BİL141 ARA SINAV II - 11.11.2018

UYARI: SORULARI DİKKATLİCE OKUYUNUZ. SINAV ESNASINDA SORU ALINAMAMAKTADIR.

KODLAMAYA DAİR UYARILAR (LÜTFEN DİKKATLE OKUYUNUZ):

1. 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.
2. Kodunuz çalıştırıldığında hata vermiyor ama aşağıdaki mesaj penceresinde uyarı(warning, kırmızı renkli ifadeler) veriyorsa bunun bir nedeni uygun kütüphaneleri eklememiş olmanız olabilir.
3. Kod tamamlandığında "return value"nun 0 olmasına özen gösteriniz.
4. Sorunun istediklerine ek özellikler eklemek (süre, renk, daktilo kodu vs. gibi) size ekstra puanlar getirmez.

SORU1 – VERİ ANALİZ TABLOSU 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 bir **uzunluk** değeri girişi alan,
- girilen uzunluk değerinde genişlik ve yine aynı değerde yüksekliğe sahip olan, **RASTGELE RAKAMLARLA** ***([0-9] aralığında sayılar)*** üretilmiş bir **üçgen motifli tabloyu** ekrana yazdıran (kod her çalıştırıldığında aynı sayıyı üretmiyor olmalı),
- bu tablonun satır ve sütunlarındaki rakamların toplamlarını (Örnek çıktı-1.1 ve 1.2'de gözüktüğü şekilde) gösteren bir kod yazmanız istenmektedir.

```
Uzunluk:3
VERI ANALIZ TABLOSU:
3       7       0       |  10
2       7               |  9
3                       |  3

-       -       -

8       1       0
        4
--------------------------------
Process exited after 0.6729 seconds with
 return value 0
```

```
Uzunluk:3
VERI ANALIZ TABLOSU:
4       7       1       |  12
0       9               |  9
4                       |  4

-       -       -

8       1       1
        6
--------------------------------
Process exited after 1.342 seconds with
return value 0
```

Örnek çıktı-1.1 ve 1.2 Soldaki örnek çıktıda kullanıcı *"3"* girişi yapıp enter'a basmıştır. Neticesinde, 3 satır ve 3 sütundan oluşan bir tablo üretilmiştir. Bu tablodaki sayı adedi her satırda bir önceki satıra göre bir eksilmektedir. Tüm sayılar [0-9] aralığında olacak şekilde rastgele üretilmiştir. İlk satırdaki sayıların toplamı 3+7+0=10 olduğu için satır sonunda 10 değeri yer almıştır. İlk sütundaki sayıların toplamı 3+2+3=8 olduğu için sütunun altında 8 değeri yer almıştır. İkinci sütundaki sayıların toplamı, iki basamaklı bir sayı olan 14'tür. 14 dikeyde 1 ve 4 rakamları alt alta getirilerek gösterilmiştir. Kod, tekrar çalıştırıldığında(sağdaki örnek çıktı), aynı uzunluk değeri girilmiş olmasına rağmen, rastgelelik nedeniyle başka sayılar üretilmiştir. Bu şekilde çalışan bir kod aşağıda gösterilen tüm parçalardan puan alabilir. Eğer kod, aşağıdaki ÖNEMLİ UYARI olarak belirtilen kısmı da uygulamışsa sorunun puanının tamamını alabilir.

```
Uzunluk:2
VERI ANALIZ TABLOSU:
3       5       |  8
7       4       |  11


--------------------------
Process exited after 0.5419
Devam etmek için bir tuşa b
```

Örnek çıktı-1.3 Kullanıcı "2" girişi yapıp enter'a basmıştır. Üçgen tablo yerine kare tablo oluşturulmuş ve sütun rakam toplamları yazılmamıştır. Bu şekilde çalışan bir kod eksiklerine rağmen parçalı puanlama sayesinde aşağıda gösterilen 1, 3, 4 nolu parçalardan puan alabilir.

