf("C");
            }
        }
        printf("\n");
    }
    return 0;
}
```


```
#include <stdio.h>
#define SAYI 8
void fonk(int);
int main()
{
    fonk(SAYI);
    return 0;
}
void fonk(int a)
{
    printf("%5.3d", a);
    return;
}
```


```
#include <stdio.h>
int main()
{
    int x = 8;
    do
    {
        x -= 2;
        printf("%d", x);
    }
    while( (x = 0) ); // == degil
    printf("%d", x);
    return 0;
}
```


```
#include <stdio.h>
int main()
{
    int x = 0;
    int y = 9;
    {
        printf("%d", y/x);
        printf("x/y");
    }
    return 0;
}
```


```
#include <stdio.h>

void k(int dizi[], int n);

int main()
{
    int d[]={8 , 1 , 5 , 7 , 3 , 4 , 2 , 6 , 7 , 9};
    int i;

    /* ipucu: sizeof(d) ifadesi d dizisinin RAMde kac bayt
        yer tuttugunu verir */

    int n = sizeof( d ) / sizeof( d[0] );
    k(d, n);
    for( i = 0 ; i < n; i++)
        printf("%d", d[i]);
    return 0;
}

void k( int dizi[] , int n )
{
    int i, t;
    int b;
    do
    {
        b = 0;
        for( i = 0 ; i < n-2; i++)
        {
            if(dizi[i+1] > dizi[i])
            {
                t = dizi[i];
                dizi[i] = dizi[i+1];
                dizi[i+1] = t;
                b = 1;
            }
        }
        n--;
    }
    while( b );
}
```


```
#include <stdio.h>

#include <string.h>

void g( char [], char [] );

int main()
{
    char a1[11]= "zebra";
    char a2[11]= "hamsi";

    g(a1, a2);

    printf("%s\n", a1);
    printf("%s\n", a2);

    return 0;
}

/* asagidaki satirda once a2, sonra a1 yazmaktadir. */

void g( char a2[], char a1[] )
{
    if( !strcmp(a1, a2) ) // ! var.

        strcpy( a2 , a1 );

    else

        strcat( a1 , a2 );

    return;
}
```


```
#include <stdio.h>
void al(char[]);
int main()
{
    char g[20] = {};
    printf("G:");
    al(g);
    int i;
    for( i = 0 ; g[i]!='\0' ; i++ ); // ; var.
    printf("%d\n", i);
    printf("%s", g);
    return 0;
}
void al(char h[])
{
    scanf("%[^2]s", h);
    return;
}
```

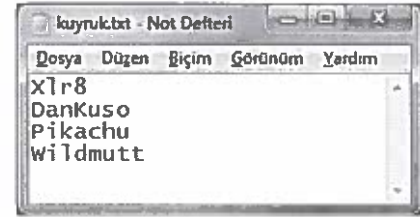
/*Kod çalıştıktan sonra kullanıcı "15 12 42" girmiş ve enter'a basmıştır. enter nedeniyle çıktının bir alt satırdan devam etmesi beklenir. Çıktının en son hali istenmektedir. Çıktının bir kısmı verilmiştir, geriye kalanını sizin tamamlamanız istenmektedir.*/

G	:	1	5		1	2		4	2

```
#include <stdio.h>
int sunger(int bob);
int kare(int sort);
int main()
{
    printf("%d", sunger(5));
    printf("%d", kare(8));
    return 0;
}
int sunger(int bob)
{
    printf("%d", bob);
    return bob;
}
int kare(int sort)
{
    sort = sort + bob;
    return sort;
}
```




```
#include <stdio.h>
int main()
{
    FILE* klasor = fopen("dosya.txt", "r");
    char x = (klasor!=NULL)?'a':'b';
    double y;
    fscanf(klasor, "%lf", &y);
    printf("%c%d", x, (int)y*10);
    return 0;
}
```

```
#include <stdio.h>
int main()
{
    char sıra[20];
    int i;
    FILE* queue = fopen("kuyruk.txt", "r");
    for( i = 1; i < 9; i += 2 )
    {
        if(i==3) continue;
        fscanf(queue, "%s", sıra);
        printf("%s\n", sıra);
    }

    return 0;
}
```


```
#include <stdio.h>

int main()
{
    int dizi[7] = {};
    int i;
    for(i=0; i<7; i++)
        dizi[i] = (i*i);
    printf("%d\n", i);
    for( i=0 ; i!=9 ; i+=3 )
        printf("%d", dizi[i]);
    // printf'lerden sadece birinde \n var.
    return 0;
}
```


```
#include <stdio.h>
#include <math.h>
double k1 (int a);
double k2 (double b);
int main()
{
    printf("%.2lf\n", k1( 36 ) );
    printf("%.2lf\n", k2( 64 ) );
    return 0;
}
double k1 (int a)
{
    double b = (double)a;
    return pow( b, (double)( 1/2 ) );
}
double k2 (double b)
{
    return pow( b, (double)1/2 );
}
```


```
#include <stdio.h>

int main()
{
    int a=4;
    while( a<2 ); // ; var
    {
        printf("%d", a);
        a++;
    }
    printf("%d", a);
    return 0;
}
```


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

int main()
{
    int x = getch() - (int)'0';
    int y = getch() - (int)'0';
    //2.si getch'e degil
    if( x>5 )
        printf("A");
    else if( y>x )
        printf("B");
    else if( x>0 )
        printf("C");
    return 0;
}
```

