

DERS NO:

ÖĞRENCİ İSMİ:

SINAV BİTİRME SAATİ (GÖZETMEN
TARAFINDAN DOLDURULACAKTIR.)

_____ : _____

BİL141 ARA SINAV II - 10.03.2019

UYARI: SORULARI DİKKATLİCE OKUYUNUZ. SINAV ESNASINDA SORU ALINAMAMAKTADIR.

SINAV SONRASINDA BU KÂĞIDI, YUKARIDAKİ BİLGİLERİNİZİ YAZMIŞ VE AŞAĞIDAKİ DÜRÜSTLÜK BEYANINIZI **İMZALAMIŞ OLARAK** GÖZETMENİNİZE TESLİM EDİNİZ.

- Sınav esnasında soruları anlamakta yaşadığınız sorunları, bilgisayarınızda yaşadığınız sorunları vs. lütfen **kendi başınıza** çözünüz. GÖZETMENLERİN GÖREVİ SINAVI GÖZETLEMektir.
- Bilgisayarınız **uzun süreli tıkalı kalırsa** kapatma düğmesine 3-5 saniye basılı tutarak kapatıp sonra yeniden açınız. Böyle bir duruma karşı önlem olarak kodunuzu **SIK SIK KAYDEDİNİZ.**

SINAVDA UYULMASI GEREKEN TEMEL KURALLAR:

- Sınav esnasında yanınızda getirdiğiniz el yazınızla yazılmış ipucu kâğıdının ve DevCpp'in kod parçaları özelliğinin haricinde herhangi bir materyalden/kaynaktan/kişiden **yardım alamazsınız.**
- Sınav süresince(kod gönderim kısmı hariç) DevCpp haricinde herhangi bir program kullanamazsınız. DevCpp'da da **SADECE KOD YAZDIĞINIZ SEKMELER(iki soru olduğu için sadece iki sekme)** açık olmalıdır. Herhangi başka bir kod, yardım kutucuğu, ipucu kutucuğu, slayt vs. açamazsınız.
- Sınav süresine bilgisayarının internet erişimini açmak, son sayfada adresi verilen siteye girmek, bu sitedeki forma kodlarını aktarmak ve formu başarılı şekilde göndermek **DÂHİLDİR**. Sınavın süresi bittikten sonra kod gönderimi yapılmamalıdır.
- 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.**

Sınav esnasında herhangi bir kopya girişiminde bulunmadığımı/bulunmayacağımı, yukarıdaki kuralları ihlal etmediğimi/etmeyeceğimi ve böyle bir girişime/kural ihlaline şahit olduysam gözetmenime bildirdiğimi/bildireceğimi beyan ederim.

Lütfen imzalayınız:

KODLAMAYA DAİR UYARILAR (LÜTFEN DİKKATLE OKUYUNUZ):

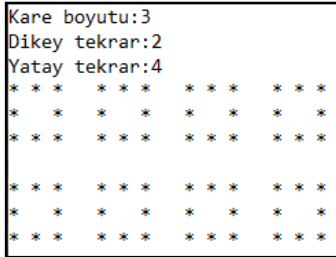
- Çözümlerinizin sadece istendiği şekilde çalışıyor olması yeterli OLMAYABİLİR! **KÖTÜ PROGRAMCI ALIŞKANLIKLARI** (goto ve küresel değişken kullanımı gibi) ve **GEREKSİZ YERE UZUN YAPILAR** (peş peşe bir sürü if else ile upuzun bir kod yazmak gibi) kullanmanız durumunda kodunuz tam olarak çalışsa bile **çok düşük/sıfır puan** alabilirsiniz.
- Kodunuzun **hata vermeden** istenen şekilde çalışıyor olması önceliğiniz olsun. Ardından **vakit kalırsa**, detaylandırılmış uyarı ayarları açıkken **uyarı vermiyor olmasına** dikkat ediniz. Öte yandan, eğer kodunuz doğru çalışmıyorsa bunun nedeni uyarı olarak aşağıda belirtili olabilir; bu açıdan **uyarılarla önem veriniz.**
- Sorunun istediklerine **ek özellikler eklemek** (süre, renk, daktilo kodu vs. gibi) size ekstra puanlar getirmez.