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

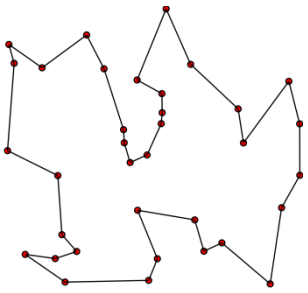
1. ***BİR KARE TABLO OLUŞTURABİLMEK (5 PUAN):*** *Parça2'yi yapan öğrencilerin bu parçayı yapmalarına gerek yoktur, bu parçadan da tam puan alabilirler. Parça2'yi yapamayan öğrenciler bu parçayı yapıp diğer parçaları yapmaya çalışabilir.* Kullanıcının girdiği uzunluk değeri n ise, nxn boyutlarında bir kare tablo oluşturabilmek (tabloda rakam yerine * kullanan öğrenci buradan puan alabilir). Tablodaki değerlerin arasında "\t" (tab) konabilir.
2. ***ÜÇGEN ŞEKİLLİ BİR TABLO OLUŞTURABİLMEK (10 PUAN):*** Kullanıcının girdiği uzunluk değeri n ise, ilk satırda n tane rakam olmalı, ondan sonraki her yeni satırda rakam adedi 1 azalmalı. (tabloda rakam yerine * kullanan öğrenci buradan puan alabilir) Tablodaki değerlerin arasında "\t" (tab) konabilir. Bu parçayı yapan öğrencinin Parça1'i yapması gerekmez.
3. ***TABLODAKİ SAYILARI OLUŞTURABİLMEK(15 PUAN):*** Sayılar rakam(([0-9] aralığında) olarak ve rastgele üretilmiş olmalı. Kod her çalıştırıldığında aynı rakamları üretmiyor olmalı.
4. ***SATIR TOPLAMLARINI ÜRETEBİLMEK(10 PUAN):*** Her satırın sonunda o satırdaki rakamların toplam değeri gösterilmeli. Toplam değerinden hemen önce "ya da" işaretindeki dikey çizgi (|) ve iki boşluk kullanılabilir.
5. ***SÜTUN TOPLAMLARINI ÜRETEBİLMEK(10 PUAN):*** *(Algoritmik zorluğu ve göreceli olarak puanının düşük olması nedeniyle bu parçayı yapmayı sınavın sonuna bırakmanız mantıklı olabilir.)* Her sütunun altında o sütundaki rakamların toplam değeri gösterilebilmeli. Birden fazla basamağa sahip toplam değerlerinin basamakları farklı satırlarda gösterilmeli. Kullanıcıdan gelecek uzunluk değerine bağlı olarak maksimum basamak sayısı değişebilir. *İpucu: Hesaplanan toplam değeri, fprintf'in dizgi versiyonu olan sprintf ile sprintf(dizgi, "%d", toplam); şeklindeki gibi bir dizgiye işlenebilir.*
6. ***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: 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 – GEZGİN SATICI PROBLEMİ 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.**



Endüstride kaynakların optimal kullanımı lojistik açıdan çok büyük öneme sahiptir. **Gezgin satıcı problemi**, 1930'lu yıllarda matematiksel olarak modellenen bir problemdir. Bir satıcının değişik illere ulaşarak elindeki malları satması gerekmektedir. *"Bu iller arasındaki mesafeler verildiğinde, tüm illere uğrayan ve çıkış iline geri dönen en kısa rota hangisidir?"* Bu problem, tüm olasılıklar denenerek çözülebilir ancak faktöriyel olarak artış gösteren olasılık miktarı il sayısı arttıkça kısa sürede çok büyük rakamlara erişmektedir. Ne var ki, programlama ile bu sayılarla (çok çok büyük olmadığı sürece) baş etmek mümkündür. Gezgin satıcı probleminin uygulama alanlarına örnek olarak GSM operatörlerinin baz istasyonları için yer belirlenmesi, uçaklar için havaalanı rotalaması, elektronik devrelerin minimum güç tüketimi yapacak şekilde tasarımı verilebilir.

