

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 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 çıktısını 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 **istenilen** kısmını yazınız.

Bu sorudaki kodun bir kısmı verilmiştir. Sizden istenen kodda geçen iki fonksiyonun (*min* ve *min2*) içeriğini tamamlamanızdır.

- Kod çalışmaya başladığında örnek ekran çıktısında görüldüğü gibi kullanıcıdan ödev notu girişleri alacaktır. Kullanıcı notları tek tek girip enter'a basacaktır. Tüm notlar birbirinden farklıdır ve her biri [0,215] aralığındadır. Kullanıcının her zaman geçerli girişler yapacağını varsayınız.
- Kod, bu notlar arasındaki **en küçük ve 2. en küçük** notları ekrana yazdıracaktır. Ardından, **tüm notları** ekrana yazdıracaktır.

```
Odev 1 not girisi: 100
Odev 2 not girisi: 115
Odev 3 not girisi: 120
Odev 4 not girisi: 106
Odev 5 not girisi: 149
Odev 6 not girisi: 165
Odev 7 not girisi: 215

En küçük notlu odev-> 100
2. en küçük notlu odev-> 106
Tumu-> 100 115 120 106 149 165 215
-----
Process exited after 33.55 seconds
Devam etmek için bir tuşa basın .
```

ÖRNEK EKRAK ÇIKTISI

Kod çalıştıktan sonra kullanıcı sırasıyla 100, 115, 120, 106, 149, 165 ve 215 girişleri yapıp her girişten sonra enter'a basmıştır. Bu sayılar arasında en küçük olanı 100 olduğu için 100, çıktıda gözüktüğü şekilde ekrana yazdırılmıştır. Ardından 100 haricindeki en küçük değer olan 106 ekrana yazdırılmıştır. Son olarak, kullanıcının girdiği tüm sayılar ekranda gösterilmiştir.

ÖNEMLİ UYARILAR:

- Sorunun başında da belirtildiği üzere **sade komutlar/ifadeler** kullanmanız beklenmektedir. Örneğin, 1 printf ile yapılabilecek kısmı 10 printf ile yaparsanız kodunuz doğru çalışsa bile puan kaybı yaşayabilirsiniz.
- Kod içerisinde **boşluk** bırakmanın kritik olduğu kısımlardaki boşluklarınız **açık olmalı**. 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**.

```
#include <stdio.h>
int min (int dizi [ 7 ] );
int min2 (int dizi [ 7 ] , int min_indis );
int main( )
{
    int notlar [ 7 ] , i;
    for( i = 0 ; i < 7 ; i++ )
    {
        printf("Odev %d not girisi: ", i+1);
        scanf("%d", &notlar [ i ] );
    }
    int min_indis = min (    notlar    );
    int min2_indis = min2 ( notlar , min_indis );
    printf("\nEn küçük notlu odev-> %d\n", notlar [ min_indis ] );
    printf("2. en küçük notlu odev-> %d\n", notlar [ min2_indis ] );
    printf("Tumu-> ");
    for( i = 0 ; i < 7 ; i++ )
        printf("%d ", notlar [ i ] );
    return 0;
} //Kodun devamı bir sonraki sayfadır.
```

KOD SOL TARAFTA VE BİR
SONRAKİ SAYFADA
GÖSTERİLMİŞTİR.
İSTENENLERİ SAĞLAYACAK
ŞEKİLDE EKSİK KISIMLARINI
DOLDURUNUZ.

//Önceki sayfada baş kısmı verilen kodun devamı

int min (**int** dizi [7]) // kendisine gönderilen 7 elemanlı dizinin en küçük elemanının indisini döner.

{ // 5 puan

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

SADECE verilen iki
fonksiyonun içini
doldurunuz



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

SADE
KOMUTLAR/İFADELER
kullanmanız
beklenmektedir.
Örneğin, 1 **printf** ile
yapılabilecek kısmı 10
printf ile yaparsanız
kodunuz doğru çalışsa
bile puan kaybı
yaşayabilirsiniz.

} //Fonksiyon kapatma süslü parantezi - ayrıca başka bir süslü parantez kullanmayınız.

int min2 (**int** dizi [7] , **int** min_indis) //kendisine gönderilen 7 elemanlı dizinin en küçük 2. elemanının indisini döner.

{ // 5 puan

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.

} //Fonksiyon kapatma süslü parantezi - ayrıca başka bir süslü parantez kullanmayınız.

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. İyi dinlenceler.

Şirin Baba: Ben uyumuyor muydum? Sen nereden çıktın?

Usta Şirin: Sana soru sormaya gelmiştim. Uyuduğunu görünce, yeni icadım şirinceptionla rüyana bağlandım. Birkaç soru sorup gideceğim. Bu sefer dönem başından beri düzenli çalıştım; tüm ödevleri yaptım, SYÇS'leri çözdüm, codingame'leri yaptım, bol bol kod yazdım... İstersen çalışmalarımı göstereyim. İnan bana Şirin Baba.

Şirin Baba: Rüyamda görsem inanmam Usta Şirin. Uyanınca gösterirsin. Sen sorularını sor hemen.

Usta Şirin: C dilinde değişkenime şu isimlerden hangisini ya da hangilerini versem C geçerli kabul eder? Altına numaralar da yazdım, böylece değişken isimlerini yazmana gerek yok, numaralarını söylersen ben anlarım.

kaLem	%agirlik	3BoyutluYazici	Continue	break	se7en
1	2	3	4	5	6

Şirin Baba: C dili şu üç numarayla(küçükten büyüğe sırasıyla) eşleşmiş ismi geçerli kabul eder: _____, _____, _____.

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

```
if( sayi > 100 )
    printf("Üç basamakli");
else ( sayi < 100 )
    printf("İki basamakli");
```

yazdım. Kodum, ekrana *sayi*'nin basamak adedini yazsın istiyorum. *sayi* değişkeni [10-999] arasında değer içeren bir değişkendir. Bu kod parçası doğru çalışır mı?

_____ : Üç basamaklı olan 100'ü de dahil etmek üzere *>100* ifadesi yerine *>=100* ifadesi yazarsan kod istediğin gibi doğru çalışacaktır.

Usta Şirin: Şunlardan hangisi geçerli bir dizgi fonksiyonu ismi **değildir**: *strtok*, *strlwr*, *strrev*, *strnet*, *strstr* ?

Şirin Baba: _____ diye bir fonksiyon yoktur. Diğer dördü *string.h* içinde bulabileceğin fonksiyonlardır.

Usta Şirin: Şirin Baba, `strcmp ( "Sirine" , "Sirine" )` fonksiyonu ne yapar ve ne değeri döner?

_____ : İki dizgiyi kıyaslar ve ikisi de aynı olduğu için olumlu anlamında 0 değerini döner.

Usta Şirin: Elimdeki bir diziyi küçükten büyüğe sıralatmak için kabarcık sıralama algoritmasını kullanmak istiyorum. Bu algoritmanın ne gibi eksiklikleri vardır?