*/*Kod çalıştıktan sonra kullanıcı "3" ve "4" tuşlarına sırasıyla basmıştır. Çıktının tamamı ve en son hali istenmektedir.*/*


```
#include <stdio.h>
void yildiz ( int );
void bosluk ( int );
int main()
{
    int w = 3;
    int i;
    for( i = 1; i <= 3 ; i++ )
    {
        bosluk( w-i );
        yildiz( i );
        printf("\n");
    }
    return 0;
}
```

```
void yildiz( int x )
{
    while( x > 0 )
    {
        printf("*");
        x--;
    }
}
```

```
void bosluk ( int x )
{
    int b = 1;
    do
    {
        printf(" "); //1 bosluk
        b++;
    }
    while( b <= x );
}
```

/* ciktida bos kalacak hucreleri bos birakiniz. Bu hucrelere 'b', '-' gibi karakterler yazmayiniz.*/


```
#include <stdio.h>
int t( int a );
int main()
{
    int a = 3;
    t( 19 );
    printf("%d", a);
    printf("%d", t( 7 ) );
    return 0;
}

int t( int a )
{
    if(a > 10)
    {
        a++;
        printf("%d", a);
    }
    else if ( a % 2 == 0 )
    {
        a--;
        printf("%d", a);
    }
    else
        printf("%d", a);
    if( a<30 )
    {
        a = 4;
        printf("%d", a);
    }
    return a;
}
```


GERİ BİLDİRİM 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 puan yakın olması durumunda ekstra 1 puan alırsınız. (Bu kısımdan alınabilecek 1 puanı dâhil etmeden sayı girişinde bulununuz. Mantıklı olan aşağıdaki boşluğu [5-95] aralığında bir değerle doldurmaktır.)

Sayı = _____ ± 5

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
ARA SINAV I



Sınav Süresi:120 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 telefonuyla 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 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) (10 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 örnek ekran çıktılarındaki görüldüğü gibi kullanıcıdan **bir sözcük** alacaktır.
 - Kullanıcı **tek sözcüklük bir metin** girip enter'a basacaktır.
 - Kullanıcı maksimum **69 harfli bir sözcük** girecektir. Maksimum (69) harfli giriş yapıldığında kodunuz bu veriyi saklamak için **gereksiz yere büyük değişken** kullanmamalıdır; kullanıcı daha az harfli (örn. 8 harfli) bir giriş yaptığında değişkeninizin bir kısmının kullanılmıyor olması normaldir.
 - Girişteki harflerin, **İngiliz alfabesindeki 52 harf** ([a-z] arası 26 küçük harf ve [A-Z] arası 26 BÜYÜK harf) arasından seçileceğini ve bu 52 harften başka harfin girişte (ğ, ş gibi) olmayacağını varsayınız. Kullanıcının her zaman geçerli girişler yapacağını (sayı, boşluk, cümle vs. girmeyeceğini) varsayınız.
- Kod, bu sözcükteki **BÜYÜK harf ([A-Z] arası harf) sayısını** örnek ekran çıktılarındaki gibi ekrana yazdıracaktır.
- Kod, bu sözcükteki küçük harflerin karşılık gelen BÜYÜK harflerle (örn. 'd' ise 'D') , BÜYÜK harflerinse karşılık gelen küçük harflerle (örn. 'R' ise 'r') **değiştirilmiş şeklini**, örnek ekran çıktılarındaki gibi, ekrana yazdıracaktır.
- Kodunuzun **return değeri 0** olmalıdır.

```
Giris: gArGaMeL
Buyuk harf sayisi: 5
Yeni sekli: GaRgAmel
-----
Process exited after 23.67 seconds with return value 0
Devam etmek için bir tuşa basın . . .
```

ÖRNEK EKRAN ÇIKTISI 1

Kod çalıştıktan sonra kullanıcı "gArGaMeL" girişi yapıp enter'a basmıştır. "A-G-M-E-L" olarak 5 tane BÜYÜK harf olduğu için bu belirtilmiştir. Sonra da yeni şekli 'g' yerine 'G', 'A' yerine 'a', 'r' yerine 'R', 'G' yerine 'g'... şekliyle ekrana yazdırılmıştır.

```
Giris: CEKoSLAVakyalilastiraMAdiklariMIZdanMIScasina
Buyuk harf sayisi: 15
Yeni sekli: cekOslavAKYALILASTIRAmADIKLARImizDANmisCASINA
-----
Process exited after 2.416 seconds with return value 0
Devam etmek için bir tuşa basın . . .
```

ÖRNEK EKRAN ÇIKTISI 2

Kod çalıştıktan sonra kullanıcı uzun bir sözcük girişi yapıp enter'a basmıştır. "C-E-K-S-L-A-V-M-A-M-I-Z-M-I-S" olarak 15 tane BÜYÜK harf olduğu için bu belirtilmiştir. Sonra da yeni şekli 'C' yerine 'c', 'E' yerine 'e', 'K' yerine 'k', 'o' yerine 'O'... şekliyle ekrana yazdırılmıştır.

