

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.

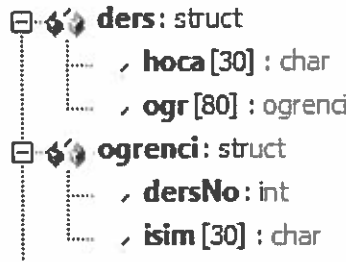
İmza:

1) (10 puan) Verilen koşulları 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.

Bir dersi veren hocanın ismi ve bu dersi alan 80 öğrenciye atanan ders numaraları bilgisayara aktarılacaktır. Kodunuzda aşağıdakileri dikkatlice gerçekleştirmeniz istenmektedir.

Kodda Şekil-1'de gözüktüğü gibi iki tane yapı tanımı olmalıdır. Şekil 1 DevCpp'daki Project/Class Browser (Proje/Sınıf Gezgini) panelinden elde edilmiştir. Sizin kodunuz da DevCpp'a aktarılacak olursa Şekil 1'dekiyle **tamamen aynı** görünümün(aynı isimlerin, aynı sayıların ve diğer detayların) elde edilebiliyor olması istenmektedir.



Şekil 1 - Kodunuzda bulunması istenen yapılar
(DevCpp Project/Class Browser altındaki görünüm)

main'de *ders* yapısından bir değişken tanımlamanız ve bu yapıya şu bilgileri yerleştirmeniz istenmektedir:

- *hoca* bilgisi "*Sirin Baba*" olarak atanacaktır.
- öğrencilere isim atanmayacaktır; ancak en düşük indisli öğrenci 101 ders numarasına sahip olmak üzere, indis arttıkça ders numaraları birer birer artarak öğrencilere atanacaktır. Bu nedenle son öğrencinin (en büyük indisli öğrenci) ders numarası 180 olmalıdır.

Kod çalıştığında ekrana **bir şey yazdırmaz** ve "**return value**" olarak 141 değerini dönerek sonlanır. Kodun görevi sadece RAM'e bu veriyi işlemek ve istenen şekilde sonlanmaktır. Kod sonlandığında verinin yok olacak olması, bu soru açısından bir sakınca oluşturmamaktadır.

Kod gereksiz hiçbir kütüphane içermemelidir.

ÖNEMLİ UYARILAR:

- Detaylandırılmış uyarı ayarları açıkken **hiçbir uyarı ya da hata vermeyecek** bir kod yazmaya çalışınız.
- 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; **tür dönüşümü, kütüphane eklenmesi** gibi olgulara dikkat etmeden yazdığınız kodlarda da kodunuz doğru çalışsa bile puan kaybı yaşanabilir. Örneğin, tür dönüşümü yapılan her yerde bunu kodunuzda açıkça belirtin; bunu derleyicinin "hoşgörüsü"ne bırakmayın.
- Kod içerisinde **boşluk bırakmanın** kritik olduğu kısımlardaki boşluklarınız **açık olmalı**. Gerekirse ilgili satıra yorum ekleyerek bunu belirtiniz.
- 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 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**.

// BİL141 DERSİ SINAVINDAKİ 1.SORU KODUM

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



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

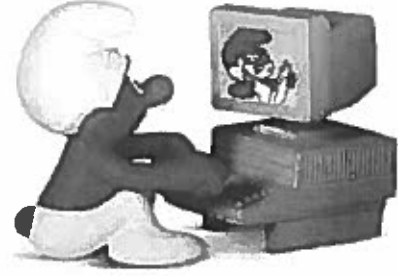
`printf("Ornek);` komutunda " eksik olması gibi,

`<=` 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: Merhaba Şirin Baba. 141 numaralı öğrenciyim. Sana birkaç sorum olacak.

Şirin Baba: Sor bakalım.

Usta Şirin: Kodumda `FILE* gozluklu = fopen("data.txt", "w");` ile dosya açıp, `fprintf` ile bu dosyaya değer yazdırıyorum, kodu 10 saniye bekletiyorum, sonra `fprintf` ile başka bir değer daha yazdırıp, 10 saniye daha bekletiyorum. Ardından da kod sonlanıyor. Ancak bu 10 saniyeler içinde dosyayı açıp içine baktığımda `fprintf`'lerin etki etmediğini görüyorum, dosyanın içi boş gözüküyor, neden?

Şirin Baba: Daha kaç kez söylemem gerek bilemiyorum. `fprintf` komutu kodda çalıştırılınca anında dosyaya yazı yazmaz. Sana `fclose` önerirdim ama ilk 10 saniyeden sonra da dosyayla işin devam edeceği için dosyayı tekrar açman gerekir ki hoş olmaz. O halde kodundaki

her `fprintf`'ten hemen sonraya _____; şeklinde bir komut ekle. Bunu her iki `fprintf`'in için de ayrı ayrı uygula.

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

```
typedef struct
{
    char sanzuman;
}vehicle;
```

Şimdi `main`'in içerisinde

```
vehicle.sanzuman = 'A';
printf("%s", vehicle.sanzuman);
```

yazdığım kod hata veriyor.

_____: %s değil de %c yazarsan sorun çözülür; çünkü `vehicle.sanzuman` değişkeni bir dizgi değildir, sade `char` türündedir.

Usta Şirin: feof fonksiyonu ne işe yarar?

_____ : Bir dosyayı *fopen* ile açmak istediğinde eğer o dosya yoksa *feof* sana 0 harici bir değer döner, aksi durumda 0 döner. Böylece dosyanın mevcut olup olmadığını anlamana ve gerekiyorsa kodu sonlandırmaya yarar.

Usta Şirin: Bilgisayarımda DevCpp artık açılmamaya başladı. Bir türlü düzeltemiyorum. Alternatif bir program mı yüklemeliyim, yoksa yüklemeyen de kodumun derlenmesini sağlayabilir miyim?

_____ : Evet, DevCpp çalışmasa bile ve alternatif bir program yüklemeksizin kodunu derletmen mümkündür.

Usta Şirin: Elimde bir cümle var ve ben bunu sözcüklere ayırmak istiyorum. Sözcükler arasında nokta, virgül ya da boşluk olabilir. Bunu nasıl yaparım?

Şirin Baba: Bunu yapmak için, _____ fonksiyonu en uygun fonksiyonlardan biridir. *string.h* kütüphanesinde. İstedğin ayraçları parametre olarak alarak cümleyi sözcüklere ayırmana yardımcı olur.

Usta Şirin: Fiboşirin sayıları 1,1,2,3,5 ... şeklinde her biri kendisinden önceki iki sayının toplamı olacak şekilde ilerleyen sayılarmış. 41. fiboşirin sayısını hesaplayan bir kod yazmak istiyorum. İşlemciyi yormamak açısından özyinelemeli şekilde kodlamak mı daha uygun olur, yoksa bir dizi ve bir for döngüsü ile mi yapayım?

_____ : Özyinelemeli olarak yapman uygun olur; çünkü özyinelemeli kodlama kodunu birkaç satıra indirgeyecek ve böylece kodunun işlemciyi daha az yormasını sağlayacaktır.

Usta Şirin: `char dizgi[100]; scanf("%s", dizgi);` şeklinde kullanıcıdan bir giriş alıyorum. "eğer bu giriş 'A' karakteri ise" ifadesine ihtiyacım var.