SORU1 – KARE ÇİZDİRME SORUSU (50 PUAN)

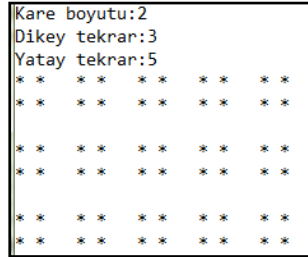
Bu soru parçalardan oluşmaktadır. Bazı parçaları yapamasanız bile diğer parçalardan puan almanız mümkündür. Yaptığınız parçalardan, kodunuzun doğruluk miktarına göre tam ya da kısmi puanlar alırsınız. **KOD GÖNDERİM FORMUNDA BU SORUYA DAİR SADECE 1 KOD GÖNDERECEKSİNİZ.** Bu soruda sizden örnek çıktı 1.1 ve 1.2'de gözüktüğü şekilde,

- kullanıcıdan sırayla **kare boyutu**, **dikey tekrar**, **yatay tekrar** değerleri girişini alan
- yatayda ve dikeyde **kare boyutu** adedince yıldızı(*) içerecek şekilde içi boş bir kare oluşturan
- bu kareleri **dikey tekrar** ve **yatay tekrar** adetlerince dikeyde ve yatayda tekrarlayan bir kod yazmanız istenmektedir.

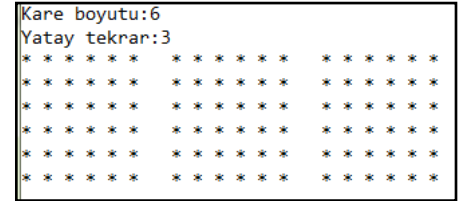
UYARI: Dikey ve yatay yıldızlar arasındaki uzaklığın kabaca birbirine denk olabilmesi için yatayda bir kareye ait yıldızlar arasında bir boşluk karakteri genişliğinde aralık olmasına (örn. *** yerine * * * şeklinde) dikkat ediniz.



Örnek çıktı-1.1 Kullanıcı sırasıyla 3, 2 ve 4 girişleri yapıp her girişten sonra enter'a basmıştır. Kod yatayda ve dikeyde istenen adette ve boyutta içi boş kareler bastırabildiği için aşağıdaki tüm parçalardan puan alabilir.



Örnek çıktı-1.2 Kullanıcı sırasıyla 2, 3 ve 5 girişleri yapıp her girişten sonra enter'a basmıştır. Aşağıda Parça2'de açıklanan nedenden dolayı boyut değeri 2 olduğu için bu çıktıda içi boş kare oluşturma beklenmez. Başka girdilere vereceği çıktıları bağlı olarak böyle bir kod aşağıdaki tüm parçalardan puan alabilir.



Örnek çıktı-1.3 Kullanıcı sırasıyla 6 ve 3 girişleri yapıp her girişten sonra enter'a basmıştır. Kod içi boş kare oluşturmadığı ve dikeyde tekrarı sağlamadığı için aşağıdaki Parça2 ve Parça3'ten puan alamaz; bu parçalar haricindeki parçalardan puan alabilir.

Çözümünüz parçalı puanlama ile puanlandırılacaktır. Bu nedenle yapamadığınız bir parçayı atlayıp diğer parçalara odaklanabilirsiniz. Parçalar ve puanları şu şekildedir:

1. **EN AZ 1 TANE KARE(İÇİ DOLU YA DA BOŞ) OLUŞTURABİLMEK (5 PUAN):** Kullanıcının girdiği kare boyutu n ise, $n \times n$ boyutlarında bir kare oluşturabilmek. Yataydaki yıldızlar arasında boşluk karakteri olması ile ilgili yukarıdaki uyarıya dikkat ediniz. Bu uyarıyı dikkate almayan kodlar bu parçadan tam puan alamaz.
2. **OLUŞTURULAN KARELERİN İÇİNİN BOŞ OLMASI(5 PUAN):** Bu kısmı yapan, Parça1'den de ayrıca puan alır. Sorunun devamı için bu parçanın yapılması zorunlu değildir. İçi dolu ve tekrarlayan kareler oluşturursanız bu parça haricinde diğer parçalardan puan alabilirsiniz. Ekranada oluşan tüm karelerin istenen boyutta ve içinin boş olması. İçi boş kare oluşturabilmek için minimum boyutun 3 olması gerektiğine dikkat ediniz. Bu nedenle boyut değeri 1 ya da 2 olduğunda içi boş kare oluşturulması beklenmemektedir.
3. **DİKEYDE TEKRARLATABİLMEK(15 PUAN):** Örnek çıktılarda gözüktüğü üzere, oluşturulan şekillerin kullanıcının girdiği **dikey tekrar** adedince dikeyde tekrarlaması. Tekrarlar arasında bir satır boşluk olduğuna dikkat ediniz.
4. **YATAYDA TEKRARLATABİLMEK(25 PUAN):** Örnek çıktılarda gözüktüğü üzere, oluşturulan şekillerin kullanıcının girdiği **yatay tekrar** adedince yatayda tekrarlaması. Tekrarlar arasında iki ayrı kare arasında(bir kareye ait olan yıldızlar arasındaki 1 boşluk ile karıştırmayınız) iki boşluk karakteri genişliğinde aralık olduğuna dikkat ediniz.
5. **BONUS(5 PUAN):** Yukarıdaki parçalardan biri hariç hepsini yapmış öğrenciler için geçerlidir. **Fonksiyonları** etkin şekilde kullanmak ve böylece kodu daha rahat geliştirilebilir hale getirmek (örn. sadece void alıp void dönen bir fonksiyon kullanmak fonksiyonların etkin kullanımı sayılmaz.)

