



# TOBB ETÜ

Ekonomi ve Teknoloji Üniversitesi

## ÇALIŞMA SORULARI 4

### SORU 1 – HESAP MAKİNESİ



4 temel matematik işlemini (toplama, çarpma, çıkarma, bölme) **ondalıklı sayılar üzerinde** yapan bir hesap makinesi kodu yazmanız istenmektedir. Kodunuz **-1 ile çıkış yapılması istenene kadar SÜREKLİ** çalışacaktır: Kodunuz şu şekilde bir menü ile çalışacaktır:

```
HESAP MAKİNESİ KODUNA HOŞGELDİNİZ
1. Toplama
2. Çıkarma
3. Çarpma
4. Bölme
Yapmak istediğiniz işlemi seçiniz. Çıkmak için "-1" giriniz.
Seciminiz: 4
Birinci sayiyi giriniz:9.0
İkinci sayiyi giriniz:4.0
İşlem sonucu: 2.250

1. Toplama
2. Çıkarma
3. Çarpma
4. Bölme
Yapmak istediğiniz işlemi seçiniz. Çıkmak için "-1" giriniz.
Seciminiz: -1
Güle güle!!! Program sonlandı...
```

Şekil 1 - Program çıktısına bir örnek. Yazmış olduğunuz hesap makinesi kodu programdan çıkış yapılması istenene kadar çalışmaya devam edecektir. Programdan "-1" girişi ile çıkış yapılacaktır.

**DEĞER KONTROLÜ ÖZELLİĞİ:** Programda belirtilen seçeneklerden(1,2,3,4 ya da -1) başka bir giriş yapıldığında program uyarı verecek ve **tekrar giriş** yapılması istenecektir. (Kullanıcının her zaman tam sayı gireceğini varsayabilirsiniz.)

```
4. BÖLME
Yapmak istediğiniz işlemi seçiniz. Çıkmak için "-1" giriniz.
Seciminz: 9
Gecerli bir giris yapiniz!!!
Seciminz: 0
Gecerli bir giris yapiniz!!!
Seciminz: 2
9 Birinci sayiyi giriniz:
```

Şekil 2 - Kullanıcının girişine DEĞER KONTROLÜ (bkz. while slaytları) uygulanacaktır.

### ÇOK ÖNEMLİ KISIM:

Sizden istenen 4 işlem aslında çok sade oldukları için main() içerisinde yapılabilirler. Ama bu soruda fonksiyon kullanımına iyice alışmanız için her işlem için main()fonksiyonu yerine **ayrı birer fonksiyon** (main'e ek olarak toplamda 4 fonksiyon) tanımlamanızı ve işlemleri bu fonksiyonlara yaptırmanızı istiyoruz.

```
#include <stdio.h>
double topla(double a, double b);
double cikar(double a, double b);
double carp(double a, double b);
double bol(double a, double b);
int main()
{
```

Şekil 3 - Örnek bir kod başlangıcı

HATIRLATMA: 4 fonksiyon da double türünde değer döndüğü için fonksiyonların sonunda **return komutunu** kullanmayı unutmayın. Bazen bu komut kullanılmadığında da istenen değer dönebilirse de lütfen bu sizi yanıltıcı olmasın: void dönen fonksiyonlar haricinde return kullanılması **ZORUNLUDUR!**

**LÜTFEN ÇÖZÜMÜNÜZÜ İYİCE KONTROL EDİNİZ!**

- Kodunuz dört işlemi (ondalıklı sayılar girildiğinde) **doğru hesaplayabiliyor** mu?
- Kodunuz menüden **-1 girilene kadar** sürekli yeni işlemleri istiyor mu?
- Kodunuz menüden **-1 girildiğinde sonlanıyor** mu?
- Kodunuz menüden 0, 5 ya da 9 girildiğinde **tekrar değer istiyor** mu?
- Kodunuz menüden (sadece 1 kez değil) **peş peşe yanlış girişler** yapılırsa da her seferinde tekrar değer istiyor mu?
- Kodunuzda **fonksiyonları** prototip, çağırılış ve tanımlarıyla doğru şekilde kullandınız mı?
- Fonksiyonlarınızın sonunda **return komutu** ile istenilen değeri dönüyor musunuz?
- "*Menüden **harf girildiğinde** de kodunuz tekrar değer isteyebilsin*" denilseydi ne yapmanız gerekirdi?
- "*Kodunuz **tam sayılarla da** doğru şekilde çalışabilsin*" denilseydi kodunuzda bir değişiklik yapmanız gerekecek miydi?
- Kodunuzda "**goto**" ya da **küresel değişken** kullanımı olmadığına emin misiniz?