_____ : `if ( strcmp(dizgi[0], 'A' ) )` işini görür. Ancak kodun başına "*string.h*" kütüphanesini eklemiş olmalısın.

Usta Şirin: `char isim[ ? ] = { 'S', 'i', 'r', 'i', 'n', 'e' };` şeklinde bir dizi tanımlıyorum. ? yerinde bir adet sayı var. Bu sayının minimum değeri ne olmalıdır?

Şirin Baba: 6 adet elemanın bu diziye sığabilmesi için ? yerine en az _____ sayısını yazmalısın.

Usta Şirin: Bir dizgi içerisinde kullanıcıdan aldığım ay değeri(örn. "*Mart*") var. Ben bunu sayıya (Mart ise 3, Nisan ise 4 gibi) çevirmek istiyorum. *enum* bu iş için uygun mudur?

_____ : Hayır, *enum* sadece kod içerisindeki *Mart* şeklindeki ifadeleri 3'e vs. çevirebilir. Kullanıcıdan aldığın ay değerini kapsamaz.

Usta Şirin: 18->B diye bir ifade gördüm. Yandan bakınca gözlüklü şirinin papyon takmış haline benzettim.

_____ : Bu ifadeden 18'in kesinlikle bir işaretçi olduğu anlaşılıyor. 18'in işaret ettiği yapı değişkeninin içindeki B değişkenine erişim sağlanmış.

Usta Şirin: Yapı değişkeni demişken elimde bellekte(RAM)'de 10MB yer tutan kocaman bir yapı değişkeni var. Bunu bir fonksiyona göndermek istiyorum. Fonksiyon bu değişkenin değerini değiştirmeyecek; sadece değişkendeki verileri ekrana yazdıracak. Belleği de verimli kullanmak istiyorum. main içerisinde fonksiyonu çağırırken fonksiyona bu değişkeni nasıl göndermeliyim? Fonksiyonun ismi *fonk*, değişkenin ismi *sirinkoy*.

_____ : Fonksiyonu *fonk (&sirinkoy);* şeklinde çağırarak değişkeni adres ile göndermen uygun olur.

Usta Şirin: Bir dizgi tanımlamak için tek yol *char dizgi[100];* yazmak mıdır? *char* dizgi* şeklinde de dizgi tanımlayabilir miyim?

Şirin Baba: 1.sorunun cevabı: Hayır, alternatifi de vardır. 2.sorunun cevabı: Evet, tanımlayabilirsin; ancak

peşinden _____ fonksiyonu ile bellekten 100 baytlık bir yer ayırt ettirmen ve bu yerin ilk baytının adresini dizgiye ataman gerekir. Sana sadece fonksiyonun ismini yazdım; sözdizimini şirinetten öğrenebilirsin.

Usta Şirin: Kullanıcı klavyeden "*Uykucu Şirin*" şeklinde bir giriş yapacak ve ben bunu *char sirinlsim[100];* şeklinde tanımlı bir dizgiye kaydetmek istiyorum.

_____ : *scanf("%s", sirinlsim);* işini görür. Dizgi olduğu için & koymuyorsun.

Usta Şirin: Gözlüklü şirine sordum, "*Benim gözüm zaten bozuk, sen bunu Şirin Baba'ya sor*" dedi. *int dizi[3][5][7][10] = { { { {1,2}, {3,4} }, { {5,6}, {7,8,9} } }, { { {10}, {11},{12,13,14} }, { {15,16}, {17,18,19,20} } } };* dizisini tanımladım. *dizi[1][0][2][1]*'in değeri nedir?

Şirin Baba: Şirin Baba başka ne işe yarar! Dur bakayım, ben de merak ettim...

Şirin Baba: Tamam buldum. Bu tanımla ve atamayla *dizi[1][0][2][1]*'in değeri _____ olmuştur.

Usta Şirin: Peki ben bu dizi elemanını, *main*'den *pembe* isimli bir fonksiyona *pembe(dizi[1][0][2][1])* şeklinde göndersem fonksiyonda yapılan değişiklik kalıcı olur mu?

_____ : Evet şirininim. Dizilerde *adres ile gönderim* söz konusu olduğu için dizinin elemanı üzerinden yapılacak bir değişiklik *main*'de de kalıcı olur.

Usta Şirin: [22, 88] (22 ve 88 dâhil) aralığında 11'in katı olan rastgele bir sayı üretmek ve bunu şirin değişkenime kaydetmek istiyorum. *int sirin_degisken = (rand()%A + B) * 11;* diye bir kod kurguladım ve kodumda kütüphane eklemek gibi ilgili düzenlemeleri yaptım. Bir tek A ve B sayılarını bulmak kaldı. Bir ipucu verir misin?

Şirin Baba: Kurguladığın koddaki A sayısından B sayısını çıkarınca _____ kalır.

Usta Şirin: Geçen gün, Güçlü Şirin gelmiş "[312-634] aralığında bir sayı tuttum...

Şirin Baba: "Aman!" diyeyim Usta Şirin...

Usta Şirin: Bak dinle lütfen Şirin Baba. Anlatıyorum: Gelmiş diyor ki "Sen tahmin edeceksin, bense her tahmininden sonra dambıl gibi 'çık', 'in'...

Şirin Baba: Ben çıkıyorum.

Usta Şirin: in Şirin Baba, aman... dur Şirin Baba. Bak öyle değil. Ben de senin gibi sıkıldım. Dedim "Yeter artık, sayı mayı tutma Güçlü Şirin. Her sınavdan önce biri gelip, çık, in, bu nedir böyle?". Derste yazılan şirin kodlardan "Sana bir fıkra anlatıyım mı?" kodunun .exe'sini verip gönderdim.

Şirin Baba: İyi yapmışsın Şirinim. Bu böyle devam etseydi, şirinameyle birinizi Optimum Şirin atayacaktım ben de.

3) (Her boşluk 2 puan - en iyi 13 cevap değerlendirmeye alınır - toplam 26 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.

SÖZDE KOD	SÖZDE EKRAN ÇIKTISI
1. Başla 2. sayac=0 3. sayi=0 4. Eğer sayi > 20 ise git 8 5. Eğer sayi mod _____ = 0 ise sayac=sayac+1 6. sayi=sayi+1 7. Git 4 8. Yaz sayac 9. Dur	7 Sözde kodun ekran görüntüsü yukarıdaki şekilde tek bir sayı olmaktadır. Soruda geçen <i>mod</i> işlemi C dilindeki %'e denk olarak kullanılmıştır ve sadece tam sayılarla çalışabilmektedir. Soruda boşlukların dışını doldurmayınız. Sadece boşluklara yazdıklarınız değerlendirmeye alınır. Bir boşluğun birden fazla doğru cevabı olabilir. Uygun(gereksiz kısımları olmayan) birini yazmanız yeterlidir. Gereksiz komut/ifadeler eklemeyiniz. Bu tür durumlarda kodunuz çıktı açısından isteneni yapıyor olsa bile puan alamayabilirsiniz.