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.
- Benzer şekilde; **kütüphane eklenmesi** gibi olgulara dikkat etmeden yazdığınız kodlarda da kodunuz doğru çalışsa bile puan kaybı yaşanabilir. Örneğin, kodunuzda gereksiz yere eklenmiş kütüphane bulunmasın.
- Kod içerisinde **boşluk** bırakmanın kritik olduğu kısımlardaki boşluklarınız **açık olmalı**. Gerekirse ilgili satıra yorum ekleyerek bunu belirtiniz.
- Bu soruyu çözebilmek için **ASCII tablosunu** ezbere bilmeye **gerek yoktur**.
- C dilinin **char** ve **int** arasındaki **tür dönüşümleri**ni arka planda başarılı şekilde yapabildiğini varsayabilirsiniz.
- Kodunuzu en baştan **girintilere** özen göstererek yazarsanız, algoritmanızı daha **okunur** kılarak olası hatalardan ({ } unutmak gibi) kolaylıkla kaçınırsınız. Kodlarda derleme imkanı olmadığı için rahatlıkla ; gibi **detaylar** unutabilmektedir. Ayrıca, **kötü/hızlı yazımdan** kaynaklı (t gibi gözüken f harfleri, ı 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**.

1.SORUNUN DEĞERLENDİRİLMESİ

<u>Hata Puanı</u> <u>(HP)</u>	<u>Temel Kısımdan</u> <u>Alınan Puan</u>
<u>0-3</u>	<u>10/10</u>
<u>4</u>	<u>9/10</u>
<u>5</u>	<u>7/10</u>
<u>6</u>	<u>5/10</u>
<u>7</u>	<u>2/10</u>
<u>8 ve üstü</u>	<u>1/10</u>
<u>Çok zayıf ya da</u> <u>boş</u>	<u>0/10</u>

Küçük Hatalar 1 HP üretirler. Tipik örnekleri:

- & unutulmuş.
- if(x==2) yerine if(x=2) ifadesi kullanılmış.

Orta Hatalar 2 HP üretirler. Tipik örnekleri:

- || yerine && yazılmış.
- return 0; yok

Büyük Hatalar 3 HP üretirler. Tipik örnekleri:

- Dizginin tüm elemanlarını ('\0' sonrasındakileri de) kapsama almak

Kod yazma sorularında **parçalı** puanlandırma

- Kod yazma sorularında parçalı puanlandırma yapılır.
 - Öğrencinin bilgisayarda kod yazmadığı, kağıt üzerine kod yazdığı göz önünde bulundurulur.
 - Her hatadan puan kırılmayabilir. Örneğin, toplamda bir iki ufak hatası (*örneğin bir yerde ; unutmak*) olan öğrenci tam ya da tama yakın puan alabilir.
- Değerlendirmeye dair daha detaylı bilgi de sizinle bu sunuda paylaşılmaktadır.

Kod yazma sorularında **parçalı** puanlandırma Detaylı anlatım

Küçük Hatalar 1 HP üretirler. Tipik örnekleri:

- `char cevap[20];` denilmiş, ancak 20 değeri örnekte verilen dizgi için yetersiz.
- Tamponda kalan `\n`'nin bir sonraki `scanf/gets`'e sızmasına önlem alınmamış.
- `{` ya da `"` ya da `;` gibi eksik temel unsurlardan her biri için
- `stdio.h` kütüphanesi eklenmemiş.
- Gereksiz kütüphane eklenmiş.
- `exit(0)` kullanılmış ama `stdlib.h` eklenmemiş.

Orta Hatalar 2 HP üretirler. Tipik örnekleri:

- `char cevap[];` denilmiş, kapasite belirtilmemiş.
- `if(!strcmp...)` ifadesindeki `!` yok.
- `return 0;` konulmadığı için kod çıkması gerektiği yerden çıkmıyor.
- `for`'un her turunda hem `i`'yi artırıyor, hem hak değeri azaltıyor.

Büyük Hatalar 3 HP üretirler. Tipik örnekleri:

- `char cevap[100];` yerine `int cevap;` yazılmış;
- `strcmp` ile dizginin karakterlerini tek tek kontrol edilmeye kalkılmış.
- `while` kullanması gereken yerde `if` kullanılmış.

Toplam Hata Puanı (HP)	Temel Kısımdan Alınan Puan
0	10/10
1	10/10
2	10/10
3	9/10
4	7/10
5	4/10
6 ve üstü	1/10
Çok zayıf ya da boş	0/10

• Soru karmaşıklık seviyesi, öğrencinin kodunu derleme imkanı olmadığı göz önünde bulundurularak belirlenir.

• Toplamda 1-2 küçük hatadan puan kırılmadığına dikkat ediniz. Ancak bu hataların küçük olması ve çok olmaması istenir.

Bu slayttaki değerler ve tipik örnekler fikir vermesi açısından paylaşılmıştır, değişkenlik gösterebilir. Bonus kısım varsa, o kısma bu yöntem, ilgili hatalar ayrıca (HBP – hata bonus puanı) işaretlenerek ayrı bir HBP-Bonus Kısımdan Alınan Puan eşleşme tablosuyla uygulanır.

// BIL141 DERSİ SINAVINDAKİ 1.SORU KODUM

#include <stdio.h>

int main()

{

char giris[70];

printf("Giris: ");

scanf("%s", giris);

int i, buyukHarf=0;

for (i=0; giris[i] != '\0'; i++)

{

if ('a' <= giris[i] && giris[i] <= 'z')

giris[i] -= ('a' - 'A');

else

{

giris[i] += ('a' - 'A');

buyukHarf++;

}

}

printf("Buyuk harf sayisi: %d\n", buyukHarf);

printf("Yeni sekli: %s", giris);

return 0;

}

Cevabınızı en
ŞİRİN YAZINIZLA,
girintilere özen
göstererek ve
dikkatlice bu şirin
sayfaya yazınız.