## SORU 2 – BONUS #1



HESAP MAKİNESİ sorusunda çıkış yapıldığında **işlemlerin özeti** sunulsun (ipucu: dizi kullanımı).  
Maksimum 1000 işlem yapılacağını varsayınız.

```
HESAP MAKİNESİ KODUNA HOŞGELDİNİZ
1. Toplama
2. Çıkarma
3. Çarpma
4. Bölme
Yapmak istediğiniz işlemi seçiniz. Çıkmak için "-1" giriniz.
Seciminiz: 1
Birinci sayiyi giriniz:5.2
İkinci sayiyi giriniz:1.4
İşlem sonucu: 6.600

1. Toplama
2. Çıkarma
3. Çarpma
4. Bölme
Yapmak istediğiniz işlemi seçiniz. Çıkmak için "-1" giriniz.
Seciminiz: 3
Birinci sayiyi giriniz:9.2
İkinci sayiyi giriniz:0.5
İşlem sonucu: 4.600

1. Toplama
2. Çıkarma
3. Çarpma
4. Bölme
Yapmak istediğiniz işlemi seçiniz. Çıkmak için "-1" giriniz.
Seciminiz: -1
-----
İşlem özeti:
1.İşlem: 5.200+1.400=6.600
2.İşlem: 9.200*0.500=4.600
-----
Güle güle!!! Program sonlandı..._
```

Şekil 4 - İşlemler sonucunda özet bilgi sunulması istenmektedir.

**LÜTFEN ÇÖZÜMÜNÜZÜ İYİCE KONTROL EDİNİZ!**

- Kodunuzu değişik değerlerle bol bol **test ettiniz** mi?
- Kodunuzda "**goto**" ya da **küresel değişken** kullanımı olmadığına emin misiniz?

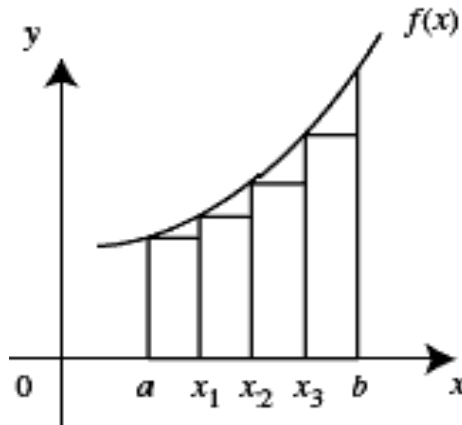


## SORU 3– SAYISAL YÖNTEMLERLE İNTEGRAL

Bu soruda, sizden, **verilen fonksiyonun** integralini, yazacağınız kod sayesinde hesaplamanızı beklemekteyiz. İntegral işlemi belirlenen sayı değerleri **[a, b]** arasında yapılacak ve **[a, b]** aralığında fonksiyonun integrali sayısal yöntemlerle hesaplanacaktır.

**UYARI: Bu soru,** geçtiğimiz dönemlerde deneme-yanılma yöntemiyle kodlama yapan öğrencileri **çok yoran bir soru tipidir.** Sayısal yöntemlerle hesaplamalar mühendisliğin önemli bir parçası olduğu için bu aşamada size bir giriş yaptırmak için bu soru tipi tercih edilmiştir. Çözmeye **ERKENDEN** başlayın, önce sorunun mantığını anlayın, sonra çok dikkatlice kodlayın.

Sayısal yöntemlerle integral hesabını iyi şekilde anlamak için aşağıdaki grafiği inceleyiniz. Şekilde, fonksiyonun altında kalan alanın **[a,b]** sayı aralığı arasında eşit genişlikteki dilimlere ayrıldığını görmekteyiz.