Bu soruda sizden Örnek çıktı-2.1'de gözüktüğü şekilde,

- kullanıcıdan sevkiyatın çıkış ilini (sadece aşağıdaki 5 ili göz önünde bulundurunuz) küçük harflerle alan,
- kullanıcıdan sevkiyatın hedefini küçük harflerle ve aralarında birer boşluk olan 3 il şeklinde alan (hedeflerin sırası önemli değildir - kullanıcının her zaman aralarında 1 boşluk bırakarak ve küçük harflerle gireceğini varsayınız),
- sevkiyatın çıkış ilinden başlayan, belirtilen 3 ile uğrayan ve çıkış iline geri dönen olası 6 farklı rotayı yazdıran,
- iller arası uzaklık tablosundaki verilere göre her rota için ardışık ifade edilen iller arasındaki uzaklıkların toplamını hesaplayan,
- bu toplamların minimumuna göre rota önerisi yapan,
- önerdiği rotalara ait *(**her 100 km için** bir tırın ortalama **40 lt mazot** harcadığını varsayarak)* mazot sarfiyatı ve *(mazotun **litresinin 6,4 TL** olduğunu varsayarak)* maliyetini ifade eden bir kod yazmanız istenmektedir.

Kullanıcının GEÇERLİ GİRİŞLER YAPACAĞINI (örn. il isimlerini her zaman küçük harflerle gireceğini) varsayınız.

Tablo 1 - İller Arası Uzaklık

İL PLAKA NO	İL ADI	ankara	bolu	kars	mus	van
06	ankara	0	191	1075	976	1194
14	bolu	191	0	1166	1157	1375
36	kars	1075	1166	0	350	365
49	mus	976	1157	350	0	218
65	van	1194	1375	365	218	0

```
Sevkiyatin cikis ili:ankara
Sevkiyatin hedefi:kars van bolu

Rota 1:ankara-kars-van-bolu-ankara
Uzunluk: 3006 km
Rota 2:ankara-kars-bolu-van-ankara
Uzunluk: 4810 km
Rota 3:ankara-bolu-kars-van-ankara
Uzunluk: 2916 km
Rota 4:ankara-bolu-van-kars-ankara
Uzunluk: 3006 km
Rota 5:ankara-van-bolu-kars-ankara
Uzunluk: 4810 km
Rota 6:ankara-van-kars-bolu-ankara
Uzunluk: 2916 km

Cevre ve butce dostu olan Rota 3 ya da Rota 6 oneriliyor.
- Tirin ortalama mazot sarfiyati 1166 lt olacaktir.
- Ortalama mazot maliyeti 7465 TL olacaktir.

--------------------------------
Process exited after 5.839 seconds with return value 0
Devam etmek için bir tuşa basın . . .
```

Örnek çıktı-2.1- Kullanıcı kod çalıştıktan sonra sevkiyatın çıkış ili olarak küçük harflerle "ankara" yazıp enter'a basmıştır. Ardından hedef olarak aralarında bir boşluk bırakarak "kars van bolu" yazıp enter'a basmıştır. İlk ve son noktaları ankara olan 6 farklı rota ve bu rotaların toplam uzunlukları hesaplanmıştır. Örneğin Rota1'de tabloya göre ankara-kars arası 1075 km'dir. Bu şekilde ardışık iller arasındaki uzaklıklar toplanınca toplam uzaklık 1075+365+1375+191= 3006 km çıkmıştır. Tüm rotalar yazdırıldıktan sonra minimum uzunluğa sahip iki rota önerilmiş ve bu rotalara ait bilgiler paylaşılmıştır. Bu şekilde çalışan bir kod yan kısımda gösterilen tüm parçalardan puan alabilir.

```
Sevkiyatin cikis ili:36
Sevkiyatin hedefi:49
Rota:36-49-36

Uzunluk: 700 km
- Tirin ortalama mazot sarfiyati 280 lt olacaktir.
- Ortalama mazot maliyeti 1792 TL olacaktir.

--------------------------------
Process exited after 1.892 seconds with return value 0
Devam etmek için bir tuşa basın . . .
```

Örnek çıktı-2.2- Kullanıcı kod çalıştıktan sonra sevkiyatın çıkış ili olarak "kars" iliinin plakası olan "36" değerini yazıp enter'a basmıştır. Ardından hedef olarak "mus" iliinin plakası olan "49" değerini yazıp enter'a basmıştır. İller Arası Uzaklık tablosunda kars-mus arası 350 km olduğu için, gidiş-geliş toplam 700 km değeri hesaplanmıştır. 40lt/100km ve 6,4TL/litre değerleri kullanarak çıktıda gösterilen değerler hesaplanmıştır. Bu şekilde çalışan bir kod yan kısımda gösterilen parçalar arasından sadece 2 nolu parçadan puan alabilir; diğer parçaları sağlamadığı için diğer parçalardan puan alamaz.