Kullanıcının GEÇERLİ GİRİŞLER YAPACAĞINI(örn. her zaman pozitif sayılar gireceğini) varsayınız.

ÖNEMLİ UYARI:

- Sınav sonunda süreniz kalırsa, 1) detaylandırılmış uyarı açık şekilde kodunuzu inceleyiniz ve kodunuzun uyarı üretmiyor olmasına dikkat ediniz 2) kodunuzu değişik değerlerle iyice test ederek doğru çalıştığına emin olunuz.
- Sadece, doğru çıktı veren bir kod yazarak tam puan alamayabilirsiniz. Aynı zamanda sizden istenen, **OLABİLDİĞİNCE SADE** bir kod yazabilmenizdir. (bkz. 1 nolu sayfanın sonundaki **KODLAMAYA DAİR UYARILAR** kısmı)

SORU2 – GENOMİK ANALİZ SORUSU (50 PUAN)

Bu soru parçalardan oluşmaktadır. Bazı parçaları yapamasanız bile diğer parçalardan puan almanız mümkündür. Yaptığınız parçalardan, kodunuzun doğruluk miktarına göre tam ya da kısmi puanlar alırsınız. **KOD GÖNDERİM FORMUNDA BU SORUYA DAİR SADECE 1 KOD GÖNDERECEKSİNİZ.**



İnsan vücudunda 37 trilyon civarında hücre vardır. Bu hücreler DNA içerisinde yüklü bilgi ile idare edilirler. **Genomik analiz** DNA dizilimi, yapısal varyasyon, gen ifadesi gibi genomik özelliklerin tespiti, ölçümü ya da kıyaslamasını kapsar. Ortaya çıkan yoğun bilginin bilgisayar veritabanlarında saklanması, işlenmesi ve anlamlandırılması gereği bilgisayar programlamayı da içeren biyoinformatik disiplinine ihtiyaç duyulur.

Bu soruda sizden kullanıcıdan bir DNA zincirindeki genetik kod parçasını almanız ve incelemeniz istenmektedir. Genetik kod **A**(Adenin), **T**(Timin), **G**(Guanin) ve **C**(Sitozin) nükleotitlerinden oluşmaktadır.

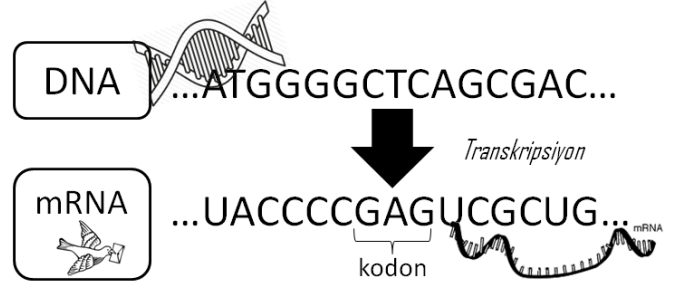
Genetik kod içerisinde protein sentezine ait bilgiler de mevcuttur. DNA'dan aldığı bilgi ile ribozoma gidip protein sentezini başlatacak materyale mRNA(mesajcı RNA) denir. DNA'dan bilgi alma sürecine ise transkripsiyon denir. Transkripsiyon, genetik koddaki

- A'lara karşılık mRNA'da U (T değil)
- T'lere karşılık mRNA'da A
- G'lere karşılık mRNA'da C
- C'lere karşılık mRNA'da G

gelmesi sürecidir.