Kısa ve sade bir
şirin **SÖZDE KOD**
ile algoritmanızı
oturtuktan sonra
koda başlarsanız
daha başarılı
olabilirsiniz.

$i < 70$ olmaz çünkü
'\0' durumunda
sonrasındaki karakterleri de
dahil eder.

$97 - 65 = 32$

yazmaya
dettir.
Böyle yazarak
ezbeden
kaçınmış oluruz.
Daha rahat
anlaşılır bir
kod olur.

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 sınav kağıdına yazdığınız 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: Merhaba Şirin Baba. Babalar günün kutlu olsun. Şirin sıcak bir ay... yine ben... ve yine programlama dersi ile ilgili sorularım.

Şirin Baba: Sağ ol şirinim. Ben de az önce düzenli ders çalışmayan seni andım. İçimden "Usta'nın düzenli çalışacağı yok, dersin hocası değişmişken karambolde belki şirin vasat bir notla dersi geçiverir." diyordum.

Usta Şirin: Yok Şirin Baba, o şekilde bir değişim değilmiş; baksana her şey aynı, yine burada seninle sınav kağıdının ortasında soru-cevap yapıyoruz. Ama zaten ben bu dönem düzenli çalışıp şirin bir not ile dersi geçeceğim. Bu ders şirin geleceğim için bana gerekli.

Şirin Baba: Peki, Şirinim. Başarılar sana sınavında. Ne sorular getirdin bakalım?

Usta Şirin: Geçen gün Şirine'ye "beni İnşiringam'ına ekler misin?" dedim, "Bilmem ki, bir kod yaz, kod karar versin." dedi. Gerekli kütüphaneleri, komutları vs. ekledikten sonra %50 ihtimalle "Ekle" diyen şu kodu yazmak istedim ama ???????? yazdığım yere ne yazmam gerek?

```
if( rand() ??????? )
    printf("Ekle");
```

```
else
    printf("Uzak dur");
```

C dilinde bağlamanın mantıksal karşılığı old. için sadece %2 de yeterlidir.

Şirin Baba: ???????? kısmını silip, oraya %2 yazman yeterli olur.

Usta Şirin: Yaptık ve "Ekle" çıktı!!! Şimdi teşekkür olarak ormandan beşli böğürtlen paketleri göndereceğim.

[50-80] aralığında (50 ve 80 dahil) 5'in katı olan sayı üreten ifade nasıldır?

Şirin Baba: Şirin bilgi yarışması ödevini yapmadın mı yoksa? $(P + rand() \% Q) * R$ şeklinde bir ifade olabilir. Ama P, Q, R değişkenlerinin sayı değerlerini sen bulmalısın. Ben sadece sana şu ipucunu vereyim. P, Q ve R sayılarını

çarpınca 350 eder.

R → 5 olmalı ki 5'in katı olsun.
Minimum P, R üretilir. P 10 olsun ki P.R 50 olsun.
rand() % Q maks. 6 üretsin diye Q 7 olmalı.
 $P \cdot Q \cdot R = \frac{5}{2} \cdot \frac{7}{3} \cdot 10 = 350$

Usta Şirin: Kullanıcı klavyeden bir harfe bastığında, bastığı harf ekranda gözükmesin, peşinden enter'a basmasına gerek kalmasın ve ben hemen girilen harfin ASCII değerini fonksiyonun döndüğü değer olarak alabileyim istiyorum. Hangi fonksiyonu önerirsin?

getch() ekranda göstermez.

Gargamel: *conio.h* kütüphanesindeki *getche* fonksiyonu tam olarak aradığın fonksiyondur.

Usta Şirin: `int array[5];` dedikten sonra dizideki son elemana 21 sayısını kaydetmek istiyorum. Ancak `array[5] = 21;` yazdığımda hata veriyor. Doğrusu nedir?

Gargamel: Bu dizinin son elemanına kayıt yapabilmek için kodunda `array[5] = 21;` ifadesini kullanman gerekir.

doğrusu array[4]=21;

Usta Şirin: Kullanıcıdan bir karakter almak istiyorum. `char karakter;` tanımını yaptım ve peşinden `scanf("%c", karakter);` şeklindeki kod parçasıyla aldığım değeri `karakter`'e kaydetmeye çalıştım. Ancak kodum çalıştıktan sonra, deneme amaçlı 1 girip enter'a basınca program çöküyor, neden?

Gargamel: `%c` karakter almak içindir ama sen denemek için rakam girmişsin. Rakam almak için `%d` kullanmalıydın.

rakam da bir karakterdir. Burada & eksikliği var.

Usta Şirin: Kodumda `main` içerisinde `char sirine[100];` şeklinde bir dizgi tanımladım. `ilanEt` isimli aşağıdaki gibi bir fonksiyonum var. Fonksiyonumun prototipini de `void ilanEt( char );` olarak `main`'in başına koydum.

```
void ilanEt( char sirine )
{
    printf("Usta kalp %c ", sirine);
    sirine = 'S';
    return;
}
```

Şirin Baba, sana sorum şu şekilde: `ilanEt` fonksiyonunu `main`'den `ilanEt ( sirine[0] );` şeklinde çağırırsam hatasız şekilde çalışır mı?