_____ : Bu algoritmanın iki büyük eksiği vardır. Birincisi, kontrol esnasında yaptığı her geçişte yan yana olan tüm sayı çiftlerine bakmak zorundadır ve ikincisi, dizi sıralı olsa da olmasa da aynı miktarda kıyaslama işlemini bitirmeden sonlanamamaktadır.

Usta Şirin: Aşçı Şirin, bir dizi açmış, içine de yeni yaptığı tatlının harflerini değer olarak atamış. "Sence çalışır mı?" diye bana sordu. Sonunda noktalı virgül vardı, köşeli parantezler içerisi boştu. Tam olarak şöyle bir kullanım:

`char dizgi[ ] = { s, u, f, l, e };`

_____ : Bu doğru bir kullanım değildir. Söz dizim hatası verir.

Usta Şirin: İlgili kütüphaneler ve `srand` komutu eklendikten sonra kod içinde kullanılan bir `( rand() + 10 ) % 141` ifadesi hangi aralıkta bir değer üretir?

Şirin Baba: Bu ifade minimum _____ ve maksimum _____ arasında değer üretir. Bu iki sayıdan birini ya da bu iki sayı arasındaki herhangi bir sayıyı üretebilir.

Usta Şirin: Taslak olarak sağdaki gibi bir kodum var. if kısmının ortasında, ÇIKIŞ İFADESİ olarak gösterdiğim yere bir komut yazmak istiyorum. Bu komuta gelindiğinde, kod satır X olarak ifade ettiğim yere gitsin ve oradan devam etsin istiyorum. Bunu nasıl sağlarım?

_____ : // ÇIKIŞ İFADESİ yerine `break;` yazman yeterli olur. `break;` komutu `if else`'i kırar ve sonrasındaki satır X'e kodu atlatır.

```
// kullanıcıdan yas değerinin alımı
if(yas>10)
{
    // kod parçası 1
    // ÇIKIŞ İFADESİ
    // kod parçası 2
}
else
{
    // kod parçası 3
}
// satır X
// kod parçası 4
```

Usta Şirin: Ekrana bir şekil çizdirmek istediğimde imleci tek komutla ekranda bir yere götürmek istiyorum. Bunun bir çözümü var mıdır?

_____ : C dilinde tek çözüm ekrana gerektiği kadar "`\n`" ve boşluk bastırarak imleci istediğin yere sürüklemektir. Örneğin, içi boş bir üçgen bastırıcaksan, şeklin içindeki boşlukları da ekrana bastırmalısın.

Usta Şirin: Sınava çalışırken şirinsayırimda şöyle 3 boyutlu bir dizgi dizisi tanımına ve değer atamasına rastladım:

```
char s[10][10][10][10]={{{"asci","uyku","gozl"}},{},{{"usta","srne"},{}},{{"azmn"}},{{"grgl","tmbl"},{"mzkc"}}};
```

Önce çok korktum, ama sonra "bu satırdan sonra printf("%c", s[2][0][1][3]); komutunu kullansam ekrana ne yazdırır?" diye merak etmeye başladım. 3 boyutlu dizi izlerken kullandığım gözlüğümü de gözüme taktım ama yine çözemedim.

Şirin Baba: Parantezleri dikkatlice takip edebilirsin. Kolaylık olması için parantez çiftlerinin üstlerine 1, 2 gibi ayırt edici numaralar yazarak eşleşen parantezleri belirtebilirsin ve bir de aynı kodu daha az korkutucu olan şu şekle getirebilirsin:

```
char s[10][10][10][10] =
{
    { {"asci" , "uyku" , "gozl" } },
    { },
    { {"usta" , "srne" } , { } , {"azmn" } },
    { {"grgl" , "tmbl" } , {"mzkc" } }
};
```

Çıktısını merak ettiğin kullanıma gelecek olursak, bahsettiğin printf komutu ekrana _____ yazdırır.

Usta Şirin: *I8->D* diye bir ifade gördüm. Yandan bakınca bir şirine benziyor. Bu nasıl bir kod Şirin Baba?

_____ : Bu ifadeden *I8*'in kesinlikle bir işaretçi olduğu anlaşılıyor. *I8*'in işaret ettiği yapı değişkeninin içindeki *D* isimli değişkene erişim sağlanmış.

Usta Şirin: Peki öyleyse, typedef ile şöyle bir yapı tanımladım: `typedef struct{ int D; }smurf;`

Şimdi main'in içerisinde `smurf* I8; I8->D = 7;` şeklindeki iki ifadeyle tanım ve değer ataması yapabilir miyim?

_____ : Hayır, yapamazsın.

Usta Şirin: Çok soru sordum. Rüyanı kabaşa çevirmeden önce son sorumu da sorayım ve ben gideyim.

```
int s = 8, r = 10;
if( s != 4 )
    if( r = 0 )
        printf("%d", s );
    else printf("%d", r / 3 + 7 );
```

Yukarıdaki kod parçası ekrana ne yazdırır?

Şirin Baba: Burada "*r=0*" ifadesinin olmadığına dikkatini çekerim. Bu kod parçası ekrana _____ yazdırır.

Usta Şirin: Çok teşekkür ederim Şirin Baba. Aslında senin bilincini bir oyuncak bebeğe kopyalayıp, seni de bu işten kurtarmak niyetindeyim. *siring.h* kütüphanesi içerisine konulacak *srncpy* isimli bir fonksiyon üzerine çalışıyorum.

Şirin Baba: Usta, onun yerine lütfen sınava tüm dikkatini ver, soruları dikkatlice oku, bildiklerini kağıda doğru şekilde yansıt, bu sınavdan harika bir not al ve ben de artık rüyama dönüp biraz dinleneyim.

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

KOD - Çalıştığında ekrana "<22 t1>" yazdıran kod	EKRAN ÇIKTISI
<pre>#include <stdio.h> #define BOSLUK _____ // ipucu: yukarıdaki boşluğun cevabının sonunda ";" yoktur. typedef union { int fiyat; }meyve; typedef union { _____ B[1]; }sepet; int main() // ipucu:22'yi saklayacak bir değişkene ihtiyacınız var ve // ... elinizdeki tek değişken G. { sepet G; BOSLUK = 22; printf("<%d t1>", BOSLUK); return 0; }</pre>	<pre><22 t1> ----- Process exited after 0.007963 seconds with return value 0</pre> ÖRNEK EKRAN ÇIKTISI Kod çalıştırıldıktan sonra herhangi bir kullanıcı girişi olmamıştır ve kod görüldüğü gibi bir ekran çıktısı vermiştir. Kod çalıştırıldığında derleyici herhangi bir uyarı ya da hata vermemelidir. Soruda boşlukların dışını doldurmayınız. Sadece boşluklara yazdıklarınız değerlendirmeye alınır.