KOD

Çifte harfli, aynı harfi(küçük ya da büyük harf şeklinde) peş peşe içermenin en az 1 kez gerçekleştiği sözcüklere (örn. saat, cool, allık) deniliyor olsun. Kullanıcıdan bir sözcük alıp çifte harfli olma durumunu ekrana yazdıran kod.

```
#include <stdio.h>
#include <string.h>
int cifteHarf(char *ptr);
void kucult(char[]);
int main()
{
    char giris[100];
    printf("Sozcuk:");
    scanf("%s", giris);
    kucult(giris);
    if(cifteHarf(giris))
        printf("Cifte harfli");
    else
        printf("Cifte harfsiz");
    return 0;
}
int cifteHarf(char *ptr)
{
    int i, l = _____(ptr);

    for(i=0; i _____ l-1; i++)
    {
        if( ( *(ptr+i) ) == ( *(ptr _____) ) )
            return 1;
    }
    return 0;
}
void kucult( char _____ )
{
    // 'A'nın ASCII'si 65, 'a'nın ASCII'si 97 olmaktadır.
    int i;

    for(i=0; input[i] != _____; i++)
        input[i] = ((int)input[i] < (int)'a'? (char)((int)input[i] + ((int)'a' - (int)'A')) : _____);
}
```

EKРАН VE DOSYA ÇIKTILARI

```
Sozcuk:mustenkif
Cifte harfsiz
-----
Process exited after
Devam etmek için bi
```

ÖRNEK EKРАН ÇIKTISI
Kod çalıştırıldıktan sonra kullanıcı "mustenkif" yazıp enter'a basmıştır.

```
Sozcuk:kubBe
Cifte harfli
-----
Process exited after
Devam etmek için bi
```

ÖRNEK EKРАН ÇIKTISI
Kod çalıştırıldıktan sonra kullanıcı "kubBe" yazıp enter'a basmıştır. b harfi peş peşe geçtiği için, ikinci b harfi büyük olarak geçmesine rağmen istenen şekilde çifte harfli olarak algılanabilmiştir ve ekrana "Cifte harfli" yazdırılmıştır.

```
Sozcuk:boolean
Cifte harfli
-----
Process exited after
Devam etmek için bir
```

ÖRNEK EKРАН ÇIKTISI
Kod çalıştırıldıktan sonra kullanıcı "boolean" yazıp enter'a basmıştır. Bu sözcükte 'o' harfi peş peşe geçtiği için "Cifte harfli" yazdırılmıştır.

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 değerlendirmeye alınır.

KOD

Sudokunun küçük bir versiyonuna benzer şekilde, 3x3'lük bir dizinin 0 olan kısımlarını [1-9] arası bir rakamla, işlem sonunda hiçbir rakam dizi içinde tekrar etmiyor olacak şekilde dolduran ve bu diziyi ekrana yazdıran kod.

ÖRNEK EKRAN ÇIKTISI

```
#include <stdio.h>
#include <stdlib.h>
void a(int[3][3], int);
void b(int [3][3]);
int main()
{
    int c[3][3] = { { 9, 1, 0 } , { 0, 0, 2 } , { 0, 0, 0 } };
    printf("Kutu:\n");
    b(c);
    a(c,0);
    return 0;
}
void b(int d[3][3])
{
    int e,f;
    for(e=0; e<3; e++)
    {
        for(f=0; f<3; f++)
        {
            printf("%d ", _____);
        }
        _____;
    }
}
void a(int g[3][3], int h)
{
    if(_____)
    {
        printf("Cozumu\n");
        b(g);
        _____;
    }
    int i = h/3;
    int j = h%3;
    if( g[i][j] )
    {
        _____;
    }
    else
    {
        int k;
        for(k=1; k<=_____; k++)
        {
            int m,n;
            int l = 1;
            for(m=0; m<3; m++)
            {
                for(n=0; n<3; n++)
                {
                    if ( g[m][n] == k )
                    {
                        _____;
                    }
                }
            }
            if(l==0) continue;
            a( g, h+1);
            g[i][j]=0;
        }
    }
    return;
}
```

Kutu:
9 1 0
0 0 2
0 0 0
Cozumu
9 1 3
4 5 2
6 7 8

ÖRNEK EKRAN ÇIKTISI

Kod çalıştırıldıktan sonra ekranda bu çıktı belirmiş ve kod sonlanmıştır. Kutudaki 0 harici değerlere dokunulmamış; 0 olan yerlere [1-9] arasında bir değer yerleştirilmiştir. Son durumda hiçbir rakamın tekrar etmeme koşulu sağlanmıştır; çünkü hiçbir rakam iki kez geçmemektedir.

4) (Her çıktı 3 puan - en iyi 18 cevap değerlendirmeye alınır - toplam 54 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 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 i;
```

```
    for(i=1;i<=5;i++)
```

```
    {
```

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

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

```
    }
```

```
    return 0;
```

```
}
```


```
//bisiklet.c dosyası
```

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

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

```
int main( int argc, char* argv[] )
```

```
{
```

```
    char dizgi[100];
```

```
    strcpy(dizgi, argv[0]);
```

```
    strncat(dizgi, argv[1], argc);
```

```
    printf("%c", dizgi[0]=='b'? 'q': 'w');
```

```
    strncpy(dizgi, argv[2], 2);
```

```
    printf("%s", dizgi[0]=='b'? "RR": "TT");
```

```
    return 0;
```

```
}
```

Komut Ekranı

```
gcc bisiklet.c -o bisiklet.exe
bisiklet araba tren ucaak 3
```


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



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

```
int main()
```

```
{
```

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

```
char dizgi[5][3][100];
```

```
int i=0;
```

```
do
```

```
{
```

```
    fscanf(dosya, "%s", dizgi[i][0]);
```

```
    i++;
```

```
}while(i<5);
```

```
printf("%s\n", dizgi[4][0]);
```

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

```
return 0;
```

```
}
```


```
#include <stdio.h> //kod.c dosyasi
```

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

```
typedef union
```

```
{
```

```
    char ulke[8];
```

```
    int vize;
```

```
}EU;
```

```
int main(int argc, char* argv[])
```

```
{
```

```
    EU AB;
```

```
    int i;
```

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

```
        AB.ulke[i]=argv[1][i];
```

```
    AB.ulke[i] = '\0';
```

```
    printf("%s\n", AB.ulke);
```

```
    AB.vize = 0;
```

```
    switch( strcmp(AB.ulke, "tr") )
```

```
    {
```

```
        case 0:
```

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

```
            break;
```

```
        default:
```

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

```
    }
```

```
    return 0;
```

```
}
```

Komut Ekranı

```
gcc kod.c -o kod.exe
```

```
kod.exe trakya
```


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

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

```
int main()
```

```
{
```

```
    int s;
```

```
    printf("Sayi:");
```

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

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

```
    s++;
```

```
    {
```

```
        s++;
```

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

```
        int s = 23;
```

```
        s+=5;
```

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

```
    }
```

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

```
    return 0;
```

```
}
```

S	a	y	i	:	3	+	1		

/*Kod çalıştıktan sonra kullanıcı "3+1" 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>
```

```
double p (double a, double b[]);
```

```
int main()
```

```
{
```

```
    double d[4]={1.11, 2.22, 3.33, 4.44};
```

```
    printf("%d\n", (int)p(3.0, d));
```

```
    return 0;
```

```
}
```

```
double p (double a, double b[])
```

```
{
```

```
    if(a<1.5)
```

```
        return 7.0;
```

```
    return p(a-1.0, b)+(int)a;
```

```
}
```


```
#include <stdio.h>
void s (int q);
int main()
{
    int q[3] = {1,2,3};
    s(q[0]);
    printf("%d", q[0]);
    return 0;
}
void s (int q)
{
    if(q==3)
        return;
    s(q+1);
    printf("%d", q);
    s(q+1);
    printf("%d", q);
    s(q+1);
}
```


```
#include <stdio.h>
int main()
{
    int r=4;
    int t=3;
    int i,j;
    for(i=1; i<=t;i++)
    {
        for(j=1; j<=r;j++)
        {
            if(i==1 || i==t || j==1 || j==r)
                printf("");
        }
        printf("\n");
    }
    return 0;
}
```


```
//kod.c dosyasi
#include <stdio.h>
int main(int argc, char* argv[])
{
    int* p = &argc;
    char* r;
    r = &argv[0][1];
    printf("%c", *(r+1));
    int dizi[4]={9,1,5,3};
    printf("%d", dizi[*p]);
    p= &dizi[2];
    int**t;
    t = &p;
    printf("%d", **t);
    p--;
    printf("%d", *p);
    return 0;
}
```