Sizden istenen adımlar şu şekildedir:

1. Kullanıcıdan genetik kodun alınması. Girilecek DNA kodunun minimum 1, maksimum 999 nükleotit içereceğini (normalde bir çekirdek genom yaklaşık 3.2 milyar nükleotit içerir) ve bunların aralarında boşluk olmadan girileceğini varsayınız.
2. Her bir nükleotit türünün alınan genetik kodda kaç kez geçtiğinin sayılması
3. Genetik koddan yola çıkarak transkripsiyon ile oluşacak mRNA'nın içereceği kodunun (kodon dizilimi) bulunması
4. Son olarak sizden genetik kodla **aynı sayıda nükleotide sahip, rastgele oluşturulmuş, birbirinden farklı** iki genetik kod oluşturmanız istenmektedir. A, T, G ve C nükleotitlerinden oluşacak olan bu genetik kodlara ait detaylar sorunun devamındaki parça başlıkları altında belirtilmiştir.



```
Genetik kod:ATAGGACC
A:3 T:1 G:2 C:2
mRNA:UAUCCUGG
Alternatif kod1:AATCTATG
Alternatif kod2:TGACAGCA
```

Örnek çıktı-2.1 Kullanıcı "ATAGGACC" girişi yapmış ve enter'a basmıştır. Bu ifadede "A" harfi 3 kez geçtiği için bu durum bir sonraki satırda "A:3" şekliyle belirtilmiştir. Aynı durum diğer harfler için de yapılmıştır. Daha sonra mRNA'da A yerine U, T yerine A, G yerine C ve C yerine G dönüşümünün yapıldığı görülmektedir. Ardından yine 8 harfli (aynı uzunlukta) olacak şekilde iki tane kod yazdırılmıştır. Bu kodların birbirinden farklı olması koşulu sağlanmıştır. Böyle bir kod, (giriş kontrolü özelliğine de sahipse) tüm parçalardan puan alabilir.

```
Genetik kod:atgc
Gecersiz giriş!
Yalnız A,T,G ve C içermeli.
Tekrar giriniz:AUGC
Gecersiz giriş!
Yalnız A,T,G ve C içermeli.
Tekrar giriniz:ATGC
A:1 T:1 G:1 C:1
mRNA:UACG
Alternatif kod1:TAAC
Alternatif kod2:CAAA
```

Örnek çıktı-2.2 Kullanıcı sırasıyla "atgc", "AUGC" ve "ATGC" girişleri yapmış ve her girişten sonra enter'a basmıştır. İlk girişi büyük harflerle olmadığı için, ikinci girişi ise geçersiz olan U harfini içerdiği için geçerli kabul edilmemiştir. Üçüncü girişi geçerli olduğu için analiz sonucu paylaşılmıştır. Böyle bir kod, sorudaki değer kontrolü parçasından puan alabilir.

```
Genetik kod:AT
A:1 T:1 G:0 C:0
mRNA:UA
Alternatif kod1:TC
Alternatif kod2:CG
```

Örnek çıktı-2.3

```
Genetik kod:AT
A:1 T:1 G:0 C:0
mRNA:UA
Alternatif kod1:TA
Alternatif kod2:CG
```

Örnek çıktı-2.4

Yukarıdaki iki örnek çıktı da aynı koda aittir. İki kod çıktısında da kullanıcı "AT" girişi yapıp enter'a basmıştır. Karşılığında üretilen mRNA değerleri (girişler aynı olduğu için) iki kodda da aynıdır. Ancak, alternatif kodlar rastgele üretildiği iki çıktıda birbirinden farklı çıkmıştır. Aynı zamanda her iki çıktıda da kod1 ve kod2 değerleri birbirinde farklıdır. Böyle bir kod, sorudaki alternatif kod üretimi parçalarından puan alabilir.

ÖNEMLİ UYARI:

- Sınav sonunda süreniz kalırsa, 1) detaylandırılmış uyarı açık şekilde kodunuzu inceleyiniz ve kodunuzun uyarı üretmiyor olmasına dikkat ediniz 2) kodunuzu değişik değerlerle iyice test ederek doğru çalıştığına emin olunuz.
- Sadece, doğru çıktı veren bir kod yazarak tam puan alamayabilirsiniz. Aynı zamanda sizden istenen, **OLABİLDİĞİNCE SADE** bir kod yazabilmenizdir. (bkz. 1 nolu sayfanın sonundaki **KODLAMAYA DAİR UYARILAR** kısmı)

Çözümünüz parçalı puanlama ile puanlandırılacaktır. Bu nedenle yapamadığınız bir parçayı atlayıp diğer parçalara odaklanabilirsiniz. Parçalar ve puanları şu şekildedir:

1. **KULLANICIDAN GENETİK KODU GEÇERLİ ŞEKİLDE ALABİLMEK(10 PUAN):** Kullanıcının bitişik şekilde (aralarında boşluk olmadan) girdiği genetik kodda "A", "T", "G" ve "C" harfleri dışında bir harf varsa Örnek Çıktı 2.2'deki gibi uyarı verdirebilmek ve tekrar giriş isteyebilmek. Bu durum kullanıcı geçerli giriş yapana kadar devam eder. Giriş kontrolü yapamayan öğrenciler bu parçadan puan alamaz ancak kontrol yapmadan (kullanıcının her zaman geçerli giriş yapacağını varsayarak) diğer parçalardan puan almaya çalışabilir.
2. **NÜKLEOTİT SAYIMLARINI YAPABİLMEK(10 PUAN):** 4 çeşit nükleotit türünün("A", "T", "G" ve "C") kullanıcının girdiği genetik kodda kaçar kez geçtiklerini örnek çıktılarda olduğu gibi doğru şekilde ekrana yazdırabilmek
3. **mRNA BİLGİSİNİ OLUŞTURABİLMEK(10 PUAN):** Örnek çıktılardaki gibi genetik kod girişinden yola çıkıp, A yerine U, T yerine A, G yerine C ve C yerine G dönüşümünün ardından mRNA bilgisini oluşturabilmek
4. **ALTERNATİF KOD1 BİLGİSİNİ OLUŞTURABİLMEK (10 PUAN):** Girilen genetik kodla aynı uzunlukta olan; "A", "T", "G" ve "C" nükleotitlerinden oluşan bir kodu rastgele olarak oluşturabilmek. *Alternatif kod1*, kod her çalıştığında aynı şekilde çıkıyor olmalıdır. Tesadüfen aynı şekilde çıkması haricinde, kod her çalıştığında ayrı bir *alternatif kod1* çıkmalıdır. Çıkacak kodun tesadüfen girilen genetik kodla aynı olabilmesinde sakınca yoktur. "*Alternatif kod1*, tüm nükleotitlerden en az bir tane içermelidir" gibi bir koşul yoktur; örneğin, tesadüfen tamamı tek nükleotit çeşidiyle oluşmuş (AAAA... gibi) bile olabilir.
5. **ALTERNATİF KOD2 BİLGİSİNİ OLUŞTURABİLMEK (10 PUAN):** Girilen genetik kodla aynı uzunlukta olan; "A", "T", "G" ve "C" nükleotitlerinden oluşan bir kodu rastgele olarak oluşturabilmek. *Alternatif kod2*, *alternatif kod1*'den farklı olmalıdır. Tesadüfen bile olsa alternatif kod1 ile aynı çıkıyor olmalıdır. Çıkacak kodun tesadüfen girilen genetik kodla aynı olabilmesinde sakınca yoktur. "*Alternatif kod2*, tüm nükleotitlerden en az bir tane içermelidir" gibi bir koşul yoktur; örneğin, tesadüfen tamamı tek nükleotit çeşidiyle oluşmuş (AAAA... gibi) bile olabilir.
6. **GELİŞTİRİLEBİLİR BİR KOD YAZMIŞ OLMAK(5 PUAN):** Yukarıdaki parçalardan biri hariç hepsinden puan alabilmiş öğrenciler için geçerlidir. Kod **fonksiyonları, dizileri ve yapıları** etkin şekilde kullanılarak kodlanmış, rahatlıkla geliştirilebilirlik özelliğine sahip olmalı. (örn. içinde sadece bir değişken tanımlı bulunan yapı, yapıların etkin kullanımı sayılmaz.)



LÜTFEN KOD GÖNDERİM AŞAMASINA **SINAVIN SON 5 DAKİKASINDAN ÖNCE** GEÇİNİZ. BU AŞAMAYA GEÇTİĞİNİZDE EN YAKININIZDAKİ GÖZETMENE HABER VERİNİZ (*Böylece DevCpp dışında bir program açtığınız için sizi uyarmaya gelmezler.*)

Kodunuzu göndereceğiniz form:

BU KÂĞIDI ÖN KISMINDAKİ BİLGİLERİ DOLDURDUKTAN VE IMZALADIKTAN SONRA **GÖZETMENİNİZE TESLİM ETMEYİ UNUTMAYINIZ.**

OLASI ÇÖZÜM - SORU1 - STANDART ÇÖZÜM

```
#include <stdio.h>
#include <conio.h>
void kare (int,int,int);
void degerAl(char metin[], int*ptr);
int main()
{
// File-> Properties -> Actual code Lines: 41
    int n, dikey, yatay ;
    degerAl("Kare boyutu:", &n);
    degerAl("Dikey tekrar:", &dikey);
    degerAl("Yatay tekrar:", &yatay);
    kare(n, dikey, yatay); //isteyen de imlec(x,y) ile yapar.
    /*
        Keşke ödev3'ü yapsaydım, şekiller çizdirseydim...
        Keşke Uygulamalı Dersteki şekil çizmeli codingame'i
        eve gidince oturup yapsaydım, kare şeklinde rastgele
        harf bastırma sorusunu yapsaydım.
        Keşke derste işlenen ikizkenar üçgen çizdirmeyi aynı
        gün evde kendim deneseydim...
        bir while döngüsü içindeyim, sürekli Bil141 alıyorum,
        if(duzenli calismak==1) break; biliyorum ama...
        ama...larım ve ben donup duruyorum...
        bir while(1) içinde kaybolmaktan korkuyorum.
    NOT: AA ALMAK İSTEMEYEN VE BUNUN GEREĞİNİ YAPAN ÖĞRENCİLERİN
    SEÇİMLERİNE SAYGI DUYUYORUZ. BU YORUM METNİ İLERLEYEN DÖNEMLERDE
    BURAYI İNCELEYECEK ÖĞRENCİLERİ UYARMAK AMAÇLIDIR.
    */
    return 0;
}

void kare (int n, int dikey, int yatay)
{
    int i, j, y, d;
    /*
        Bir for satırı basıyorsa ve sonra, bir sonraki satıra geçiliyorsa...
        0 for'u, "yatay" kez tekrar ettirirsem, satırların yatay kopyaları oluşur.
        Tüm işlemi "dikey" kez tekrar ettirirsem de dikey kopya oluşur.
    */
    for(d=0 ; d<dikey; d++) //dikeyde tekrarlama 15 puan
    {
        for(i=0 ; i<n; i++)
        {
            for(y=1; y<=yatay; y++) //yatayda tekrarlama 25 puan
            {
                for(j=0; j<n; j++)
                {
                    if(i==0 || i==n-1 || j==0 || j==n-1)
                        printf("* ");
                    else
                        printf(" ");
                }
                printf(" ");
            }
            printf("\n");
        }
        printf("\n");
    }
}

void degerAl(char metin[], int*ptr)
{
    printf("%s", metin);
    scanf("%d", ptr);
    // TODO: Buraya negatif girise karsi bir deger kontrolu eklenebilir.
}
```

OLASI ÇÖZÜM - SORU1 - imlec_xy 'Lİ ÇÖZÜM

```
#include <conio.h>
#include <stdio.h>
#include <windows.h>
void imlec_xy(int x, int y);
void kare (int x, int y, int n);
int main()
{
    int n, dikey, yatay ;
    printf("Kare boyutu:");
    scanf("%d", &n);
    printf("Dikey tekrar:");
    scanf("%d", &dikey);
    printf("Yatay tekrar:");
    scanf("%d", &yatay);
    int i,j;
    for(i=0; i<yatay; i++ )
        for(j=0; j<dikey; j++ )
            kare(i*(n+1),j*(n+1),n); //her kare n+1 yer kaplar

    return 0;
}

void kare (int x, int y, int n) //x ve y karenin sol ust noktasi koordinatlari
{
    int i,j;
    for(i=0; i<n; i++) // imlecin satirini oynatir
        for(j=0; j<n; j++) // imlecin sutununu oynatir
            if(i==0 || i==n-1 || j==0 || j==n-1)
            {
                imlec_xy(x+i,y+j);
                printf("*");
            }
}

void imlec_xy(int x, int y)
{
    COORD coord;
    coord.X = 2*x;//karakterler arasi bosluk
    coord.Y = y+3; // Veri aldigimiz 3 satiri atla.
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);
}
```