KOD - Çalıştığında ekrana "10 160" yazdıran kod	EKRAN ÇIKTISI
<pre>#include <stdio.h> void fonk(int d[2][2], int *r); int main() { int dizi[2][2]={ { 1 , 2 } , { 3 , 4 } }; int toplam = 0; fonk(_____, _____); printf("%d ", toplam); int array[2][2]={ { 40 , 10 } , { 60 , 50 } }; toplam = 0; fonk(_____, _____); printf("%d", toplam); return 0; } void fonk(int d[2][2], _____) { int i, j; for(i = 0 ; i < 2 ; i++) for(j = 0 ; j < 2 ; j++) *r _____ d[j][i]; //yukarıdaki boşlugun cevabı sadece 2 karakter içermelidir. }</pre>	<pre>10 160 ----- Process exited after 0.005241 seconds with return value 0</pre> ÖRNEK EKRAN ÇIKTISI Kod çalıştırıldıktan sonra herhangi bir kullanıcı girişi olmamıştır ve kod görüldüğü gibi bir ekran çıktısı vermiştir. 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 - Kullanıcının belirttiği sayı miktarınca satırdan oluşan içi boş üçgen çizdiren kod

```
#include <stdio.h>
void ekrana_bastir( char , int );
int main()
{
    int n;
    printf("Satir sayisi:");
    scanf("%d", &n);
    int sol_bosluk_adet = n-1;
    int orta_bosluk_adet = _____ ;
    int i;
    for( i = 1 ; i <= n ; i++ )
    {
        if( _____ )
        { //Aşağıdaki tek tırnaklar arasında 1 boşluk vardır.
            ekrana_bastir(' ', sol_bosluk_adet);
            ekrana_bastir('*', 1);
        }
        else if ( i == n )
        {
            ekrana_bastir('*', _____ );
        }
        else
        {
            ekrana_bastir(' ', sol_bosluk_adet);
            ekrana_bastir('*', 1);
            ekrana_bastir(' ', orta_bosluk_adet);
            ekrana_bastir('*', 1);
        }
        sol_bosluk_adet = sol_bosluk_adet _____;
        orta_bosluk_adet = orta_bosluk_adet + 2;
        printf("\n");
    }
    return 0;
}

void ekrana_bastir( char k, int adet )
{
    int i;
    for( i = 0 ; i < _____ ; i++ )

        printf("%c ", _____); //'%c'den sonra 1 boşluk var.
}
```

EKRAN ÇIKTISI

```
Satir sayisi:3
  *
 * *
* * * * *
-----
Process exited after 1.654 seconds with
return value 0
```

ÖRNEK EKRAN ÇIKTISI

Kod çalıştırıldıktan sonra kullanıcı "3" yazıp enter'a basmıştır.
Kod 3 satırdan oluşan şekildeki içi boş üçgeni bastırmıştır.

```
Satir sayisi:10
          *
        * *
      * * *
    * * * *
  * * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
-----
Process exited after 2.33 seconds with
return value 0
```

ÖRNEK EKRAN ÇIKTISI

Kod çalıştırıldıktan sonra kullanıcı "10" yazıp enter'a basmıştır.
Kod 10 satırdan oluşan şekildeki içi boş üçgeni bastırmıştır.

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.

Soruda boşlukların dışını doldurmayınız. Sadece boşluklara yazdıklarınız değerlendirmeye alınır.

KOD - Kullanıcının belirttiği sayı miktarınca satırdan oluşan bir paralel kenarı ve paralel kenarın bir köşegenini (başka bir ifadeyle bitişik şekilde içi boş iki üçgen) çizdiren kod

```
#include <stdio.h>
int main()
{
    int i, j, n;
    printf("Satır sayısı:"); scanf("%d", &n);
    for( i = 0; i < n; i++ )
    {
        for( j = 0 ; j <= 3*(n-1); j++ )

            if( i == 0 && _____ )
                printf(" * "); //bir yıldız ve bir boşluk

            else if( i == n-1 && _____ )
                printf(" * "); //bir yıldız ve bir boşluk

            else if( i + j == n-1 || _____ == 3*(n-1) || j - i == n-1 )
                printf(" * "); //bir yıldız ve bir boşluk

            _____ printf(" "); //iki boşluk

        printf("\n");
    }
    return 0;
}
```

EKLAN ÇIKTISI

Satır sayısı:4

```

      * * * *
     * * * *
    * * * *
   * * * *
  
```

ÖRNEK EKLAN ÇIKTISI

Kod çalıştırıldıktan sonra kullanıcı "4" yazıp enter'a basmıştır. Kod 4 satırdan oluşan, çıktındaki şekli bastırmıştır.

Satır sayısı:5

```

      * * * * *
     * * * * *
    * * * * *
   * * * * *
  * * * * *
 
```

ÖRNEK EKLAN ÇIKTISI

Kod çalıştırıldıktan sonra kullanıcı "5" yazıp enter'a basmıştır. Kod 5 satırdan oluşan, çıktındaki şekli bastırmıştır.

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.

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 ALABİLMEYİ 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 dizi[7] = {};
    int i;
    for( i = 0 ; i < 7 ; i++ )
        dizi[i] = ( i * i ) % 5;
    printf( "%d\n" , i );
    do
    {
        dizi[ i - 2 ] += dizi[ i - 1 ];
        i--;
    }
    while( i > 1 );
    for( i = 0 ; i != 9 ; i += 3 )
        printf( "%d" , dizi[ i ] );
    // printf'lerden sadece birinde \n var.
    return 0;
}
```


```
#include <stdio.h> //sprintf'i de içerir.
void yaz( char [ ] , int );
int main()
{
    char dizgi[100] = "AA";
    sprintf( dizgi , "%s" , "12345" );
    yaz ( dizgi , 2 );
    return 0;
}
void yaz( char e[ ] , int f )
{
    if( f == 0 || f == 6 )
        return;
    printf( "%c" , e[f] != '\0' ? e[f] : e[f-1] );
    yaz ( e , f+1 );
    return;
}
```


```
#include <stdio.h>
typedef struct
{
    int gun , ay , yıl ;
} date;
typedef struct
{
    date tarih;
    int pasta;
} birth;
int main()
{
    birth can = { { 9 , 2 , 1999 } , 4 };
    int i;
    for( i = 0 ; i < can.pasta ; i++ )
        printf( "%d" , can.tarih.ay );
    return 0;
}
```


```
#include <stdio.h>
int fn ( int a );
int main()
{
    int e = 23;
    while( 1 == 1 )
    {
        e += 2;
        if( fn(e) == 1 )
            break;
    }
    printf( "%d" , e );
    return 0;
}
int fn ( int a )
{
    int i;
    for( i = 2 ; i < a ; i++ )
        if( a % i == 0 )
            return 0;
    return 1;
}
```


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

```
int main()
```

```
{
```

```
    int s = 1, r = 1;
```

```
    do
```

```
    {
```

```
        for( r = 1 ; r <= 8 ; r++ )
```

```
        {
```

```
            if( s + r == 4 )
```

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

```
            else if ( s - r == -4 )
```

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

```
            else
```

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

```
        }
```

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

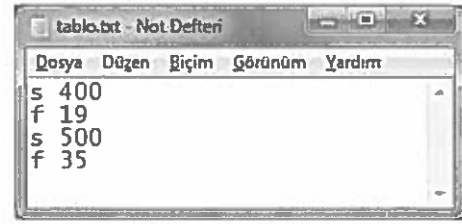
```
        s++;
```