Komut Ekranı

```
gcc kod.c -o kod.exe
kod.exe 141
```


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

```
#include <stdio.h>
int main()
{
    double dolar;
    int tl = 0;
    dolar = tl * 5.75;
    int* y = &tl;
    printf("Sayi:");
    scanf("%d", y);
    printf("%d+", *y);
    printf("%d+", tl);
    printf("%.0lf", dolar);
    return 0;
}
```

S	a	y	i	:	1	0	0		

/*Kod çalıştıktan sonra kullanıcı "100" 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>
void yazdir(int a[]);
int artir(int a);
int main()
{
    int a[4] = {1,2,3,4};
    yazdir(a);
    artir(a[2]);
    yazdir(a);
    a[3] = artir(a[0]);
    yazdir(a);
    return 0;
}
int artir(int a)
{
    a++;
    return a;
}
void yazdir(int a[])
{
    int i;
    for(i=0; i<4; i++)
        printf("%d", a[i]);
    printf("\n");
}
```


//kod.c dosyası

```
#include <stdio.h>
int main(int argc, char* argv[])
{
    // a'nin asciisi 97
    int x= (int)argv[2][1];
    printf("%d",--x);
    /*
    x+=x+++++x;
    */
    x/=argc;
    printf("%d",x++);
    return 0;
}
```

Komut Ekranı

```
gcc kod.c -o Sirin.exe
Sirin Baba acaba nerede ki?
```


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

#include <stdio.h>

```
typedef struct
{
    char isim[10];
    int zar;
}backgammon;
int main()
```



```
{
    FILE* dosya = fopen("tavla.txt","r");
    backgammon tavla;
    int i = 0;
    while(i<3)
    {
        fscanf(dosya, "%s", tavla.isim);
        fscanf(dosya, "%d", &tavla.zar);
        i++;
    }
    for(i=0; i<2; i++)
        printf("%s\n", tavla.isim);

    return 0;
}
```


#include <stdio.h>

enum boolean{ dogru, yanlis };

typedef struct

```
{
    int x,y[3];
} var;
void fonk(var*);
int main()
{
    var yok = {0, {}};
    yok.x=yanlis;
    yok.y[dogru]=yok.x;
    printf("%d%d\n",yok.x,yok.y[1]);
    fonk(&yok);
    printf("%d%d\n",yok.x,yok.y[0]);
    return 0;
}
void fonk(var* P)
{
    if(P->x<-9)
        P->y[0]=4;
    (*P).x=5;
}
```


```
#include <stdio.h>
typedef struct
{
    int sayi;
} t;
int p(t* R);
int main()
{
    t x;
    x.sayi=14;
    p(&x);
    if(x.sayi>20)
        printf("P");
    else if(x.sayi>25)
        printf("R");
    else
        printf("T");
    return 0;
}
int p(t* R)
{
    R->sayi=28;
    return 19;
}
```


```
#include <stdio.h>
#include <string.h>
typedef struct
{
    char isim[10];
    int fiyat;
}apple;
int main()
{
    apple elma = {"sari", 3};
    // a'nin ASCIIsi 97'dir.
    strcpy(elma.isim,"100");
    printf("%s", elma.isim);
    printf("%d", elma.fiyat);
    return 0;
}
```


```
#include <stdio.h>
#include <string.h>
#define S 4
typedef struct
{
    char name[4][10];
    int quota;
} boat;
int main()
{
    boat tekne= { {"4","13","2","141"}, S };
    int i;
    for(i=2; i<tekne.quota; i++)
    {
        strcpy(tekne.name[i],tekne.name[i-1]);
    }
    while(1==1)
    {
        i--;
        tekne.quota = strlen(tekne.name[i]);
        if(!(tekne.quota>1))
            break;
        printf("%s\n", tekne.name[i]);
    }
    return 0;
}
```


```
#include <stdio.h>
void ciz(int t, int y);
int main()
{
    ciz(0, 0); ciz(0, 1); ciz(0, 2);
    return 0;
}
void ciz(int t, int y)
{
    char bosluk = ' '; //1 bosluk
    if(t==2)
    {
        printf("\n");
        return;
    }
    else if(y%2==1)
        printf("*****");
    else
        printf("***c**", bosluk);
    printf("%c", bosluk);
    ciz(t+1, y+1);
}
```


```
#include <stdio.h>
void x(int dizi[], int n);
int main()
{
    int a[10]= {7,2,8,6,9,0,4,2};
    x(a,9);
    printf("%d", a[0]);
    printf("%d", a[4]);
    printf("%d", a[7]);
    printf("%d", a[9]);
    return 0;
}
void x(int dizi[], int n)
{
    int i, temp;
    int d;
    do
    {
        d = 1;
        for(i = 1; i<n - 2; i++)
        {
            if( dizi[i] > dizi[i-1] )
            {
                temp = dizi[i];
                dizi[i] = dizi[i-1];
                dizi[i-1] = temp;
                d = 0;
            }
        }
        n--;
    }
    while(!d);
}
```