Gargamel: Çalışmaz, çünkü sen `char` değişkeni göndereceğin yerde `main`'deki `sirine` dizgisini gönderiyorsun.

Çalışır → dizinin ilk elemanını gönderir. Yani char gönderir.

Usta Şirin: Peki, doğru şekilde çağırıldığında fonksiyon içerisinde yapılan 'S' değer ataması `main`'de de geçerli olur mu?

Şirin Baba: Hayır, gönderdiğin değişkenin fonksiyonda kopyası çıkarıldığı için değer atamasının `main`'e etkisi olmaz.

char değil de diziyi gönderseydi 'adres ile gönderim' olurdu.

Usta Şirin: Daldan dala atlıyorum ama kusura bakma. *main* içerisinde şöyle bir kod yazdım.

```
FILE* araba= fopen("veri.txt", "w");
```

```
fprintf(araba, "12");
```

```
// ????????
```

```
araba= fopen("veri.txt", "r");
```

```
int x = -1;
```

```
fscanf(araba, "%d", &x);
```

```
printf("%d", x);
```

buna bekletiyor. ilk gardiyanı
• fflush
• fclose
• kod bitiminde dosyaya
döküyor.

Çalıştırdığımda ekrana 12 yazmasını beklerken -1 yazdırıyor. Nasıl düzelir acaba?

Şirin Baba: // ???????? ifadesini silip, onun yerine fclose(araba); yazarsan düzelir.

Usta Şirin: Uygun kütüphaneleri vs. ekledikten sonra *main* içerisine

↳ fflush(araba); ✓
fclose(dosya); X

```
if( sayi > 10 )
```

```
printf("İki basamaklı");
```

```
else sayi < 10 )
```

```
printf("Tek basamaklı");
```

if gerekli.

yazdım. Kodum, ekrana *sayi*'nin basamak adedini yazsın istiyorum. *sayi* değişkeni [1-99] arasında değer içeren bir değişkendir. Bu kod parçası doğru çalışır mı?

Gargamel: İki basamaklı olan 10'u da dahil etmek üzere >10 ifadesi yerine >=10 ifadesi yazarsan kod istediğin gibi doğru çalışacaktır.

Usta Şirin: Herhangi bir fonksiyon içinde tanımlanmamış değişkenlere ne denir? Baş harfi k idi ama gerisini hatırlayamadım.

kötü, kanıtlanmamış, kağıncası

Şirin Baba: Fonksiyonların dışında tanımlanmış bu değişkenlere C dilinde ve Türkçe olarak küresel global değişken denir. ŞirinC'de *gargamel* değişken olarak da geçer.

Usta Şirin: Unutkan Şirin gelmiş, *istGelin* dizisindeki [1][1] indisli elemanının değerini unuttum diyor. Dizinin tanım satırı şu şekilde: *int istGelin*[2][3] = { { 74, 233, 667 } , { 93, 46, 142 } };

istGelin[0][0] istGelin[1][1]
46

Şirin Baba: Bu tanım ve değer atama işlemine göre *istGelin*[1][1], 46 değerine sahiptir.

Unutmamak için çok çalışmak gerekiyor.

Usta Şirin: Blok içerisinde *atanan* blok içinde kalır, diye hatırlıyorum. Uygun şekilde kütüphaneler vs. eklendiğinde, main içerisine yerleştirilen aşağıdaki kod parçası ekrana 7 yazdırması lazım, değil mi?

```
int azman = 7;
{
    azman = 8;
    int azman;
    azman = 9;
}
printf("%d", azman);
```

Şirin Baba : Hayır, öyle bir şey yoktur; doğrusu "*Blok içinde tanımlanan blok içinde kalır*" şeklindedir.

Bu kod ekrana 8 yazdıracaktır.

Usta Şirin: Peki, *kabarcık sıralama* algoritmasını 64 elemanlı bir diziye uyguluyorum. Baştan sona doğru, çiftler halinde sayıları incelediğim her tura bir *geçiş* diyorum. En az kaç *geçiş* sonrasında "*dizi kesinlikle sıralanmıştır*" diyebiliyoruz?

Şirin Baba: 64 elemanlı bir diziye kesinlikle sıralı hale getirmek için en az 63 *geçiş* gerekir.

Usta Şirin: `scanf("%d", &sayi);` şeklindeki bir kod parçasını içeren bir kodum var. Kod çalışırken bu satıra gelindiğinde, klavyeden "a" harfini yazıp enter'a basarsam ne olur?

Gargamel : Harflerin de C dilinde sayısal karşılıkları olduğu için, "a" harfinin ASCII değeri *sayi* değişkenine kaydedilir. *→ scanf eli boş döner. sayi eski değeri neyse o olur.*

Usta Şirin: Demek ki *sayi*, şirin 97 değerini alır diyorsun. Tamamdır. İçinde "*tutku*" olan bir karakter dizisi istiyorum. `char dizi[] = { t, u, t, k, u };` şeklinde yapsam "*tutku*" ifadesindeki harfleri dizide saklamış olur muyum?

Şirin Baba : Bir tek harfleri tek tırnak içerisinde (örn. 't') alman eksik olmuş. Onu da yaparsan başarılı bir şekilde saklamış olursun.