OLASI ÇÖZÜM - SORU2 - SIRADAN ÇÖZÜM

```
// SIRADAN BİR ÇÖZÜM.
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <string.h>
int nukleotitdenSiranoya(char x); // A Sirano:0, T Sirano:1 ...
int gecerliMi(char dna[]);
void uret(char kod[], int l);
int main()
{
// File-> Properties -> Actual code Lines: 84
    srand(time(NULL));
// SÖZDE KOD 1) KULLANICIDAN DEĞER AL
    printf("Genetik kod:");
    char dna[1000];
    gets(dna);
    while(!gecerliMi(dna))
    {
        printf("Gecersiz giris!\nYalnız A,T,G ve C icermeli.\n");
        printf("Tekrar giriniz:");
        gets(dna);
    }

// SÖZDE KOD 2) TEK TEK İNCELE, SAYIM YAP VE YAZDIR
    int l= strlen(dna);
    int i;
    int sirano[4] = {};
    for(i=0; i<l; i++)
        sirano[nukleotitdenSiranoya(dna[i])]++;

    char nukleotit[4] = {'A', 'T', 'G', 'C'};
    for(i=0; i<4; i++)
        printf("%c:%d ", nukleotit[i], sirano[i]);

// SÖZDE KOD 3) TEK TEK İNCELE VE KODON HALİNİ YAZDIR
    printf("\nmRNA:");
    for(i=0; i<l; i++)
    {
        if(dna[i]=='A') //switch daha iyi gidermis.
            printf("U");
        else if(dna[i]=='T')
            printf("A");
        else if(dna[i]=='G')
            printf("C");
        else if(dna[i]=='C')
            printf("G");
    }

// SÖZDE KOD 4) ALTERNATİF KOD1 ÜRET VE YAZDIR
    char dna1[l+1]; // +1'in nedeni printf %s ile yazdirabilmek
    dna1[l]='\0'; // printf %s ile yazdirabilmek için
    uret(dna1, l);
    uret(dna1, l);
    dna1[l]='\0'; // printf %s ile yazdirabilmek için
    printf("\nAlternatif kod1:%s", dna1);
// SÖZDE KOD 5) ALTERNATİF KOD2 ÜRET, GEREKIYORSA TEKRAR ÜRET VE YAZDIR
    char dna2[l+1]; // +1'in nedeni printf %s ile yazdirabilmek
    dna2[l]='\0'; // printf %s ile yazdirabilmek için
    do
    {
        uret(dna2, l);
    }
    while(!strcmp(dna1,dna2));
    printf("\nAlternatif kod2:%s", dna2);
/*
    Keşke rastgele harf değeri üreterek ekrana bastırduğumuz
    o uygulamalı ders sorusunu yapsaydım, keşke dizgiden harf attığımız
    o soruyu, dersteki Red Kit sorusunu, uygulamalı dersteki anagram
*/
}
```

sorusunu yapsaydım... Keşkelerim bir dizginin içerisine dolmuş,
bir '\0' arıyorum ki durayım, ama yok... keşke dönem başından beri
düzenli şekilde çalışsaydım, keşke ben de bu dersi eğlenerek ve
başararak sürdürseydim, keşke "sev-çalış-başar" döngüsünden hiç
çıkmasaydım... dizgi bitti, RAM'de umarsızca dolanıyorum, yorgunum...

NOT: AA ALMAK İSTEMEYEN VE BUNUN GEREĞİNİ YAPAN ÖĞRENCİLERİN
SEÇİMLERİNE SAYGI DUYUYORUZ. BU YORUM METNİ İLERLEYEN DÖNEMLERDE
BURAYI İNCELEYECEK ÖĞRENCİLERİ UYARMAK AMAÇLIDIR.

*/

return 0;

}

void uret(char kod[], int l)

{

char nukleotit[4] = {'A', 'T', 'G', 'C'};

int i;

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

{

kod[i] = nukleotit[rand()%4] ;

}

}

int nukleotitdenSiranoya(char x)

{

// sıra no A:0 T:1 G:2 C:3

char nukleotit[4] = {'A', 'T', 'G', 'C'};

int i;

for(i=0; i<4; i++)

if(nukleotit[i]==x)

return i;

return -1; *//yoksa*

}

int gecerliMi(char dna[])

{

int i;

int l = strlen(dna);

/*

*Girilecek DNA kodunun minimum 1, maksimum 999 nükleotit içeriğini varsayınız.
dendigi için asagidaki kisma gerek yok aslında.*

*/

if(l==0)*//kullanici sadece entera bastıysa*

return 0; *//bu kısım olmasa da olur*

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

{

if(nukleotitdenSiranoya(dna[i])==-1)

return 0;

}

return 1;