```
#include <stdio.h>
int main()
{
    int a[3][5]=
    {
        {1,2,3,4,5},
        {9,8,3,9,2},
        {3,9,0,3,1},
    };
    int i,j;
    for(i=141; i<=142; i++)
    {
        for(j=0; j<2; j++)
        {
            if( a[j][0] < a[j+1][0] )
            {
                int x,t;
                for(x=0; x<5; x++)
                {
                    t=a[j][x];
                    a[j][x]=a[j+1][x];
                    a[j+1][x]=t;
                }
            }
        }
        for(i=0; i<3; i++)
        {
            for(j=0; j<5; j++)
                printf("%d", a[i][j]);
            printf("\n");
        }
        return 0;
    }
}
```


GERİ BİLDİRİM SORULARI (EKSTRA 5 PUAN)

5 PUAN: Eğer aşağıdaki maddeleri cevaplarsanız

1. Bu dersten almayı beklediğiniz harf notu nedir? AA BA BB CB CC DC DD FF
2. Bu sınava yeterince çalışabildiğinizi düşünüyor musunuz?
3. Kuramsal dersler nasıl daha iyi olabilir?
4. Uygulamalı dersler hakkında neler düşünüyorsunuz?
5. Ödevlerin verimini(öğrenciyi sınava hazırlama özelliğini) artırmak için ödev zorluğu ve miktarı nasıl düzenlenebilir?
6. Geri bildirimde bulunmak istediğiniz diğer konular nelerdir?

BİLGİSAYAR PROGRAMLAMA
DÖNEM SONU SINAVI

CEVAP

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 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 yakalanıldığı 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) (10 puan) Verilen koşulları 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.

Bir dersi veren hocanın ismi ve bu dersi alan 80 öğrenciye atanan ders numaraları bilgisayara aktarılacaktır. Kodunuzda aşağıdakileri dikkatlice gerçekleştirmeniz istenmektedir.

Kodda Şekil-1'de gözüktüğü gibi iki tane yapı tanımı olmalıdır. Şekil 1 DevCpp'daki Project/Class Browser (Proje/Sınıf Gezgini) panelinden elde edilmiştir. Sizin kodunuz da DevCpp'a aktarılacak olursa Şekil 1'dekiyle **tamamen aynı** görünümün (aynı isimlerin, aynı sayıların ve diğer detayların) elde edilebiliyor olması istenmektedir.



Şekil 1 - Kodunuzda bulunması istenen yapılar
(DevCpp Project/Class Browser altındaki görünüm)

main'de *ders* yapısından bir değişken tanımlamanız ve bu yapıya şu bilgileri yerleştirmeniz istenmektedir:

- *hoca* bilgisi "*Sirin Baba*" olarak atanacaktır.
- öğrencilere isim atanmayacaktır; ancak en düşük indisli öğrenci 101 ders numarasına sahip olmak üzere, indis arttıkça ders numaraları birer birer artarak öğrencilere atanacaktır. Bu nedenle son öğrencinin (en büyük indisli öğrenci) ders numarası 180 olmalıdır.

Kod çalıştığında ekrana **bir şey yazdırmaz** ve "**return value**" olarak 141 değerini dönerek sonlanır. Kodun görevi sadece RAM'e bu veriyi işlemek ve istenen şekilde sonlanmaktır. Kod sonlandığında verinin yok olacak olması, bu soru açısından bir sakınca oluşturmamaktadır.

Kod **gerekless hiçbir kütüphane içermemelidir**.

ÖNEMLİ UYARILAR:

- Detaylandırılmış uyarı ayarları açıkken **hiçbir uyarı ya da hata vermeyecek** bir kod yazmaya çalışınız.
- 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; **tür dönüşümü**, **kütüphane eklenmesi** gibi olgulara dikkat etmeden yazdığınız kodlarda da kodunuz doğru çalışsa bile puan kaybı yaşanabilir. Örneğin, tür dönüşümü yapılan her yerde bunu kodunuzda açıkça belirtin; bunu derleyicinin "hoşgörüsü"ne bırakmayın.
- Kod içerisinde **boşluk** bırakmanın kritik olduğu kısımlardaki boşluklarınız **açık olmalı**. Gerekirse ilgili satıra yorum ekleyerek bunu belirtiniz.
- 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** unutulabilmektedir. 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**.

#include <string.h>

// BİL141 DERSİ SINAVINDAKİ 1.SORU KODUM

```
typedef struct
{
    char isim[30];
    int dersNo;
} ogrenci;
```

Notlandırma şekli
için bkz.

Piazza → YAZILI
SINAV

BİLGİLENDİRME
pdf'i



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

```
typedef struct
```

```
{
    char hoca[30];
    ogrenci ogr[80];
} ders;
```

★ TEMEL BİLGİ ★

ders yapısı üstte tanımlandıysa da
kod yukarıdan aşağı aktığı için bu
satırda "ogrenci tanımsız" hatası
gelecekti.