Usta Şirin: Geçen gün, Uykucu Şirin gelmiş, "*Geçen uzanmış uyanmaya çalışırken aklıma takıldı: [15-X] (15 ve X dahil) aralığında bir sayı tutacak olayım, sen de bu sayıyı tahmin edecek ol. Ben, senin her tahmininden sonra "çık", "in" ya da "buldun" şeklinde seni yönlendireceğim. Senin 7 tahmin hakkın var ve optimum algoritma ile sayımı bulmaya çalışacaksın. Her durumda sayımı bulabilmen için X sayısı en çok kaç olabilir Usta?*" diye bana soruyor.

Şirin Baba: Optimum algoritma kullanılacaksa X en büyük 141 değerini alabilir Ustacığım. Şu *optimum* sözcüğünü de atalım artık dilimizden, onun özşirincesi *şirinmumdur*.

Usta Şirin: Tamam Şirin Baba. `fprintf(stdout, "Şirince kal.");`

3) (Her boşluk 2 puan - en iyi 16 cevap değerlendirmeye alınır - toplam 32 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, $\leq$ yerine $\leq$ 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.

SÖZDE KOD	SÖZDE EKRAN ÇIKTISI
1. başla 2. <u>Sayac = 0</u> 3. sayi=2 4. Eğer sayi > 300 ise <u>Git 8</u> 5. Eğer sayi mod 10 = 0 ise sayac=sayac+1 6. sayi=sayi+2 7. Git 4 8. Yaz sayac 9. Dur <i>[2, 300) aralığındaki çift sayılardan kaç tanesi 30'un katıdır? "</i>	30 Sözde kodun ekran görüntüsü yukarıdaki şekilde bir sayıdan oluşmaktadır. Soruda boşlukların dışını doldurmayınız. Sadece boşluklara yazdıklarınız değerlendirmeye alınır. Bir boşluğa bir komut yazınız. Gereksiz komut/ifadeler eklemeyiniz. Bu tür durumlarda kodunuz çıktı açısından isteneni yapıyor olsa bile puan alamayabilirsiniz.

KOD - Çalıştığında ekrana "1" yazdıran kod	EKRAN ÇIKTISI
<pre>#include <stdio.h> // iyi ki Odev3'u fonksiyonlu yaptım char harftenRakama (char); int main() { char harf1 = <u>'b'</u>; printf("%<u>C</u>", harftenRakama(harf1)); return 0; } char harftenRakama (char <u>X</u>) { <u>Switch</u> (x) { case 'b': return '1'; default: return '0'; } }</pre>	<pre>1 ----- Process exited after 0.006962 seconds with return value 0 Devam etmek için bir tuşa basın . . .</pre> ÖRNEK EKRAN ÇIKTISI Kod çalıştırıldıktan sonra herhangi bir kullanıcı girişi olmamıştır ve görüldüğü gibi bir ekran çıktısı sağlanmıştır. 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. Soruda boşlukların dışını doldurmayınız. Sadece boşluklara yazdıklarınız değerlendirmeye alınır.

KOD - Kullanıcıdan bir para değeri alan, bu değer 20'den küçük olduğu sürece tekrar değer girişi isteyen, kendisi de [25, 50] arasında 5'in katı olan bir para değeri üreten ve örnek çıktılardaki gibi çalışan kod

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
int main()
{
    srand ( time(NULL) );
    int tl;
    printf("Ne yiyelim kanka?\n\t\tKac liran var?\n");
    scanf("%d", &tl);

    while ( ! (tl>=20) )
    {
        printf("\t\tCok az.\n");
        printf("\t\tTum ceplerine iyice bak.\n");
        printf("\t\tKac liran var?\n");
        scanf("%d", &tl);
    }

    int ben = 25+ rand() % 6 * 5;
    printf("\t\tBende de %d lira var.\n", ben );
    if(tl+ben>=30*2)
        printf("Iki iskender soyleyelim mi?");
    else if (tl+ben<60) X
    {
        printf("Ben ac degilim zaten.");
        printf("\n\t\tBana uyar kanka.");
        return 0;
    }
}
```

EKRAN ÇIKTISI

```
Ne yiyelim kanka?
Kac liran var?
27
Bende de 45 lira var.
Iki iskender soyleyelim mi?
Bana uyar kanka.
```

ÖRNEK EKRAN ÇIKTISI

Kod çalıştırıldıktan kullanıcı "27" yazıp enter'a basmıştır. Kod [25, 50] aralığında ve 5'in katı olan 45 değerini üretmiştir.

```
Ne yiyelim kanka?
Kac liran var?
17
Cok az.
Tum ceplerine iyice bak.
Kac liran var?
19
Cok az.
Tum ceplerine iyice bak.
Kac liran var?
20
Bende de 25 lira var.
Ben ac degilim zaten.
Bana uyar kanka.
```

ÖRNEK EKRAN ÇIKTISI

Kod çalıştırıldıktan kullanıcı sırasıyla "17", "19" ve "20" yazıp her girişten sonra da enter'a basmıştır. İlk iki giriş 20'den küçük olduğu için kod tekrar giriş yapılmasını istemiştir. Kod [25, 50] aralığında ve 5'in katı olan 25 değerini üretmiştir.

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. Soruda boşlukların dışını doldurmayınız. Sadece boşluklara yazdıklarınız değerlendirmeye alınır.

KOD - Çalıştığında ekrana örnek çıktıdaki şekilde çıktı veren kod