ÖNEMLİ UYARI: 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. ***İLLERİ HARFLERLE İFADELİ ŞEKİLDE ALABİLMEK(5 PUAN):*** Kullanıcı illeri Örnek Çıktı-2.1'deki gibi harflerle girebilmeli. Bu şekilde çalışan bir kodun aynı zamanda plaka nolarını(numaralarını) da tanıması **gerekmez**. Harflerle ifadeli şekilde alabilmeyi yapamayan öğrenciler bu parçadan puan alamaz ancak plaka no üzerinden(Örnek Çıktı-2.2'deki gibi) devam ederek, diğer parçalardan puan almaya çalışabilir.
2. ***İKİ İL ARASINDAKİ UZAKLIK VE DİĞER BİLGİLERİ BELİRLEYEBİLMEK (15 PUAN):*** Sade şekliyle Örnek Çıktı-2.2'deki gibi kullanıcıdan (plaka değerleri üzerinden de olsa) iki giriş alıp, istenen *toplam uzunluk*, *mazot sarfiyatı* ve *mazot maliyeti* değerlerini hesaplayabilmek.
3. ***ÜÇ HEDEF İLİ ALMAK VE BU İLLERDEN OLUŞAN 6 FARKLI ROTAYI OLUŞTURABİLMEK (5 PUAN):*** Kullanıcının her zaman 3 hedef ili gireceği varsayılacak ve 3!=6 farklı rota oluşturulacaktır. 6 farklı rotayı tek tek elle herhangi bir algoritmaya bağlı olmadan oluşturuyor olmak bu parçadan puan alabilmek için yeterlidir. Bu parçayı yapamayan öğrenciler Örnek Çıktı-2.2'deki gibi Parça 2'yi yapmaya çalışabilirler.
4. ***6 FARKLI ROTAYI SIRALI İKİLİ DEĞİŞİMLERLE OLUŞTURABİLMEK(10 PUAN):*** Bir rotadan yola çıkarak diğer rotayı oluştururken önce 3. ve 4. illeri kendi aralarında değiştirin, bir sonraki rotayı oluştururken de 2. ve 3. illeri kendi aralarında değiştirin ve bunu dönüşümlü olarak tekrarlayın. 1. ve 5. illerin hep aynı kalacağına dikkat ediniz. Örnek Çıktı-2.1'deki değişimleri inceleyerek bu algoritmanın nasıl çalıştığını takip edebilirsiniz.
5. ***5 İLDEN OLUŞAN HER ROTA İÇİN TOPLAM UZUNLUK DEĞERİNİ HESAPLAYABİLMEK (5 PUAN):*** Verilen bir rotada 5 il ve ardışık iller arasında 4 uzunluk değeri bulunmaktadır. Kodunuz bu uzunluk değerlerinin toplamanı hesaplayabilmeli. Bu işlemleri yapış tarzınıza göre Parça7'den de ilave puan alabileceksiniz.
6. ***EN KISA UZUNLUĞA SAHİP İKİ ROTAYI BELİRLEYEBİLMEK(5 PUAN):*** Bir rotayı tersten takip ettiğinizde bir diğer rota çıkacağı için her durumda minimum uzunluğa sahip rota sayısı en az 2 olacaktır. Minimum uzunluğa sahip rota sayısının hep 2 olacağını varsayabilirsiniz. 6 farklı rota arasından hangi ikisinin minimum uzunluğa sahip olduğu belirlenebilmeli. (Belirlenen rotaya dair değerleri hesaplayabilmek Parça 2'de puanlandırılır.)
7. ***GELİŞTİRİLEBİLİR BİR KOD YAZMIŞ OLMAK(5 PUAN+BONUS 5 PUAN):*** *Yukarıdaki parçalardan ikisi hariç hepsinden puan alabilmiş öğrenciler için geçerlidir.*