```
int main()
```

```
{
    ders Bil141;
    strcpy(Bil141.hoca, "Şirin Baba");
```

```
    int i;
```

```
    for(i=0; i<80; i++)
```

```
        Bil141.ogr[i] = 101 + i;
```

```
    return 141;
```

```
}
```

// printf, scanf gibi input/output fonksiyonları yok.

// bu şekilde standart input output

// kütüphanesine (stdio.h) gerek yok.

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: Merhaba Şirin Baba. 141 numaralı öğrenciyim. Sana birkaç sorum olacak.

Şirin Baba: Sor bakalım.

Usta Şirin: Kodumda `FILE* gozluklu = fopen("data.txt", "w");` ile dosya açıp, `fprintf` ile bu dosyaya değer yazdırıyorum, kodu 10 saniye bekletiyorum, sonra `fprintf` ile başka bir değer daha yazdırıp, 10 saniye daha bekletiyorum. Ardından da kod sonlanıyor. Ancak bu 10 saniyeler içinde dosyayı açıp içine baktığımda `fprintf`'lerin etki etmediğini görüyorum, dosyanın içi boş gözüküyor, neden?

Şirin Baba: Daha kaç kez söylemem gerek bilemiyorum. `fprintf` komutu kodda çalıştırılınca anında dosyaya yazı yazmaz. Sana `fclose` önerirdim ama ilk 10 saniyeden sonra da dosyayla işin devam edeceği için dosyayı tekrar açman gerekir ki hoş olmaz. O halde kodundaki

her `fprintf`'ten hemen sonraya `fflush(gozluklu)`; şeklinde bir komut ekle. Bunu her iki `fprintf`'in için de ayrı ayrı uygula.

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

`typedef struct`

{

`char sanzuman;`

`}vehicle;`

Şimdi `main`'in içerisinde

`vehicle.sanzuman = 'A';`

`printf("%s", vehicle.sanzuman);`

yazdığımda kod hata veriyor.

`vehicle arac;`

`arac.sanzuman = 'A';`

Gargamel

: %s değil de %c yazarsan sorun çözülür; çünkü `vehicle.sanzuman` değişkeni bir dizgi değildir, sade `char` türündedir.

Usta Şirin: feof fonksiyonu ne işe yarar?

Gargamel: Bir dosyayı *fopen* ile açmak istediğinde eğer o dosya yoksa feof sana 0 harici bir değer döner, aksi durumda 0 döner. Böylece dosyanın mevcut olup olmadığını anlamana ve gerekiyorsa kodu sonlandırmana yarar.

*dosyada Sana
gelindi mi
File end of
if (dosya == NULL)*

Usta Şirin: Bilgisayarımda DevCpp artık açılmamaya başladı. Bir türlü düzeltemiyorum. Alternatif bir program mı yüklemeliyim, yoksa yüklemeyen de kodumun derlenmesini sağlayabilir miyim?

Şirin Baba: Evet, DevCpp çalışmasa bile ve alternatif bir program yüklemeksizin kodunu derletmen mümkündür. *→ gcc kod.c -o kod.exe (KODU EKZANİ)*

Usta Şirin: Elimde bir cümle var ve ben bunu sözcüklere ayırmak istiyorum. Sözcükler arasında nokta, virgül ya da boşluk olabilir. Bunu nasıl yaparım?

Şirin Baba: Bunu yapmak için, *str tok* fonksiyonu en uygun fonksiyonlardan biridir. *string tokenizer*
bilet kağıdı gibi kırpık kırpık eder
string.h kütüphanesindedir. İstediyiğin ayraçları parametre olarak alarak cümleyi sözcüklere ayırmana yardımcı olur.

Usta Şirin: Fibonşirin sayıları 1,1,2,3,5 ... şeklinde her biri kendisinden önceki iki sayının toplamı olacak şekilde ilerleyen sayılarmış. 41. fibonşirin sayısını hesaplayan bir kod yazmak istiyorum. İşlemciyi yormamak açısından özyinelemeli şekilde kodlamak mı daha uygun olur, yoksa bir dizi ve bir for döngüsü ile mi yapayım?

Gargamel: Özyinelemeli olarak yapman uygun olur; çünkü özyinelemeli kodlama kodunu birkaç satıra indirgeyecek ve böylece kodunun işlemciyi daha az yormasını sağlayacaktır. *→ kısa kod = iyi kod*

Usta Şirin: `char dizgi[100]; scanf("%s", dizgi);` şeklinde kullanıcıdan bir giriş alıyorum. *gereksiz yer fazla hesap*
"eğer bu giriş 'A' karakteri ise" ifadesine ihtiyacım var.

Gargamel: *if (dizgi[0] == 'A')*
karakter!
`if ( strcmp(dizgi[0], 'A') )` işini görür. Ancak kodun başına "string.h" kütüphanesini eklemiş olmalısın.

Usta Şirin: `char isim[?] = { 'S', 'i', 'r', 'i', 'n', 'e' };` şeklinde bir dizi tanımlıyorum. ? yerinde bir adet sayı var. Bu sayının minimum değeri ne olmalıdır? *0 1 2 3 4 5*

Şirin Baba: 6 adet elemanın bu diziye sığabilmesi için ? yerine en az 6 sayısını yazmalısın. *printf %s ile yazdırmak istersen, '\0' a da gerek olurdu.*

Usta Şirin: Bir dizgi içerisinde kullanıcıdan aldığım ay değeri (örn. "Mart") var. Ben bunu sayıya (Mart ise 3, Nisan ise 4 gibi) çevirmek istiyorum. *enum* bu iş için uygun mudur?

Şirin Baba: Hayır, *enum* sadece kod içerisindeki Mart şeklindeki ifadeleri 3'e vs. çevirebilir. Kullanıcıdan aldığın ay değerini kapsamaz. *enum iyidir ama her yerde deva değildir.*

Usta Şirin: 18->B diye bir ifade gördüm. Yandan bakınca gözlüklü şirinin papyon takmış haline benzettim.

Şirin Baba: Bu ifadeden 18'in kesinlikle bir işaretçi olduğu anlaşılıyor. 18'in işaret ettiği yapı değişkeninin içindeki B değişkenine erişim sağlanmış. *Çünkü (*18). B yazmaya denk.*

Usta Şirin: Yapı değişkeni demişken elimde bellekte(RAM)'de 10MB yer tutan kocaman bir yapı değişkeni var. Bunu bir fonksiyona göndermek istiyorum. Fonksiyon bu değişkenin değerini değiştirmeyecek; sadece değişkendeki verileri ekrana yazdıracak. Belleği de verimli kullanmak istiyorum. main içerisinde fonksiyonu çağırırken fonksiyona bu değişkeni nasıl göndermeliyim? Fonksiyonun ismi *fonk*, değişkenin ismi *sirinkoy*.

Şirin Baba: Fonksiyonu *fonk (&sirinkoy)*; şeklinde çağırarak değişkeni adres ile göndermen uygun olur. *adres ile göndermenin bir avantajı da bellek yönetimi*

Usta Şirin: Bir dizgi tanımlamak için tek yol *char dizgi[100]*; yazmak mıdır? *char* dizgi* şeklinde de dizgi tanımlayabilir miyim?

Şirin Baba: 1.sorunun cevabı: Hayır, alternatifi de vardır. 2.sorunun cevabı: Evet, tanımlayabilirsin; ancak

peşinden *malloc* fonksiyonu ile bellekten 100 baytlık bir yer ayırt ettirmen ve bu yerin ilk baytının adresini dizgiye ataman gerekir. Sana sadece fonksiyonun ismini yazdım; sözdizimini şirinnetten öğrenebilirsin.

Usta Şirin: Kullanıcı klavyeden "*Uykucu Şirin*" şeklinde bir giriş yapacak ve ben bunu *char sirinlism[100]* şeklinde tanımlı bir dizgiye kaydetmek istiyorum. *Tek sözcük alır. → gets / fgets*

Gargamel: *scanf("%s", sirinlism);* işini görür. Dizgi olduğu için & koymuyorsun. *→ sıpkalı scanf*

Usta Şirin: Gözlüklü şirine sordum, "*Benim gözüm zaten bozuk, sen bunu Şirin Baba'ya sor*" dedi. *dizi[]... ailesi*

$$\text{int dizi}[3][5][7][10] = \{ \{ \{ \{1,2\}, \{3,4\} \}, \{ \{5,6\}, \{7,8,9\} \}, \{ \{ \{10\}, \{11\}, \{12,13,14\} \}, \{ \{15,16\}, \{17,18,19,20\} \} \} \};$$
...x+1... ++
 dizisini tanımladım. *dizi[1][0][2][1]*'in değeri nedir? *→ dizi[23]... ailesi yok, {} operatörü onları 0 ile doldurur.*

Şirin Baba: Şirin Baba başka ne işe yarar! Dur bakayım, ben de merak ettim...

Şirin Baba: Tamam buldum. Bu tanımla ve atamayla *dizi[1][0][2][1]*'in değeri 13 olmuştur.

Usta Şirin: Peki ben bu dizi elemanını, *main*'den *pembe* isimli bir fonksiyona *pembe(dizi[1][0][2][1])* şeklinde göndersem fonksiyonda yapılan değişiklik kalıcı olur mu? *dizi değil dizinin bir elemanı (int) geçiyor → pass by value*

Gargamel: Evet şirinim. Dizilerde *adres ile gönderim* söz konusu olduğu için dizinin elemanı üzerinden yapılacak bir değişiklik *main*'de de kalıcı olur.

Usta Şirin: [22, 88] (22 ve 88 dâhil) aralığında 11'in-katı olan rastgele bir sayı üretmek ve bunu şirin değişkenime kaydetmek istiyorum. *int sirin_degisken = (rand()%A + B) * 11;* diye bir kod kurguladım ve kodumda kütüphane eklemek gibi ilgili düzenlemeleri yaptım. Bir tek A ve B sayılarını bulmak kaldı. Bir ipucu verir misin?

Şirin Baba: Kurguladığın koddaki A sayısından B sayısını çıkarınca 5 kalır.

[2-8] aralığında 6

$$B = 2 \rightarrow \text{rand()} \% A \rightarrow 7 = A \rightarrow 7 - 2$$

$$\text{rand()} \% A \rightarrow [0-6]$$

Usta Şirin: Geçen gün, Güçlü Şirin gelmiş "[312-634] aralığında bir sayı tuttum...

Şirin Baba: "Aman!" diyeyim Usta Şirin...

Usta Şirin: Bak dinle lütfen Şirin Baba. Anlatıyorum: Gelmiş diyor ki "Sen tahmin edeceksin, bense her tahmininden sonra dambıl gibi 'çık', 'in'...

Şirin Baba: Ben çıkıyorum.

Usta Şirin: in Şirin Baba, aman... dur Şirin Baba. Bak öyle değil. Ben de senin gibi sıkıldım. Dedim "Yeter artık, sayı mayı tutma Güçlü Şirin. Her sınavdan önce biri gelip, çık, in, bu nedir böyle?". Derste yazılan şirin kodlardan "Sana bir fıkra anlatıyım mı?" kodunun .exe'sini verip gönderdim.

Şirin Baba: İyi yapmışsın Şirinim. Bu böyle devam etseydi, şirinameyle birinizi Optimum Şirin atayacaktım ben de.

3) (Her boşluk 2 puan - en iyi 13 cevap değerlendirmeye alınır - toplam 26 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.

SÖZDE KOD	SÖZDE EKRAN ÇIKTISI
1. Başla 2. sayac=0 3. sayi=0 4. Eğer sayi > 20 ise git 8 5. Eğer sayi mod <u>3</u> = 0 ise sayac=sayac+1 6. sayi=sayi+1 7. Git 4 8. Yaz sayac 9. Dur <i>[Handwritten note: [0-20] aralığında X'in katı olan sayı sayısı 7 ise X kaçtır? 3'tür.]</i>	7 Sözde kodun ekran görüntüsü yukarıdaki şekilde tek bir sayı olmaktadır. Soruda geçen <i>mod</i> işlemi C dilindeki %'e denk olarak kullanılmıştır ve sadece tam sayılarla çalışabilmektedir. Soruda boşlukların dışını doldurmayınız. Sadece boşluklara yazdıklarınız değerlendirmeye alınır. Bir boşluğun birden fazla doğru cevabı olabilir. Uygun(gereksiz kısımları olmayan) birini yazmanız yeterlidir. Gereksiz komut/ifadeler eklemeyiniz. Bu tür durumlarda kodunuz çıktı açısından isteneni yapıyor olsa bile puan alamayabilirsiniz.

KOD

Çifte harfli, aynı harfi(küçük ya da büyük harf şeklinde) peş peşe içermenin en az 1 kez gerçekleştiği sözcüklere (örn. saat, cool, allık) deniliyor olsun. Kullanıcıdan bir sözcük alıp çifte harfli olma durumunu ekrana yazdıran kod.