```
    }
```

```
    while( s <= 3 );
```

```
    return 0;
```

```
}
```

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

```
typedef struct
```

```
{
```

```
    int a;
```

```
    char b;
```

```
} t;
```

```
int main()
```

```
{
```

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

```
    int i, son=3;
```

```
    t r;
```

```
    son = son % 2;
```

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

```
    {
```

```
        fscanf(dosya, " %c", &r.b);
```

```
        // %c'den önce boşluk var.
```

```
        fscanf(dosya, "%d", &r.a);
```

```
    }
```

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

```
        printf("%c%d\n", r.b, r.a);
```

```
    return 0;
```

```
}
```


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

int main()
{
    char isim[100], name[100];
    scanf("%s", isim);
    printf("%s\n", isim);
    gets(name);
    printf("%s\n", name+1);
    return 0;
}
```

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

```
typedef struct
```

```
{
```

```
    int a;
```

```
    char b;
```

```
} t;
```

```
char fonk( t* );
```

```
int main()
```

```
{
```

```
    t r;
```

```
    r.a = 5;
```

```
    r.b = fonk( &r );
```

```
    printf("%d%c\n", r.a , r.b );
```

```
    fonk( &r );
```

```
    printf("%d%c\n", r.a , r.b );
```

```
    return 0;
```

```
}
```

```
char fonk( t* z )
```

```
{
```

```
    (z->a)++;
```

```
    z->b = 'e';
```

```
    return 'h';
```

```
}
```


```
//kod.c dosyası
```

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

```
#include <stdlib.h>
```

```
typedef union // union değil!
```

```
{
```

```
    int a,b;
```

```
} c;
```

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

```
{
```

```
    c.d = {};
```

```
    d.a = argc;
```

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

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

```
    d.b = atoi ( argv[1] );
```

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

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

```
//bu kodda hiç '\n' yoktur.
```

```
    return 0;
```

```
}
```

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

Komut Ekranı

gcc	kod.c	-o	kod.exe						
kod	9								

//kod.c dosyası

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

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

Komut Ekranı

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

```
#include <stdio.h>
void fonk( int );
int main()
{
    int sayac = 5;
    fonk( sayac );
    return 0;
}
void fonk( int sayac )
{
    if( sayac == 0 )
        return;
    printf("%d\n", --sayac );
    fonk( sayac-1 );
    return;
}
```

```
#include <stdio.h>
void fonk( int x );
int main()
{
    fonk( 4 );
    return 0;
}
void fonk(int x)
{
    if( x != 7 )
    {
        fonk( x+1 );
        printf("%d", x);
        fonk( x+1 );
    }
    return;
}
```

```
#include <stdio.h>
int fonk( int x );
int main()
{
    int a = 7;
    int b[3] = { 1 , 2 , 3 };
    fonk( a );
    printf("%d", a);
    fonk( b[1] );
    printf("%d", b[1]);
    return 0;
}
int fonk( int x )
{
    x = x + 2;
    return x;
}
```

```
#include <stdio.h>
void y ( int [ ] );
int main()
{
    int d[9] = {1,2,3,4,5,6,7,8,9};
    y( d );
    int i;
    for( i = 0 ; i < 9 ; i++ )
    {
        printf("%d", d[i] );
        if( i == 4 )
            printf("\n");
    }
    return 0;
}
void y ( int dizi[ ] )
{
    int b, i;
    do
    {
        b = 1;
        i = 3;
        while( i < 8 )
        {
            if( dizi[ i ] > dizi[ i - 1 ] )
            {
                int t = dizi[ i ];
                dizi[ i ] = dizi[ i - 1 ];
                dizi[ i - 1 ] = t;
                b = 0;
            }
            i++;
        }
    }
    while( b == 0 );
}
```


```
#include <stdio.h>
#include <string.h>
void a( char s[ ] );
void b( char s[ ] );
void c( char s[ ] );
int main()
{
    char s[100] = "97";
    int i;
    for( i = 1 ; i <= 35 ; i++ )
        switch( i%3 )
        {
            case 0:
                a(s);
                break;
            case 1:
                b(s);
                break;
            default:
                c(s);
                break;
        }
    printf("%s", s);
    return 0;
}
void a( char s[ ] )
{
    char w[100] = "a";
    strcpy(s,w);
    strcpy(w,s);
}
void b( char s[ ] )
{
    char w[100] = "b";
    strcat(w,s);
    strcat(s,w);
}
void c( char s[ ] )
{
    char w[100] = "can";
    strncat(w,s,2);
    strncat(s,w,2);
}
```


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. Dersin öğretim kadrosu (hocalar ve asistanlar) hakkında neler düşünüyorsunuz?

4. Uygulamalı derslerin yeterince faydalı olduğunu düşünüyor musunuz? Nasıl daha geliştirilebilir?

5. Dönemin ortasından sonraki ödevlere katılım nasıl daha arttırılabilir(ödev yüzdelerini arttırmadan)?

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 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 çıktısını 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.

Bu sorudaki kodun bir kısmı verilmiştir. Sizden istenen kodda geçen iki fonksiyonun (*min* ve *min2*) içini tamamlamanızdır.

- Kod çalışmaya başladığında örnek ekran çıktısında görüldüğü gibi kullanıcıdan ödev notu girişleri alacaktır. Kullanıcı notları tek tek girip enter'a basacaktır. Tüm notlar birbirinden farklıdır ve her biri **[0,215]** aralığındadır. Kullanıcının her zaman geçerli girişler yapacağını varsayınız.
- Kod, bu notlar arasındaki **en küçük ve 2. en küçük** notları ekrana yazdıracaktır. Ardından, **tüm notları** ekrana yazdıracaktır.

```
Odev 1 not girisi: 100
Odev 2 not girisi: 115
Odev 3 not girisi: 120
Odev 4 not girisi: 106
Odev 5 not girisi: 149
Odev 6 not girisi: 165
Odev 7 not girisi: 215

En küçük notlu odev-> 100
2. en küçük notlu odev-> 106
Tumu-> 100 115 120 106 149 165 215
-----
Process exited after 33.55 seconds
Devam etmek için bir tuşa basın .
```

ÖRNEK EKRAN ÇIKTISI

Kod çalıştıktan sonra kullanıcı sırasıyla 100, 115, 120, 106, 149, 165 ve 215 girişleri yapıp her girişten sonra enter'a basmıştır. Bu sayılar arasında en küçük olanı 100 olduğu için 100, çıktıda gözüktüğü şekilde ekrana yazdırılmıştır. Ardından 100 haricindeki en küçük değer olan 106 ekrana yazdırılmıştır. Son olarak, kullanıcının girdiği tüm sayılar ekranda gösterilmiştir.