    a. Kod **parametrik** şekilde ifade edilmiş olmalı ve **diziler** etkin kullanılarak kodlanmış olmalı. İlave iller eklemek gerektiğinde, kodda ufak düzenlemelerle bu gerçekleşebilmeli. Örneğin, illeri algılamak için ilk harflerine bakmak gibi basit algoritmalar bu parçadan tam puan alamaz. İlleri algılamak için tüm harflerine bakan algoritma tercih edilmelidir. Çünkü ilave iller, mevcut illerden biriyle aynı harfle (örneğin "adana") başlıyor olabilir.

    b. Kod **fonksiyonlar** etkin kullanılarak kodlanmış ve rahat geliştirilebilir hale getirilmiş olmalı. Örneğin, illeri sıralı şekilde içeren diziyi alıp rota uzunluğunu hesaplayıp dönen bir fonksiyon kullanılabilir. Böylece 6 farklı rotanın her biri için aynı hesaplama kod parçası kodda 6 kez tekrarlanmaz. Her seferinde sadece bu fonksiyonun çağırılması yeterli olur.

OLASI ÇÖZÜM - SORU1

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <time.h>
int main()
{
	srand(time(NULL));
	int uzunluk;
	printf("Uzunluk:");
	scanf("%d", &uzunluk);
	int i,j;
	int dizi[uzunluk][uzunluk]; // değerleri kare bir dizinin üçgen yarısında saklayacağız.
	int sutunToplamlari[uzunluk]; // her sütunun toplamını tutar.
	for(i=0; i<uzunluk; i++)
	{
		sutunToplamlari[i]=0; // toplam hesabında kullanılacak, ilk değerler 0 olmalı.
	}

	//ÜÇGENLİ DİZİNİN RASTGELE DEĞERLERLE DOLDURULMASI
	for(i=0; i<uzunluk; i++)
	{
		for(j=0; j<uzunluk-i; j++) //uzunluk-i yazdığımız için sadece üçgen kısma değer
atıyoruz.
		{
			dizi[i][j] = rand()%10;
		}
	}

	printf("VERI ANALIZ TABLOSU:\n");
	for(i=0; i<uzunluk; i++) //her turda bir satırı yazdırır.
	{
		int satirinToplami=0;
		for(j=0; j<uzunluk; j++)
		{
			if(j<uzunluk-i) //uzunluk-i yazdığımız için sadece üçgen kısmı
yazdırıyoruz.
			{
				printf("%d", dizi[i][j]);
				satirinToplami+= dizi[i][j];
				sutunToplamlari[j]+= dizi[i][j];
				//satırları yazdırırken bir yandan da sütun toplamı oluşturuyoruz.
			}
			printf("\t"); //her durumda bir \t koyuyoruz.
		}
		printf("|  %d\n",satirinToplami); //bu değere bir daha ihtiyaç duymayacağız
	}
	printf("\n");
	for(j=0; j<uzunluk; j++) //süsleme kısmı - en alttaki - - - - - leri basar.
	{
		printf("-\t");
	}
	printf("\n");

	char dizgi[uzunluk][100];


	for(j=0; j<uzunluk; j++)
	{
		sprintf(dizgi[j], "%d", sutunToplamlari[j]);
			// 1234 halindeki int sayıyı "1234" halinde dizgi[j]'e kaydeder.

	}

	int maksUzunluk=0;
	for(j=0; j<uzunluk; j++) // sutunToplamlari arasındaki maksimum uzunluğa sahip olanı tespit
edelim.
	{
		int l = strlen(dizgi[j]);
		if(l>maksUzunluk )
			maksUzunluk = l;
	}
```