```
#include <stdio.h>
/*burada fonk prototipinin dogru sekilde bulunduğunu varsayınız*/
int main()
{
    int a[4] = { 3 , 4 , 5 }, b[4]={ 7 , 9 , 1 , 2 };
    printf("% .21f - % .21f\n", fonk( a ), fonk( b ) );
    int i = 0;
    while( i < 4 )
    {
        printf("%d ", a[ i++ ]);
    }
    return 0;
}

double fonk ( int v[] )
{
    v[1] = 9;
    return v[0] / (double) v[2];
}
```

EKRAN ÇIKTISI

```
0.60-7.00
3 9 5 0
-----
Process exited after 0.009524 seconds
with return value 0
```

ÖRNEK EKRAN ÇIKTISI

Kod çalıştırıldıktan sonra herhangi bir kullanıcı girişi olmamıştır ve görüldüğü gibi bir ekran çıktısı sağlanmıştır.

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.

Soruda boşlukların dışını doldurmayınız. Sadece boşluklara yazdıklarınız değerlendirmeye alınır.


```
#include <stdio.h>
int main()
{
    int i, j;
    for( i = 1 ; i <= 2; i++)
    {
        for( j = 1 ; j <= 4; j++)
        {
            switch( (i+j) % 2 )
            {
                case 1:
                    printf("A");
                    break;
                case 0:
                    printf("B");
                    break;
                default:
                    printf("C");
            }
        }
        printf("\n");
    }
    return 0;
}
```

B	A	B	A																
A	B	A	B																

```
#include <stdio.h>
#define SAYI 8
void fonk(int);
int main()
{
    fonk(SAYI);
    return 0;
}
void fonk(int a)
{
    printf("%5.3d", a);
    return;
}
```

			0	0	8														

5 hane rezerve edildi.

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

```
int main()
```

```
{
```

```
int x = 8;
```

```
do
```

```
{
```

```
x -= 2;
```

```
printf("%d", x);
```

```
}
```

```
while( (x = 0) ); // == degil
```

```
printf("%d", x);
```

```
return 0;
```

```
}
```

6	0																		

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

```
int main()
```

```
{
```

```
int x = 0;
```

```
int y = 9;
```

```
{
```

```
printf("%d", y/x);
```

```
printf("x/y");
```

```
}
```

```
return 0;
```

```
}
```

H	A	T	A	2															

HATA 2 çünkü kod çöker → geçersiz
adddna
Zaten HATA2'nin
anlamı 0.

#include <stdio.h>

void k(int dizi[], int n);

int main()

```
{
    int d[] = {8, 1, 5, 7, 3, 4, 2, 6, 7, 9};
```

int i;

```
/* ipucu: sizeof(d) ifadesi d dizisinin RAMde kac bayt
   yer tuttuğunu verir */
```

int n = sizeof(d) / sizeof(d[0]);

k(d, n);

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

printf("%d", d[i]);

return 0;

```
}
void k( int dizi[], int n )
```

{

int i, t;

int b;

do

{

b = 0;

for(i = 0; i < n-2; i++)

{

if(dizi[i+1] > dizi[i])

{ büyükten küçüğe

t = dizi[i];

dizi[i] = dizi[i+1];

dizi[i+1] = t;

b = 1;

}

}

n--;

}

while(b);

}

8	7	7	6	5	4	3	2	1	9

#include <stdio.h>

#include <string.h>

void g(char [], char []);

int main()

{

char a1[11] = "zebra";

char a2[11] = "hamsi";

g(a1, a2);

printf("%s\n", a1);

printf("%s\n", a2);

return 0;

}

/* asagidaki satirda once a2, sonra a1 yazmaktadır. */

void g(char a2[], char a1[])

{

if(!strcmp(a1, a2)) // ! var.

strcpy(a2, a1);

else

strcat(a1, a2);

return;

}

z	e	b	r	a					
h	a	m	s	i	z	e	b	r	a

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

```
void al(char[];
```

```
int main()
```

```
{
```

```
char g[20] = {};
```

```
printf("G:");
```

```
al(g);
```

```
int i;
```

```
for( i = 0 ; g[i]!='\0' ; i++) ; // var.
```

```
printf("%d\n", i);
```

```
printf("%s", g);
```

```
return 0;
```

```
}
```

```
void al(char h[])
```

```
{
```

```
scanf("%[^2]s", h);
```

```
return;
```

```
}
```

/*Kod çalıştıktan sonra kullanıcı "15 12 42" girmiş ve enter'a basmıştır. enter nedeniyle çıktının bir alt satırdan devam etmesi beklenir. Çıktının en son hali istenmektedir. Çıktının bir kısmı verilmiştir, geriye kalanını sizin tamamlamanız istenmektedir.*/

G	:	1	5		1	2		4	2
4									
1	5		1						

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

```
int sunger(int bob);
```

```
int kare(int sort);
```

```
int main()
```

```
{
```

```
printf("%d", sunger(5));
```

```
printf("%d", kare(8));
```

```
return 0;
```

```
}
```

```
int sunger(int bob)
```

```
{
```

```
printf("%d", bob);
```

```
return bob;
```

```
}
```

```
int kare(int sort)
```

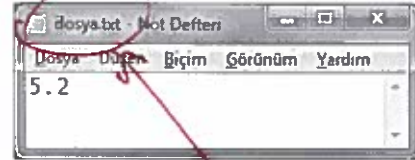
```
{
```

```
sort = sort + bob;
```

```
return sort;
```

```
}
```

H	A	T	A	1					



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