ÖNEMLİ UYARILAR:

- Sorunun başında da belirtildiği üzere **sade komutlar/ifadeler** kullanmanız beklenmektedir. Örneğin, 1 printf ile yapılabilecek kısmı 10 printf ile yaparsanız kodunuz doğru çalışsa bile puan kaybı yaşayabilirsiniz.
- Kod içerisinde **boşluk** bırakmanın kritik olduğu kısımlardaki boşluklarınız **açık olmalı**. 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**.

```
#include <stdio.h>
int min (int dizi [ 7 ] );
int min2 (int dizi [ 7 ] , int min_indis );
int main( )
{
    int notlar [ 7 ], i;
    for( i = 0 ; i < 7 ; i++ )
    {
        printf("Odev %d not girisi: ", i+1 );
        scanf("%d", &notlar [ i ] );
    }
    int min_indis = min ( notlar );
    int min2_indis = min2 ( notlar , min_indis );
    printf("\nEn küçük notlu odev-> %d\n", notlar [ min_indis ] );
    printf("2. en küçük notlu odev-> %d\n", notlar [ min2_indis ] );
    printf("Tumu-> ");
    for( i = 0 ; i < 7 ; i++ )
        printf("%d ", notlar [ i ] );
    return 0;
} //Kodun devamı bir sonraki sayfadadır.
```

KOD SOL TARAFTA VE BİR
SONRAKİ SAYFADA
GÖSTERİLMİŞTİR.
İSTENENLERİ SAĞLAYACAK
ŞEKİLDE EKSİK KISIMLARINI
DOLDURUNUZ.

//Önceki sayfada baş kısmı verilen kodun devamı

int min (int dizi [7]) // kendisine gönderilen 7 elemanlı dizinin en küçük elemanının indisini döner.

{ // 5 puan

```
int i, indis, min_not = 999;
for(i=0; i < 7; i++)
{
    if(dizi[i] < min_not)
    {
        min_not = dizi[i];
        indis = i;
    }
}
return indis;
```

Kabarcık sıralama;

1) Diziyi koluca olarak bozardı.

2) Gereksiz işlem yapılıyordu

uygun bir seçim değil

} //Fonksiyon kapatma süslü parantezi - ayrıca başka bir süslü parantez kullanmayınız

int min2 (int dizi [7] , int min_indis) //kendisine gönderilen 7 elemanlı dizinin en küçük 2. elemanının indisini döner.

{ // 5 puan

```
int i, indis, min_not = 999;
for(i=0; i < 7; i++)
{
    if( i != min_indis && dizi[i] < min_not)
    {
        min_not = dizi[i];
        indis = i;
    }
}
return indis;
```

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.

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

SADECE verilen iki fonksiyonun içeriğini doldurunuz



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

SADE KOMUTLAR/İFADELER kullanmanız beklenmektedir. Örneğin, 1 printf ile yapılabilecek kısmı 10 printf ile yaparsanız kodunuz doğru çalışsa bile puan kaybı yaşayabilirsiniz.

} //Fonksiyon kapatma süslü parantezi - ayrıca başka bir süslü parantez kullanmayınız.

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. İyi dinlenceler.

Şirin Baba: Ben uyumuyor muydum? Sen nereden çıktın?

Usta Şirin: Sana soru sormaya gelmiştim. Uyuduğunu görünce, yeni icadım şirinceptionla rüyana bağlandım. Birkaç soru sorup gideceğim. Bu sefer dönem başından beri düzenli çalıştım; tüm ödevleri yaptım, SYÇS'leri çözdüm, codingame'leri yaptım, bol bol kod yazdım... İstersen çalışmalarımı göstereyim. İnan bana Şirin Baba.

Şirin Baba: Rüyamda görsem inanmam Usta Şirin. Uyanınca gösterirsin. Sen sorularını sor hemen.

Usta Şirin: C dilinde değişkenime şu isimlerden hangisini ya da hangilerini versem C geçerli kabul eder? Altına numaralar da yazdım, böylece değişken isimlerini yazmana gerek yok, numaralarını söylersen ben anlarım.

kaLem	%agirlik	3BoyutluYazici	Continue	break	se7en
1	2	3	4	5	6

Şirin Baba: C dili şu üç numarayla(küçükten büyüğe sırasıyla) eşleşmiş ismi geçerli kabul eder: 1, 4, 6.

Usta Şirin: Uygun kütüphaneleri vs. ekledikten sonra main içerisine

```

if( sayi > 100 )
    printf("Üç basamaklı");
else( sayi < 100 )
    printf("İki basamaklı");

```

yazdım. Kodum, ekrana sayi'nin basamak adedini yazsın istiyorum. sayi değişkeni [10-999] arasında değer içeren bir değişkendir. Bu kod parçası doğru çalışır mı?

Gargamel: Üç basamaklı olan 100'ü de dahil etmek üzere >100 ifadesi yerine >=100 ifadesi yazarsan kod istediğin gibi doğru çalışacaktır.

Usta Şirin: Şunlardan hangisi geçerli bir dizgi fonksiyonu ismi değildir: strtok, strlwr, strrev, strnet, strstr ?

Şirin Baba: strnet diye bir fonksiyon yoktur. Diğer dördü string.h içinde bulabileceğin fonksiyonlardır.

Usta Şirin: Şirin Baba, `strcmp ( "Sirine" , "Sirine" )` fonksiyonu ne yapar ve ne değeri döner?

Şirin Baba: İki diziyi kıyaslar ve ikisi de aynı olduğu için olumlu anlamında 0 değerini döner.

Usta Şirin: Elimdeki bir diziyi küçükten büyüğe sıralatmak için kabarcık sıralama algoritmasını kullanmak istiyorum. Bu algoritmanın ne gibi eksiklikleri vardır?

Gargamel: Bu algoritmanın iki büyük eksiği vardır. Birincisi, kontrol esnasında yaptığı her geçişte yan yana olan tüm sayı çiftlerine bakmak zorundadır ve ikincisi, dizi sıralı olsa da olmasa da aynı miktarda kıyaslama işlemini bitirmeden sonlanamamaktadır.

→ zorunda değildir.
→ sonuna bitmektedir.

Usta Şirin: Aşçı Şirin, bir dizi açmış, içine de yeni yaptığı tatlının harflerini değer olarak atamış. "Sence çalışır mı?" diye bana sordu. Sonunda noktalı virgül vardı, köşeli parantezler içerisi boştu. Tam olarak şöyle bir kullanım:

`char dizgi[] = { s, u, f, l, e };`

Şirin Baba: Bu doğru bir kullanım değildir. Söz dizim hatası verir.

Usta Şirin: İlgili kütüphaneler ve `rand` komutu eklendikten sonra kod içinde kullanılan bir ifade hangi aralıkta bir değer üretir?

En son 0/141 [0-140]a kırparaktır.

[0-32767?]

[10-32787?]

Şirin Baba: Bu ifade minimum 0 ve maksimum 140 arasında değer üretir. Bu iki sayıdan birini ya da bu iki sayı arasındaki herhangi bir sayıyı üretebilir.

Usta Şirin: Taslak olarak sağdaki gibi bir kodum var. if kısmının ortasında, ÇIKIŞ İFADESİ olarak gösterdiğim yere bir komut yazmak istiyorum. Bu komuta gelindiğinde, kod satır X olarak ifade ettiğim yere gitsin ve oradan devam etsin istiyorum. Bunu nasıl sağlarım?

Gargamel: // ÇIKIŞ İFADESİ yerine `break;` yazman yeterli olur. `break;` komutu `if else`'i kırar ve sonrasındaki satır X'e kodu atlatır.