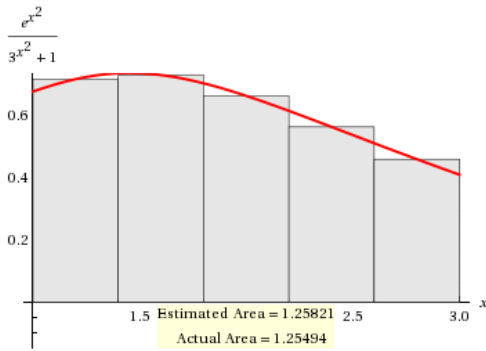
Şekil 5 - Bir fonksiyonun grafiği ve altındaki alanların gösterimi

Dilimleme yöntemi (Riemann toplamı) ile integral alma işlemi şu şekilde gerçekleşmektedir:

1. Verilen **f(x)** fonksiyonunun altında kalan alanın kaç adet dilime ayrılacağı belirlenir.
2. **f(x)** fonksiyonunun hangi değer aralıkları arasında integralinin alınacağı seçilir. Yani, **[a,b]** sayı aralığı belirlenir.
3. **[a,b]** sayı aralığı arasında belirlenen dilimlerin her birinin genişliği birbirine eşit olacaktır. (Dilimleme yönteminde dilim sayısı ne kadar fazla olursa, integral sonucu gerçeğe o kadar yakın çıkacaktır.)
4. Ortaya çıkan her bir dilimin alanı hesaplanır. Dilimlerin alan değerini hesaplarırken, dilimlerin orta noktasını, yani ilgili dilimin sol ve sağ uçlarının tam ortasını kullanacaksınız. Örneğin, Şekil 6'te verilen grafikte birinci dilimin alanı şu şekilde hesaplanır: **A<sub>1</sub>=dilimGenisligi×f(a ile x<sub>1</sub>'in ortası)**. Devamında, ikinci dilimin alanı şu şekilde hesaplanır: **A<sub>2</sub>=dilimGenisligi×f(x<sub>1</sub> ile x<sub>2</sub>'nin ortası)**. Bu tür işlemler, bütün dilimler için tekrar edilerek her bir dilimin alanı hesaplanır ve bütün dilimlerin alanları toplanır. Bu soruda, dilimlerin alanları hesaplanırken orta noktaları referans olarak alınmaktadır. Ancak, dilimlerin sol- ve sağ- noktalarının kullanıldığı dilimleme yöntemleri de bulunmaktadır.
5. Elde edilen toplam, fonksiyonun belirlenen **[a, b]** sayı aralığında integral sonucunu vermektedir.

Dilimleme yöntemi hakkında anlatılanları dikkatli bir şekilde okuyup, inceledikten sonra kod yazma işlemine artık başlayabilirsiniz. Bu soruda, sizden, yukarıda verilen adımları referans alarak, aşağıdaki fonksiyonun integralini almanızı beklemekteyiz.

$$f(x) = \frac{e^{x^2}}{1 + 3x^2}$$



Graph the Riemann sum of

as x goes from  to

using  rectangles

taking samples at the

Print estimated and actual areas?

Rectangle Color  Plot Color

POWERED BY  
webMathematica

Şekil 7 - <http://mathworld.wolfram.com/RiemannSum.html> adresinden alınan ekran görüntüsü (Verilen bağlantı adresindeki hesaplayıcı ile yazdığınız kodun doğruluğunu test edebilir, dilimleme yöntemi hakkında daha fazla bilgiye erişebilirsiniz.)

Yazacağınız kod kullanıcıdan sırasıyla **1) integralin hesaplanacağı uç noktaları**, yani **a, b** sayılarını **2) dilim sayısını** alacaktır. Daha sonra, **aşağıdaki gibi, integral sonucunu ekrana yazdıracaktır.** Bu işlemler sırasında, size örneklerde verilen sade formata uymaya özen gösteriniz.

```
SAYISAL YONTEMLE INTEGRAL!  
Sol uc:1.0  
Sag uc:3.0  
Dilim:5  
SONUC: 1.258209  
-----  
Process exited after 6.464 seconds with return value 0  
Devam etmek için bir tuşa basın . . . █
```