}

OLASI ÇÖZÜM - SORU2 - GELİŞTİRİLEBİLİR ÇÖZÜM

```

#include <stdio.h>
#include <time.h>
#include <stdlib.h>
#include <string.h>

typedef struct
{
    char kod[1000];
    int l; //uzunluk
    int A,G,T,C; //nukleotit adetleri
} gen_turu;

int gecerliMi(char []); //dna'nın içinde ATGC harici varsa 0, uygunsa 1.
void uret(gen_turu* ); //rastgele dizilim üretir
int nukleotitdenSiranoya(char x); //A alırsa 1, ... C alırsa 4 doner, diger turlu -1

int main()
{
    srand(time(NULL));

    // SÖZDE KOD 1) KULLANICIDAN DEGER AL
    printf("Genetik kod:");
    gen_turu dna = {"",0,0,0,0,0}; //sayac olarak kullanacagimiz icin sifirlama sart A,T,G,C'e.
    gets(dna.kod);
    while(!gecerliMi(dna.kod)) //DEĞER KONTROLÜ
    {
        printf("Gecersiz giris!\nYalniz A,T,G ve C icermeli.\n");
        printf("Tekrar giriniz:");
        gets(dna.kod);
    }
    /*
    yukarıdaki kod DEGER ALMA ismi icin ek bilgi:
    nesnesel programlama dna.gecerliMi() şeklinde dna içinden fonksiyon(metot diye geçiyor) çağırılabilir.
    dahası, dna.girisAl(); şeklinde yapılır, o esnada girisAl, gecerliMi'yi kullanabilir.
    */
    int i;
    // SÖZDE KOD 2) TEK TEK İNCELE VE İNCELERKEN a)SAYIM YAP b) MRNA OLUŞTUR.
    gen_turu mRNA; //içindeki A,T,G,C değerlerini kullanmayacağız gerçi ama sorun değil.
    dna.l = strlen(dna.kod);
    //sürekli strlen'i çağırıp kodu yavaşlatmamak için bir yere kaydettik
    //ek bilgi: nesnesel programlamada bu kısım da otomatize edilebiliyor

    for(i=0; i<dna.l; i++)
    {
        switch(dna.kod[i])
        {
            case 'A': //Kodda A varsa
                dna.A++; //A sayimini guncelle.
                mRNA.kod[i]='U'; //mRNA'a U yerlestir.
                break;
            case 'T':
                dna.T++;
                mRNA.kod[i]='A';
                break;
            case 'G':
                dna.G++;
                mRNA.kod[i]='C';
                break;
            case 'C':
                dna.C++;
                mRNA.kod[i]='G';
                break;
        }
    }
    mRNA.kod[i]='\0'; // printf %s ile yazdirabilmek icin
    // SÖZDE KOD 2) c) RASTGELE İKİ KOD ÜRET.
    gen_turu dna1, dna2;
    dna1.l = dna2.l = dna.l; //dna.l dna1 ve dna2'nin l'sine kaydedilir.

```

```

/*
= az onceliklidir.
Kod sagdan sola dogru ilerleyecegi icin sunu yazmaya denktir.
dna2.l = dna.l
dna1.l = dna2.l;
*/

uret(&dna1);
do //dna1'e aynı ise tekrar üret
{
    uret(&dna2);
}
while(!strcmp(dna1.kod, dna2.kod));

// SÖZDE KOD 3) BULDUKLARINI YAZDIR.
printf("A:%d T:%d G:%d C:%d\n", dna.A, dna.T, dna.G, dna.C);
printf("mRNA:%s\n", mRNA.kod);
printf("Alternatif kod1:%s\n", dna1.kod);
printf("Alternatif kod2:%s\n", dna2.kod);
return 0;
}

int gecerliMi(char dna[])
{
    int i;
    int l = strlen(dna);
    /*
    Girilecek DNA kodunun minimum 1, maksimum 999 nükleotit içeriğini varsayınız.
    dendigi için asagidaki kısma gerek yok aslında.
    */
    if(l==0)//kullanici sadece entera bastıysa
        return 0; //bu kısım olmasa da olur
    for(i=0; i<l; i++)
    {
        if(nukleotitdenSiranoya(dna[i])==-1)
            return 0;
    }
    return 1;
}

void uret(gen_turu *ptr )
{
    static char harf[4] = {'A', 'T', 'G', 'C'};
    //static şart değil; (bkz. işaretçiler sunusu ek slaytlar)
    //static fonksiyona her giriş, çıkışta yeniden harf dizisi oluşturmasın diye
    int i;
    for(i=0; i<ptr->l; i++)
    {
        ptr->kod[i] = harf[rand()%4];
    }
    ptr->kod[i]='\0';// printf %s ile yazdirabilmek için
}

int nukleotitdenSiranoya(char x)
{
    // sıra no  A:0  T:1  G:2  C:3
    static char nukleotit[4] = {'A', 'T', 'G', 'C'};
    //static şart değil; (bkz. işaretçiler sunusu ek slaytlar)
    //static fonksiyona her giriş, çıkışta yeniden harf dizisi oluşturmasın diye
    int i;
    for(i=0; i<4; i++)
        if(nukleotit[i]==x)
            return i;
    return -1; //yoksa
}

```