X → switch ya da döngüye kırar.

```
// kullanıcıdan yas değerinin alımı
if(yas>10)
{
    // kod parçası 1
    // ÇIKIŞ İFADESİ
    // kod parçası 2
}
else
{
    // kod parçası 3
}
// satır X
// kod parçası 4
```

Usta Şirin: Ekrana bir şekil çizdirmek istediğimde imleci tek komutla ekranda bir yere götürmek istiyorum. Bunun bir çözümü var mıdır?

Gargamel: C dilinde tek çözüm ekrana gerektiği kadar "\n" ve boşluk bastırarak imleci istediğin yere sürüklemektir. Örneğin, içi boş bir üçgen bastırıcaksan, şeklin içindeki boşlukları da ekrana bastırmalısın.

→ windows.h içinde imleci oynatabiliyoruz ekranda ile (bkz. `imlecxy` fonk.)

Usta Şirin: Sınava çalışırken şirinsayarımında şöyle 3 boyutlu bir dizgi dizisi tanımına ve değer atamasına rastladım:

```
char s[10][10][10][10] = {{"asci", "uyku", "gozl"}, {}, {"usta", "srne"}, {}, {"azmn"}, {"grgl", "tmbl"}, {"mzkc"}};
```

Önce çok korktum, ama sonra "bu satırdan sonra printf("%c", s[2][0][1][3]); komutunu kullansam ekrana ne yazdırır?" diye merak etmeye başladım. 3 boyutlu dizi izlerken kullandığım gözlüğümü de gözüme taktım ama yine çözemedim.

Şirin Baba: Parantezleri dikkatlice takip edebilirsin. Kolaylık olması için parantez çiftlerinin üstlerine 1, 2 gibi ayırt edici numaralar yazarak eşleşen parantezleri belirtebilirsin ve bir de aynı kodu daha az korkutucu olan şu şekle getirebilirsin:

```
char s[10][10][10][10] =
{
    { {"asci", "uyku", "gozl"} },
    { },
    { {"usta", "srne"}, {}, {"azmn"} },
    { {"grgl", "tmbl"}, {"mzkc"} }
};
```

Handwritten notes: s[2][0][1][3] ailesi, "srne" → s[2][0][1][3], s[2][0][1][3] ailesi

Çıktısını merak ettiğin kullanıma gelecek olursak, bahsettiğin printf komutu ekrana e yazdırır.

Usta Şirin: $I8 \rightarrow D$ diye bir ifade gördüm. Yandan bakınca bir şirine benziyor. Bu nasıl bir kod Şirin Baba?

Şirin Baba: Bu ifadeden $I8$ 'in kesinlikle bir işaretçi olduğu anlaşılıyor. $I8$ 'in işaret ettiği yapı değişkeninin içindeki D isimli değişkene erişim sağlanmış.

Usta Şirin: Peki öyleyse, typedef ile şöyle bir yapı tanımladım: typedef struct{ int D; }smurf;

Şimdi main'in içerisinde smurf* $I8$; $I8 \rightarrow D = 7$; şeklindeki iki ifadeyle tanım ve değer ataması yapabilir miyim?

Şirin Baba: Hayır, yapamazsın. *Handwritten notes:* Bu an bir yere işaret etmiyor. $I8 = \& \dots$; eksik.

Usta Şirin: Çok soru sordum. Rüyanı kabusu çevirmeden önce son sorumu da sorayım ve ben gideyim.

```
int s = 8, r = 10;
if( s != 4 )
    if( r = 0 )
        printf("%d", s );
    else printf("%d", r / 3 + 7 );
```

Handwritten notes: r=0 ata. if(0) var p.b.i devam et else'ye git.

Yukarıdaki kod parçası ekrana ne yazdırır?

Handwritten: 0/3 + 7 → 7

Şirin Baba: Burada " $r=0$ " ifadesinin olmadığına dikkatini çekerim. Bu kod parçası ekrana 7 yazdırır.

Usta Şirin: Çok teşekkür ederim Şirin Baba. Aslında senin bilincini bir oyuncak bebeğe kopyalayıp, seni de bu işten kurtarmak niyetindeyim. siring.h kütüphanesi içerisine konulacak srncpy isimli bir fonksiyon üzerine çalışıyorum.

Şirin Baba: Usta, onun yerine lütfen sınava tüm dikkatini ver, soruları dikkatlice oku, bildiklerini kağıda doğru şekilde yansıt, bu sınavdan harika bir not al ve ben de artık rüyama dönüp biraz dinleneyim.

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

KOD - Çalıştırıldığında ekrana "<22 t1>" yazdıran kod	EKRAN ÇIKTISI
<pre>#include <stdio.h> #define BOSLUK <u>G.B[0].fiyat</u> // ipucu: yukarıdaki boşluğun cevabının sonunda ";" yoktur. typedef union { int fiyat; }meyve; typedef union { <u>meyve</u> B[1]; }sepet; int main() // ipucu:22'yi saklayacak bir değişkene ihtiyacınız var ve // ... elinizdeki tek değişken G. { sepet G; BOSLUK = 22; printf("<%d t1>", BOSLUK); return 0; }</pre> <i>G: türü sepet B[0]: türü meyve fiyat: türü int</i>	<pre><22 t1> ----- Process exited after 0.007963 seconds with return value 0</pre> ÖRNEK EKRAN ÇIKTISI Kod çalıştırıldıktan sonra herhangi bir kullanıcı girişi olmamıştır ve kod görüldüğü gibi bir ekran çıktısı vermiştir. Kod çalıştırıldığında derleyici herhangi bir uyarı ya da hata vermemelidir. Soruda boşlukların dışını doldurmayınız. Sadece boşluklara yazdıklarınız değerlendirmeye alınır.