**Şekil 8 - Kullanıcı verilen fonksiyonun [1.0-3.0] aralığındaki integralinin 5 dilimli hesabını yaptırmak üzere sırasıyla 1.0, enter, 3.0, enter, 5, enter girişleri yapmıştır.**

```
SAYISAL YONTEMLE INTEGRAL!  
Sol uc:1.50  
Sag uc:3.14  
Dilim:2  
SONUC: 0.955131  
-----  
Process exited after 7.327 seconds with return value 0  
Devam etmek için bir tuşa basın . . . █
```

**Şekil 9 - Kullanıcı verilen fonksiyonun [1.50-3.14] aralığındaki integralinin 2 dilimli hesabını yaptırmak üzere sırasıyla 1.50, enter, 3.14, enter, 2, enter girişleri yapmıştır.**

### **ÇOK ÖNEMLİ KISIM:**

- Yazacağınız kod, tam olarak ***double riemann(double, double);*** prototipine sahip bir fonksiyon **İÇERMELİDİR**. *double riemann(double, double, double);* vb. şekillerde **DEĞİL!**
- Bu fonksiyon giriş parametresi olarak istenen aralığın sol ve sağ uç değerlerini alacaktır. İntegral sonucunu hesaplama operasyonunu bu fonksiyon içerisinde yapacaksınız. Bu fonksiyonun çıkış parametresi, yani döndüreceği double değer ise **hesaplanan integralin sonucu** olacaktır. **Başka türlü kurgulanan ve kodlanan fonksiyonlar aynı sonucu veriyor olsa bile istenmemtedir.** Buradaki amaç fonksiyonları kodunuzda etkin bir şekilde kullanmanızdır.
- Asla **küresel değişken** ( fonksiyonların dışında tanımlı değişken, #include'lardan hemen sonra tanımlı değişken ...) **KULLANMAYIN!**

```
#include <stdio.h>  
#include <math.h>  
double riemann (double, double);  
int main()  
{  
    printf("SAYISAL YONTEMLE INTEGRAL!\n");
```

**Şekil 10 - Örnek bir kod başlangıcı**

Matematiksel olarak takıldığınız ufak yerlerde **saatlerce vakit harcamayın**, [su gibi](#) kaynaklardan konu yardımı alın, gerekiyorsa asistanlara e-posta ile sorup cevap alın.

"e üzeri x kare" fonksiyonunu **pow(M\_E, ...** şeklinde math.h kütüphanesiyle birlikte kullanın. e sabitini elle tanımlamak, exp fonksiyonunu kullanmak... ufak miktarda farklı sonuçlar üretebilir.

double'ların ondalıklı sayıları ideal anlamda temsil edemediğini unutmayın.

```
printf("%.20lf\n", 2.2+0.2);
```

```
printf("%.20lf\n", 2.4);
```

```
2.400000000000000040000
```

```
2.39999999999999990000
```



Bu nedenle kodunuzda  
x= 2.4; if(x == 2.2+0.2) ...

gibi ifadeler kullanmamaya çalışın.

### LÜTFEN ÇÖZÜMÜNÜZÜ İYİCE KONTROL EDİNİZ!

- **Soru metnini** tamamen(kısmen değil) okuyup, sizden istenenleri anladığınıza emin misiniz?
- İstenen şekilde double türünde **SADECE İKİ TANE** (üç değil) girdi alan bir fonksiyon kullandınız mı?
- Kodlama yaparken takıldığınız noktalarda slaytlarda anlatılan **hata ayıklama tekniklerini** (geçici printfler vs.) kullanıyor musunuz?
- Kullanıcı verilen fonksiyonun [1.0-3.0] aralığındaki integralinin 5 dilimli hesabını yaptırmak istediğinizde kodunuz 1.258209 değerini **buluyor mu?**
- Kullanıcı verilen fonksiyonun [1.50-3.14] aralığındaki integralinin 2 dilimli hesabını yaptırmak istediğinizde kodunuz 0.955131 değerini **buluyor mu?**
- Size adresi verilen internet sitesindeki değişik aralıklara ait sonuçlar ile kendi sonucunuzu **karşılaştırdınız** mı? Karşılaştırdığınız sonuçlar birbirleri ile örtüşüyor mu?
- **double değerlerle kıyaslama** ifadeleri oluşturmak niçin tehlikelidir?
- Aynı kodu **hiç fonksiyon kullanmadan** da yazabilir miydik? Yazabilirsek, o halde **niçin** fonksiyon kullanıyoruz?
- Kodunuzda **"goto" ya da küresel değişken** kullanımı olmadığına emin misiniz?