```
#include <stdio.h>
#include <string.h>
int cifteHarf(char *ptr);
void kucult(char[]);
int main()
{
    char giris[100];
    printf("Sozcuk:");
    scanf("%s", giris);
    kucult(giris);
    if(cifteHarf(giris))
        printf("Cifte harfli");
    else
        printf("Cifte harfsiz");
    return 0;
}

int cifteHarf(char *ptr)
{
    int i, l = strlen(ptr);
    for(i=0; i < l-1; i++)
    {
        if( (*(ptr+i)) == (*(ptr+i+1)) )
            return 1;
    }
    return 0;
}

void kucult( char input[])
{
    // 'A'nın ASCII'si 65, 'a'nın ASCII'si 97 olmaktadır.
    int i;

    for(i=0; input[i]!='\0'; i++)
        input[i] = ((int)input[i] < (int)'a'? (char)((int)input[i] + ((int)'a' - (int)'A')) : input[i]);
}
```

main'de bu ptr'nin işaret ettiği yer rezerve (char ptr[100]) vs. olduğu için malloc'a gerek olmaz.

≤ olmalı

burası en son gereksiz yer '\0'a erişirdi.

ptr[i] yazmaya denk.

bu sayede ASCII'leri ezberlemeye gerek kalmaz.

EKRAN VE DOSYA ÇIKTILARI

Sozcuk:mustenkif
Cifte harfsiz

Process exited after
Devam etmek için bir

ÖRNEK EKRAN ÇIKTISI

Kod çalıştırıldıktan sonra kullanıcı "mustenkif" yazıp enter'a basmıştır.

Sozcuk:kubBe
Cifte harfli

Process exited after
Devam etmek için bir

ÖRNEK EKRAN ÇIKTISI

Kod çalıştırıldıktan sonra kullanıcı "kubBe" yazıp enter'a basmıştır. b harfi peş peşe geçtiği için, ikinci b harfi büyük olarak geçmesine rağmen istenen şekilde çifte harfli olarak algılanabilmiştir ve ekrana "Cifte harfli" yazdırılmıştır.

Sozcuk:boolean
Cifte harfli

Process exited after
Devam etmek için bir

ÖRNEK EKRAN ÇIKTISI

Kod çalıştırıldıktan sonra kullanıcı "boolean" yazıp enter'a basmıştır. Bu sözcükte 'o' harfi peş peşe geçtiği için "Cifte harfli" yazdırılmıştır.

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

Sudoku'nun küçük bir versiyonuna benzer şekilde, 3x3'lük bir dizinin 0 olan kısımlarını [1-9] arası bir rakamla, işlem sonunda hiçbir rakam dizi içinde tekrar etmiyor olacak şekilde dolduran ve bu diziyi ekrana yazdıran kod.

```
#include <stdio.h>
#include <stdlib.h>
void a(int[3][3], int);
void b(int [3][3]);
int main()
```

```
{
    int c[3][3] = { {9, 1, 0}, {0, 0, 2}, {0, 0, 0} };
    printf("Kutu:\n");
    b(c);
    a(c,0);
    return 0;
}
```

```
void b(int d[3][3])
```

```
{
    int e,f;
    for(e=0; e<3; e++)
    {
        for(f=0; f<3; f++)
        {
            printf("%d ", d[e][f]);
        }
        printf("\n");
    }
}
```

```
void a(int g[3][3], int h)
```

```
{
    if(h == 9)
    {
        printf("Cozumu\n");
        b(g);
        exit(0);
    }
```

```
    int i = h/3;
    int j = h%3;
    if( g[i][j] )
```

```
    {
        a(g, h+1);
    }
```

```
    else
```

```
    {
        int k;
        for(k=1; k<=9; k++)
```

```
        {
            int m,n;
            int l = 1;
            for(m=0; m<3; m++)
```

```
            {
                for(n=0; n<3; n++)
                {
                    if ( g[m][n] == k )
```

```
                    {
                        break;
                    }
                }
            }
```

```
            if(l==0) continue;
```

```
            /* Keske son hafta iyice dinlesaydim.*/
            a(g, h+1);
```

```
            g[i][j]=0;
        }
```

```
    }
    return;
}
```

0	1	2
3	0	2
0	0	0

bu hane
gizlenmiş
is
bitmiştir.

Kutu:

9 1 0
0 0 2
0 0 0

Cozumu

9 1 3
4 5 2
6 7 8

ÖRNEK EKRAN
ÇIKTISI

Kod çalıştırıldıktan sonra ekranda bu çıktı belirmiş ve kod sonlanmıştır.

Kutudaki 0 harici değerlere dokunulmamış; 0 olan yerlere [1-9] arasında bir değer yerleştirilmiştir. Son durumda hiçbir rakamın tekrar etmeme koşulu sağlanmıştır; çünkü hiçbir rakam iki kez geçmemektedir.

Aşağıdaki kodların altındaki kutulara ekran çıktılarını **çok dikkatlice** yazınız.

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

10

1	2	3	4	5				

```
gcc bisiklet.c -o bisiklet.exe
bisiklet araba tren ucak 3
```

11

```
#include <stdio.h>
void s (int q);
int main()
{
```

```
    int q[3] = {1,2,3};
```

```
    s(q[0]);
```

```
    printf("%d", q[0]);
```

```
    return 0;
```

```
}
void s (int q)
{
```

```
    if(q==3)
```

```
        return;
```

```
    s(q+1);
```

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

```
    s(q+1);
```

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

```
    s(q+1);
```

```
}
```

2	2	1	2	2	1	2	2	1	

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

```
int main()
```

```
{
```

```
    int r=4;
```

```
    int t=3;
```

```
    int i,j;
```

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

```
    {
```

```
        for(j=1; j<=r;j++)
```

```
        {
```

```
            if(i==1 || i==t || j==1 || j==r)
```

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

```
        }
```

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

```
    }
```

```
    return 0;
```

```
}
```

*	*	*	*						
*	*								
*	*	*	*						

```
//kod.c dosyasi
```

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

```
int main(int argc, char* argv[])
```

```
{
```

```
    int* p = &argc;
```

```
    char* r;
```

```
    r = &argv[0][1];
```

```
    printf("%c", *(r+1));
```

```
    int dizi[4]={9,1,5,3};
```

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