KOD - Çalıştırıldığında ekrana "10 160" yazdıran kod	EKRAN ÇIKTISI
<pre>#include <stdio.h> void fonk(int d[2][2], int *r); int main() { int dizi[2][2]={{ 1 , 2 }, { 3 , 4 }}; int toplam = 0; fonk(<u>dizi</u> , <u>&toplam</u>); printf("%d ", toplam); int array[2][2]={{ 40 , 10 }, { 60 , 50 }}; toplam = 0; fonk(<u>array</u> , <u>&toplam</u>); printf("%d", toplam); return 0; } void fonk(int d[2][2], <u>int * r</u>) { int i, j; for(i = 0 ; i < 2 ; i++) for(j = 0 ; j < 2 ; j++) *r <u>+=</u> d[j][i]; //yukarıdaki boşluğun cevabı sadece 2 karakter içermelidir. }</pre> <i>adres ile gönderim</i>	<pre>10 160 ----- Process exited after 0.005241 seconds with return value 0</pre> ÖRNEK EKRAN ÇIKTISI Kod çalıştırıldıktan sonra herhangi bir kullanıcı girişi olmamıştır ve kod görüldüğü gibi bir ekran çıktısı vermiştir. Gereksiz komut/ifadeler eklemeyiniz. Bu tür durumlarda kodunuz sonra herhangi 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 - Kullanıcının belirttiği sayı miktarınca satırdan oluşan içi boş üçgen çizdiren kod

```
#include <stdio.h>
void ekrana_bastir( char , int );
int main()
{
    int n;
    printf("Satir sayisi:");
    scanf("%d", &n);
    int sol_bosluk_adet = n-1;
    int orta_bosluk_adet = -1 ;
    int i;
    for( i = 1 ; i <= n ; i++ )
    {
        if( i == 1 )
        { //Aşağıdaki tek tırnaklar arasında 1 boşluk vardır.
            ekrana_bastir( ' ', sol_bosluk_adet );
            ekrana_bastir( '*', 1 );
        }
        else if ( i == n )
        {
            ekrana_bastir( '*', 2*n-1 );
        }
        else
        {
            ekrana_bastir( ' ', sol_bosluk_adet );
            ekrana_bastir( '*', 1 );
            ekrana_bastir( ' ', orta_bosluk_adet );
            ekrana_bastir( '*', 1 );
        }
        sol_bosluk_adet = sol_bosluk_adet - 1 ;
        orta_bosluk_adet = orta_bosluk_adet + 2;
        printf("\n");
    }
    return 0;
}

void ekrana_bastir( char k, int adet )
{
    int i;
    for( i = 0 ; i < adet ; i++ )
    {
        printf("%c ", k ); //"%c'den sonra 1 boşluk var.
    }
}
```

** 2 yıldız arası boşluk adedi. Her satırda değeri 2 artıyor ve 2. satırda değeri 1 ise, en başta -1 olmalı.*

ilk satırdaysak bunu yap.

son satırdaysak bunu yap.

bu kadar yıldız bas.

adet tane k bastırır.

EKRAN ÇIKTISI

```
Satir sayisi:3
*
* *
* * *

-----
Process exited after 1.654 seconds with return value 0
```

ÖRNEK EKRAN ÇIKTISI

Kod çalıştırıldıktan sonra kullanıcı "3" yazıp enter'a basmıştır. Kod 3 satırdan oluşan şekildeki içi boş üçgeni bastırmıştır.

```
Satir sayisi:10
*
* *
* * *
* * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *

-----
Process exited after 2.33 seconds with return value 0
```

ÖRNEK EKRAN ÇIKTISI

Kod çalıştırıldıktan sonra kullanıcı "10" yazıp enter'a basmıştır. Kod 10 satırdan oluşan şekildeki içi boş üçgeni bastırmıştır.

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.

Soruda boşlukların dışını doldurmayınız. Sadece boşluklara yazdıklarınız değerlendirilmeye alınır.

KOD - Kullanıcının belirttiği sayı miktarınca satırdan oluşan bir paralel kenarı ve paralel kenarın bir köşegenini (başka bir ifadeyle bitişik şekilde içi boş iki üçgen) çizdiren kod

```
#include <stdio.h>
int main()
{
    int i, j, n;
    printf("Satir sayisi:"); scanf("%d", &n);
    for(i = 0; i < n; i++)
    {
        for(j = 0; j <= 3*(n-1); j++)

            if(i == 0 && j > n-1)
                printf("* "); //bir yıldız ve bir boşluk

            else if(i == n-1 && j < 2*(n-1))
                printf("* "); //bir yıldız ve bir boşluk

            else if(i + j == n-1 || i + j == 3*(n-1) || j - i == n-1)
                printf("* "); //bir yıldız ve bir boşluk

            else
                printf(" "); //iki boşluk

        printf("\n");
    }
    return 0;
}
```

Handwritten notes:
 - İlk satır basar (arrow to first row)
 - Son satır basar (arrow to last row)
 - Sol kenar (arrow to left edge)
 - Sağ kenar (arrow to right edge)
 - Köşegen (arrow to diagonal)

EKRAN ÇIKTISI

Satir sayisi:4

```

      *
     *
    *
   *
  *
 *
*

```

ÖRNEK EKRAN ÇIKTISI

Kod çalıştırıldıktan sonra kullanıcı "4" yazıp enter'a basmıştır. Kod 4 satırdan oluşan, çıktıdaki şekli bastırmıştır.

Satir sayisi:5

```

      *
     *
    *
   *
  *
 *
*

```

ÖRNEK EKRAN ÇIKTISI

Kod çalıştırıldıktan sonra kullanıcı "5" yazıp enter'a basmıştır. Kod 5 satırdan oluşan, çıktıdaki şekli bastırmıştır.

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 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 ALABİLMEYİZ 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 ifadelerle 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 dizi[7] = {};

int i;

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

dizi[i] = (i * i) % 5;

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

do

{

dizi[i - 2] += dizi[i - 1];

i--;

}

while(i > 1);

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

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

// printf'lerden sadece birinde \n var.

return 0;

}

dizi

0 1 4 4 1 0 1

i 7 olarak
üstteki for'dan çıkar.

dizi[5]'e [6]'e ekle.
dizi[4]'e [5]'e ekle

dizi

11 11 10 6 2 1 1

dizi[0] dizi[3] dizi[6]

7																			
1	1	6	1																

#include <stdio.h> //sprintf'i de içerir.

void yaz(char [] , int);

int main()

{

char dizgi[100] = "AA";

sprintf(dizgi , "%s" , "12345");

yaz (dizgi , 2);

return 0;

}

void yaz(char e[] , int f)

{

if(f == 0 || f == 6)

return;

printf("%c" , e[f] != '\0' ? e[f] : e[f-1]);

yaz (e , f+1);

return;

}

dizgi

1 2 3 4 5 10

'10' yerine
'5' yazarak

3	4	5	5																
---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

yaz(dizgi, 2) '3' yazdırır,
yaz(dizgi, 3)'ü çağırır, o '4' yazdırır

#include <stdio.h>

typedef struct

{

int gun , ay , yıl ;

} date;

typedef struct

{

date tarih;

int pasta;

} birth;

int main()

{

birth can = { { 9 , 2 , 1999 } , 4 };

int i;

for(i = 0 ; i < can.pasta ; i++)

printf("%d" , can.tarih.ay);

return 0;

}

can-tarih

can-pasta

tanımlı
gynl
singly
pratik
değer
ataması

2	2	2	2																
---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

#include <stdio.h>

int fn (int a);

int main()

{

int e = 23;

while(1 == 1)

{

e += 2;

if(fn(e) == 1)

break;

}

printf("%d" , e);

return 0;

}

int fn (int a)

{

int i;

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

if(a % i == 0)

return 0;

return 1;

}

23'ten sonraki
tek sayılar arasında
asal olanı
bulunca
çıkıyor.

asal sayı ise
1
değilse
0
dönüyor.