```
        for(i=0; i<maksUzunluk; i++)
        // maksUzunluk kadar satır basar, her satırda da her sütunun bir karakterini basar.
        {
                printf("\n");
                for(j=0; j<uzunluk; j++)
                        printf("%c\t",dizgi[j][i] );
        }


        return 0;
}
```

OLASI ÇÖZÜM - SORU2

```
#include <stdio.h>
#include <string.h>
int sozcuktenIndise(char ilismi[]); // ankara dersin 0 doner, bolu dersin 1 doner...
int km_hesapla(char ilIsmi[5][50], int yol[5][5]); // il dizilimine bakar, yol uzunluğunu verir.
void degistir(char dizgi1[], char dizgi2[]); // iki int'in yerini kendi içinde değiştirir.
int main()
{
        char ilIsmi[5][50];
        int yol[5][5]= // DİKKAT! ankara 0, bolu 1, kars 2, mus 3, van ise 4 indisiyle anilacaktir.
        {
                {0,             191,            1075,           976,            1194},
                {0,             0,              1166,           1157,           1375},
                {0,             0,              0,              350,            365},
                {0,             0,              0,              0,              218},
                {0,             0,              0,              0,              0}
        };
        int i,j;
        //tablonun geriye kalan kismini doldurur. A-B illeri arasi uzaklik ile B-A illeri arasi
uzaklik aynidir.
        for(i=0; i<5; i++)
        {
                for(j=0; j<i; j++)
                        yol[i][j]=yol[j][i];
        }

        int km=0;
        printf("Sevkiyatin cikis ili:");
        scanf("%s", ilIsmi[0]);
        strcpy(ilIsmi[4],ilIsmi[0]); //baslangic ve cikis illeri ayni olacak.
        printf("Sevkiyatin hedefi:");
        for(i=1; i<=3 ; i++)
        {
                scanf("%s", ilIsmi[i]);
        }

        printf("\n");

        int min = km_hesapla(ilIsmi,yol); //ilk basta rota1'i min varsayalim.
        int minRotaNo =1;
        int k;
        for(j=1; j<=6; j++)
        {
                printf("Rota %d:",j);
                for(i=0; i<5; i++)
                {
                        printf("%s", ilIsmi[i]);
                        if(i!=5-1) //ekran suslemesi ankara - van - 'lerdeki - 'leri koyar.
                                printf("-");
                        else
                                printf("\n");
                }

                km = km_hesapla(ilIsmi,yol); //mevcut il diziliminin uzunlugunu verir.
                printf("Uzunluk: %d km\n", km);

                if(km<min) // il dizilimleri arasindaki minimum uzunlugu olani tespit icin
                {
                        min=km;
                        minRotaNo= j;
                }

                if(j%2==1) // permutasyon icin - farkli rota dizilimi uretir.
                {
                        degistir(ilIsmi[2],ilIsmi[3]);
                }
                else
                {
                        degistir(ilIsmi[1],ilIsmi[2]);
                }
        }
```

```
        printf("\nCevre ve butce dostu olan Rota %d ya da Rota %d oneriliyor.\n", minRotaNo,
minRotaNo+3);
        printf("- Tirin ortalama mazot sarfiyati %.0lf lt olacaktir.\n", km*0.40);
        printf("- Ortalama mazot maliyeti %.0lf TL olacaktir.\n", km*0.40*6.40);

        return 0;
}

int sozcuktenIndise(char ilismi[])
{
        char iller[5][50]= {"ankara", "bolu", "kars", "mus", "van"};
        int i;
        for(i=0; i<5; i++)
        {
                if(!strcmp(iller[i], ilismi))
                        return i;
        }
        return -1;
}

int km_hesapla(char ilIsmi[5][50], int yol[5][5])
{
        int il[5];
        int i;
        for(i=0; i<5 ; i++) //butun il sozcukleri, il indislerine cevrildi. ankara 0 oldu, bolu 1
oldu...
                il[i] = sozcuktenIndise(ilIsmi[i]);
        int km=0;
        for(i=0; i<5-1; i++) // içeride i+1 geçtiği için -1 geliyor. Böylece i+1 maksimum 4
değerini alır.
        {
                km += yol[ il[i] ][ il[i+1] ];
        }
        return km;
}

void degistir(char dizgi1[], char dizgi2[])
{
        char temp[50];
        strcpy(temp, dizgi1);
        strcpy(dizgi1, dizgi2);
        strcpy(dizgi2, temp);
}
```