```
    p = &dizi[2];
```

```
    int**;
```

```
    t = &p;
```

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

```
    p--;
```

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

```
    return 0;
```

```
}
```

```
Komut Ekranı
```

```
gcc kod.c -o kod.exe
kod.exe 141
```

```
1551
```

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

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

```
int main()
```

```
{
```

```
    double dolar;
```

```
    int tl = 0;
```

```
    dolar = tl * 5.75;
```

```
    int* y = &tl;
```

```
    printf("Sayi:");
```

```
    scanf("%d", y);
```

```
    printf("%d+", *y);
```

```
    printf("%d+", tl);
```

```
    printf("%.0lf", dolar);
```

```
    return 0;
```

```
}
```

```
S a y i : 1 0 0
```

```
100 + 100 + 0
```

/*Kod çalıştıktan sonra kullanıcı "100" 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>
void yazdir(int a[]);
int artir(int a);
int main()
{
    int a[4] = {1,2,3,4};
    yazdir(a);
    artir(a[2]);
    yazdir(a);
    a[3] = artir(a[0]);
    yazdir(a);
    return 0;
}
int artir(int a)
{
    a++;
    return a;
}
void yazdir(int a[])
{
    int i;
    for(i=0; i<4; i++)
        printf("%d", a[i]);
    printf("\n");
}
```

Handwritten notes:
 - *1. 2. 3. 4. için a[2] olması di.*
 - *a[3] burada (main'de) pncellendi.*

1	2	3	4						
1	2	3	4						
1	2	3	2						

//kod.c dosyası

#include <stdio.h>

int main(int argc, char* argv[])

```
{
    // a'nin asclisi 97
    int x= (int)argv[2][1];
    printf("%d",--x);
    /*
    x+=x+++++x;
    */
    x/=argc;
    printf("%d",x++);
    return 0;
}
```

Handwritten notes:
 - *a@aba*
 - *ASCII → 99*
 - *98*
 - *5*
 - *x = x / argc*

Komut Ekranı

```
gcc kod.c -o Sirin.exe
Sirin Baba acaba nerede ki?
```

9	8	1	9						

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

#include <stdio.h>

```
typedef struct
{
    char isim[10];
    int zar;
}backgammon;
int main()
{
```



Handwritten note: boşluk yok! Bu nedenle o/s tamamlanıyor.

```
FILE* dosya = fopen("tavla.txt", "r");
backgammon tavla;
int i = 0;
while(i<3)
{
    fscanf(dosya, "%s", tavla.isim);
    fscanf(dosya, "%d", &tavla.zar);
    i++;
}
for(i=0; i<2; i++)
    printf("%s\n", tavla.isim);

return 0;
```

Handwritten note: dosyanın değerleri hep aynı yere kaydedilir.

2	e	k	i	M	u	r	e	n	
2	e	k	i	M	u	r	e	n	

#include <stdio.h>

enum boolean{ dogru, yanlis};

typedef struct

```
{
    int x,y[3];
}
```

} var;

void fonk(var*);

int main()

```
{
    var yok = {0, {} };
    yok.x=yanlis;
    yok.y[dogru]=yok.x;
    printf("%d%d\n",yok.x,yok.y[1]);
    fonk(&yok);
    printf("%d%d\n",yok.x,yok.y[0]);
    return 0;
}
```

Handwritten notes:
 - *yok x*
 - *5 yazıyor*
 - *1 0 0*

void fonk(var* P)

{

if(P->x<-9)

P->y[0]=4;

(*P).x=5;

}

1	0								
5	1								

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

```
    int sayi;
} t;
int p(t* R);
int main()
{
    t x;
    x.sayi=14;
    p(&x);
    if(x.sayi>20)
        printf("P");
    else if(x.sayi>25)
        printf("R");
    else
        printf("T");
    return 0;
}
int p(t* R)
{
    R->sayi=28;
    return 19;
}
```

P																			

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

```
    char isim[10];
    int fiyat;
}apple;
int main()
{
    apple elma = {"sari", 3};
    // a'nin ASCII'si 97'dir.
    strcpy(elma.isim, "100");
    printf("%s", elma.isim);
    printf("%d", elma.fiyat);
    return 0;
}
```

1	0	0	3																

```
#include <stdio.h>
#include <string.h>
#define S 4
typedef struct
{
```

```
    char name[4][10];
    int quota;
} boat;
int main()
{
    boat tekne= {"4","13","2","141"}, S;
    int i;
    for(i=2; i<tekne.quota; i++)
    {
        strcpy(tekne.name[i],tekne.name[i-1]);
    }
    while(1==1)
    {
        i--;
        tekne.quota = strlen(tekne.name[i]);
        if(!(tekne.quota>1))
            break;
        printf("%s\n", tekne.name[i]);
    }
    return 0;
}
```

1	3																		
1	3																		
1	3																		

```
#include <stdio.h>
void ciz(int t, int y);
int main()
{
```

```
    ciz(0, 0); ciz(0, 1); ciz(0, 2);
    return 0;
}
```

```
void ciz(int t, int y)
{
```

```
    char bosluk = ' '; //1 bosluk
    if(t==2)
    {
```

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

```
    else if(y%2==1)
        printf("****");
    else
```

```
        printf("%*c", bosluk);
        printf("%c", bosluk);
        ciz(t+1, y+1);
    }
```

*	*	*	*	*															
*	*	*	*	*															
*	*	*	*	*															

#include <stdio.h>

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

int main()

{

int a[10] = {7, 2, 8, 6, 9, 0, 4, 2};

x(a, 9);

printf("%d", a[0]);

printf("%d", a[4]);

printf("%d", a[7]);

printf("%d", a[9]);

return 0;

}

void x(int dizi[], int n)

{

int i, temp;

int d;

do

{

d = 1;

for(i = 1; i < n - 2; i++)

{

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

{

temp = dizi[i];

dizi[i] = dizi[i-1];

dizi[i-1] = temp;

d = 0;

}

}

n--;

}

while(!d);

}

9	4	2	0						

#include <stdio.h>

int main()

{

int a[3][5] =

{

{1, 2, 3, 4, 5},

{9, 8, 3, 9, 2},

{3, 9, 0, 3, 1},

};

int i, j;

for(i=141; i<=142; i++)

{

for(j=0; j<2; j++)

{

if(a[j][0] < a[j+1][0])

{

int x, t;

for(x=0; x<5; x++)

{

t = a[j][x];

a[j][x] = a[j+1][x];

a[j+1][x] = t;

}

}

}

}

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

{

for(j=0; j<5; j++)

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

printf("\n");

}

return 0;

}

9	8	3	9	2					
3	9	0	3	1					
1	2	3	4	5					

GERİ BİLDİRİM SORULARI (EKSTRA 5 PUAN)

5 PUAN: Eğer aşağıdaki maddeleri cevaplarsanız

1. Bu dersten almayı beklediğiniz harf notu nedir? AA BA BB CB CC DC DD FF
2. Bu sınava yeterince çalışabildiğinizi düşünüyor musunuz?
3. Kuramsal dersler nasıl daha iyi olabilir?
4. Uygulamalı dersler hakkında neler düşünüyorsunuz?
5. Ödevlerin verimini(öğrenciyi sınava hazırlama özelliğini) artırmak için ödev zorluğu ve miktarı nasıl düzenlenebilir?
6. Geri bildirimde bulunmak istediğiniz diğer konular nelerdir?