2	9																		
---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

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

```
int main()
```

```
{
```

```
    int s = 1, r = 1;
```

```
    do
```

```
    {
```

```
        for( r = 1 ; r <= 8 ; r++ )
```

```
        {
```

```
            if( s + r == 4 )
```

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

```
            else if ( s - r == -4 )
```

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

```
            else
```

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

```
        }
```

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

```
    s++;
```

```
}
```

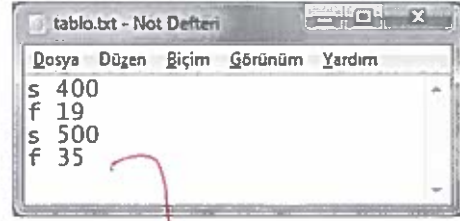
```
while( s <= 3 );
```

```
return 0;
```

Handwritten notes for the first table:

- Red arrow pointing to column 4: $s+r=4$
- Red arrow pointing to column 5: $s-r=-4$

	1	2	3	4	5	6	7	8		
1	x	x	/	x	\	x	x	x		
2	x	/	x	x	x	\	x	x		
3	/	x	x	x	x	x	\	x		



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

```
typedef struct
```

```
{
```

```
    int a;
```

```
    char b;
```

```
} t;
```

```
int main()
```

```
{
```

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

```
    int i, son=3;
```

```
    t r;
```

```
    son = son % 2; → son 1 oldu.
```

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

```
    {
```

```
        fscanf(dosya, " %c", &r.b);
```

```
        // %c'den önce boşluk var.
```

```
        fscanf(dosya, "%d", &r.a);
```

```
    }
```

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

```
        printf( "%c%d\n", r.b, r.a );
```

```
    return 0;
```

```
}
```

Handwritten notes for the second table:

- Red arrow pointing to the first row: $r.b$ → f
- Red arrow pointing to the second row: $r.a$ → 400 and 19
- Red arrow pointing to the first row: $r[i].a$ vs. $r[i].b$ değil!
- Red arrow pointing to the first row: Dosyadan feler, estisim üzerine yazılıyor.

f	1	9								
f	1	9								

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

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

```
int main()
```

```
{
```

```
    char isim[100], name[100];
```

```
    scanf("%s", isim);
```

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

```
    gets(name);
```

```
    printf("%s\n", name+1);
```

```
    return 0;
```

```
}
```

/*Kod çalıştıktan sonra kullanıcı "bal al mia" yazmış 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.*/

b	a	l	a	l		m	i	a
b	a	l						
a	l		m	i	a			

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

```
int main()
```

```
{
```

```
    double t[4] = {1.2, 4.7, 7.3, 9.2};
```

```
    double* ptr;
```

```
    ptr = t;
```

```
    printf("%.21f\n", *ptr);
```

```
    ptr++;
```

```
    printf("%.21f\n", *ptr);
```

```
    double** ptr2;
```

```
    ptr2 = &ptr;
```

```
    (*ptr2)--;
```

```
    printf("%.21f\n", *ptr);
```

```
    return 0;
```

```
}
```

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

```
//kod.c dosyası
```

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

```
enum hayvanlar
```

```
{
```

```
    kedi, kopek, kus, sincap
```

```
};
```

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

```
{
```

```
    printf("%s%d\n", argv[sincap], (argc+1)*kus);
```

```
    return 0;
```

```
}
```

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

Komut Ekranı

gcc	kod.c	-o	Alfabe.exe																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															
-----	-------	----	------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

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

```
int fonk(int [ ], int);
```

```
int main()
```

```
{
```

```
    int d[10] = {1,2,3,4,5,6,7};
```

```
    printf("%d", fonk(d, 3));
```

```
    return 0;
```

```
}
```

```
int fonk(int a[ ], int b)
```

```
{
```

```
    if (b >= 12)
```

```
        return a[ b/2 ];
```

```
    return fonk(a, b+3) + fonk(a, b+6);
```

```
}
```

2	1							

#include <stdio.h>

typedef struct

{

int a;

char b;

} t;

char fonk(t*);

int main()

{

t r;

r.a = 5;

r.b = fonk(&r);

printf("%d%c\n", r.a , r.b);

fonk(&r);

printf("%d%c\n", r.a , r.b);

return 0;

}

char fonk(t* z)

{

(z->a)++;

z->b = 'e';

return 'h';

}

Handwritten notes for main():
 a: 5 → 6 → 7
 b: 'e' → 'h' → 'e'

Handwritten note for fonk():
 'h'

Handwritten note for fonk():
 ilk çağırış

a: 5 → 6

b: 'e'

Handwritten note for fonk():
 2. çağırış

a: 6 → 7

b: 'e'

6	h								
7	e								

//kod.c dosyası

#include <stdio.h>

#include <stdlib.h>

typedef union // union değil!

{

int a,b;

} c;

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

{

c.d = {};

d.a = argc;

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

printf("%d", d.b);

d.b = atoi (argv[1]);

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

printf("%d", d.b);

//bu kodda hiç '\n' yoktur.

return 0;

}

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

Komut Ekranı

gcc kod.c -o kod.exe
kod 9

2	2	9	9						

Handwritten notes for union:
 d.a ile d.b
 RAM'de aynı
 yerde!
 ikisi de aynı tür
 (4 bayt)
 olduğu için
 birinin değeri neyse
 öteki de aynı.

//kod.c dosyası

#include <stdio.h>

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

{

int i;

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

printf("%s\n", argv[i]);

return 0;

}

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

Komut Ekranı

gcc kod.c -o duzenli.exe
duzenli calisaydim

d	u	z	e	n	l	i			
c	a	l	i	s	a	y	d	i	m

#include <stdio.h>

void fonk(int);

int main()

{

int sayac = 5;

fonk(sayac);

return 0;

}

void fonk(int sayac)

{

if(sayac == 0)

return;

printf("%d\n", --sayac);

fonk(sayac-1);

return;

}

H	A	T	A	2					
---	---	---	---	---	--	--	--	--	--

fonk temel olguya hiç uğrayamıyor.

#include <stdio.h>

void fonk(int x);

int main()

{

fonk(4);

return 0;

}

void fonk(int x)

{

if(x != 7)

{

fonk(x+1);

printf("%d", x);

fonk(x+1);

}

return;

}

6	5	6	4	6	5	6			
---	---	---	---	---	---	---	--	--	--

#include <stdio.h>

int fonk(int x);

int main()

{

int a = 7;

int b[3] = { 1, 2, 3 };

fonk(a);

printf("%d", a);

fonk(b[1]);

printf("%d", b[1]);

return 0;

}

int fonk(int x)

{

x = x + 2;

return x;

}

7	2								
---	---	--	--	--	--	--	--	--	--

ya7 dır dı7ı
fonk(7): X
fonk(6): 6
fonk(5): 656
fonk(4): 656 4 656

değer ile gönderim
yine değer ile gönderim
fonk(&b[1])'us değil.
Kod çalıştırıldığında b[1]
yerine 2 yazar ve
fonk(2); yazılmış
gibi çalışır.

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. Dersin öğretim kadrosu (hocalar ve asistanlar) hakkında neler düşünüyorsunuz?

4. Uygulamalı derslerin yeterince faydalı olduğunu düşünüyor musunuz? Nasıl daha geliştirilebilir?

5. Dönemin ortasından sonraki ödevlere katılım nasıl daha arttırılabilir(ödev yüzdelerini arttırmadan)?

6. Geri bildirimde bulunmak istediğiniz diğer konular nelerdir?