```
int main()
```

```
{
```

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

```
char x = (klasor!=NULL)?'a':'b';
```

```
double y;
```

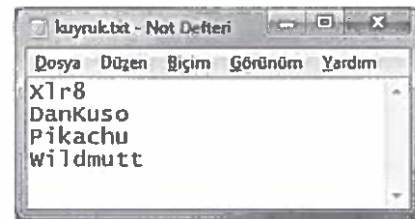
```
fscanf(klasor, "%lf", &y);
```

```
printf("%c%d", x, (int)y*10);
```

```
return 0;
```

```
}
```

2	5	0							



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

```
int main()
```

```
{
```

```
char sir[20];
```

```
int i;
```

```
FILE* queue = fopen("kuyruk.txt", "r");
```

```
for( i = 1; i < 9; i += 2)
```

```
{
```

```
if(i==3) continue;
```

```
fscanf(queue, "%s", sir);
```

```
printf("%s\n", sir);
```

```
}
```

```
return 0;
```

```
}
```

X	l	r	8						
D	a	n	K	u	s	o			
P	i	k	a	c	h	u			

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

```
int main()
```

```
{
```

```
    int dizi[7] = {};
```

```
    int i;
```

```
    for(i=0; i<7; i++)
```

```
        dizi[i] = (i*i);
```

```
        printf("%d\n", i);
```

```
    for( i=0 ; i!=9 ; i+=3 )
```

```
        printf("%d", dizi[i]);
```

```
    // printf'lerden sadece birinde \n var.
```

```
    return 0;
```

```
}
```

7									
0	9	3	6						

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

```
#include <math.h>
```

```
double k1 (int a);
```

```
double k2 (double b);
```

```
int main()
```

```
{
```

```
    printf("%.2lf\n", k1( 36 ) );
```

```
    printf("%.2lf\n", k2( 64 ) );
```

```
    return 0;
```

```
}
```

```
double k1 (int a)
```

```
{
```

```
    double b = (double)a;
```

```
    return pow( b, (double)( 1/2 ) );
```

```
}
```

```
double k2 (double b)
```

```
{
```

```
    return pow( b, (double)1/2 );
```

```
}
```

1	.	0	0						
8	.	0	0						

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

```
int main()
```

```
{
```

```
    int a=4;
```

```
    while( a<2 ); // ; var
```

```
{
```

```
        printf("%d", a);
```

```
        a++;
```

```
}
```

```
    printf("%d", a);
```

```
    return 0;
```

```
}
```

4	5								

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

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
    int x = getch() - (int)'0';
```

```
    int y = getch() - (int)'0';
```

```
    //2.si getch'e degil
```

```
    if( x>5 )
```

```
        printf("A");
```

```
    else if( y>x )
```

```
        printf("B");
```

```
    else if( x>0 )
```

```
        printf("C");
```

```
    return 0;
```

```
}
```

/*Kod çalıştıktan sonra kullanıcı "3" ve "4" tuşlarına sırasıyla basmıştır. Çıktının tamamı ve en son hali istenmektedir.*/

3	B								

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

```
void yildiz ( int );
```

```
void bosluk ( int );
```

```
int main()
```

```
{
```

```
int w = 3;
```

```
int i;
```

```
for( i = 1; i <= 3; i++)
```

```
{
```

```
bosluk( w-1);
```

```
yildiz( i );
```

```
printf("\n");
```

```
}
```

```
return 0;
```

```
}
```

```
void yildiz( int x )
```

```
{
```

```
while( x > 0 )
```

```
{
```

```
printf("*");
```

```
x--;
```

```
}
```

```
}
```

```
void bosluk ( int x )
```

```
{
```

```
int b = 1;
```

```
do
```

```
{
```

```
printf(" "); //1 bosluk
```

```
b++;
```

```
}
```

```
while( b <= x );
```

```
}
```

```
/* ciktida bos kalacak hucreleri bos birakiniz. Bu
```

```
hucrelere 'b', '-' gibi karakterler yazmayiniz.*/
```

		*							
	*	*							
	*	*	*						

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

```
int t( int a );
```

```
int main()
```

```
{
```

```
int a = 3;
```

```
t( 19 );
```

```
printf("%d", a);
```

```
printf("%d", t( 7 ) );
```

```
return 0;
```

```
}
```

```
int t( int a )
```

```
{
```

```
if( a > 10 )
```

```
{
```

```
a++;
```

```
printf("%d", a);
```

```
}
```

```
else if( a % 2 == 0 )
```

```
{
```

```
a--;
```

```
printf("%d", a);
```

```
}
```

```
else
```

```
printf("%d", a);
```

```
if( a < 30 )
```

```
{
```

```
a = 4;
```

```
printf("%d", a);
```

```
}
```

```
return a;
```

```
}
```

2	0	4	3	7	4	4			

GERİ BİLDİRİM 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 puan yakın olması durumunda ekstra 1 puan alırsınız. (Bu kısımdan alınabilecek 1 puanı dâhil etmeden sayı girişinde bulununuz. Mantıklı olan aşağıdaki boşluğu [5-95] aralığında bir değerle doldurmaktır.)

Sayı = _____ ± 5

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?