## SORU 4 – BONUS



SAYISAL YÖNTEMLERLE İNTEGRAL sorusundaki şu adımlarla geliştiriniz:

1. İntegral işlemi hesaplanıp sonucu ekrana bastırıldıktan sonra, kullanıcıya devam etmek isteyip istemediğini soran özelliği kodunuza ekleyiniz. Yani, integral işlemi yapıldıktan sonra programı çalıştıran kullanıcıya "**Tekrar hesaplamak ister misiniz(e/h):**" sorusu sorulacak ve kullanıcının 'e' – 'h' girişlerine göre integral alma işlemi ya baştan başlayacak ya da program sonlanacaktır. (ipucu: do while 'ın güzel bir uygulaması)
2. 'sol uç' ve 'sağ uç' değerleri, klavyeden kullanıcı tarafından girilmetedir. Normalde, bir sayı doğrusu üzerinde sol uç olarak girilen değer sağ uç değerden mutlaka küçük olması beklenmektedir. Ancak bazı kullanıcılar, sol uç değeri, sağ uç değerden büyük girebilir ve integralin yanlış hesaplanmasına neden olabilirler. Bu olası hatalı girişlere karşı kodunuzda şu şekilde bir önlem alınız: sol uç değer, sağ uç değerden büyük girildiğinde, kodunuz **hangisinin sol veya sağ uç değer olduğunu anlayabilmelidir.**

Örneğin;

Sol uc:-5.1 (doğru giriş)

Sag uc:6.8

veya

Sol uc:6.8 (hatalı giriş)

Sag uc:-5.1

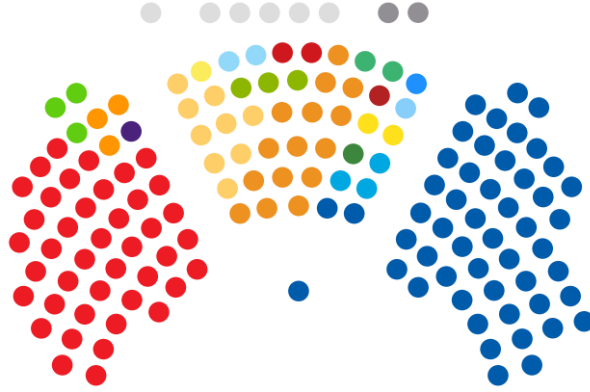
girişlerinde, sonucu hesaplanan integral değerlerinin **birbirleri ile aynı olması** gerekmektedir. Böylece kullanıcının bu şekildeki hatalı girişlerini kodunuz kendi içinde telafi edebiliyor olmaktadır.

### LÜTFEN ÇÖZÜMÜNÜZÜ İYİCE KONTROL EDİNİZ!

- Kodun sonunda 'e' girişi yaparak **devam** ve 'h' girişi yaparak **çıkış** sağlayabiliyor musunuz?
- [6.8,-5.1] aralığı girilmek istendiğinde çıkan sonuç, [-5.1,6.8] girildiğinde çıkan sonuçla **aynı mı?**
- Kodunuzda "**goto**" ya da **küresel değişken** kullanımı olmadığına emin misiniz?



## SORU 5 - MİLLETVEKİLİ ATAMA SORUSU



Ülkemizdeki seçimlerde milletvekili dağılımlarında dünyada da sıklıkla kullanılan D'hondt yöntemi tercih edilmektedir. Bu sayede milletvekilleri partilere tam sayılı şekilde dağılmaktadır.

1) Şu videonun 1:00-2:00 dakika aralarında güzelce anlatılmış bu yöntemi izleyiniz.

<https://youtu.be/FzspzvkdIrY>

2) Bir while döngüsü ve fonksiyonların etkin kullanımı ile durumu kodlayın.

3) Aşağıdaki yayında toplamda 5, 6, 7, 8 ya da 11 milletvekili olduğunda verilen oylara göre dağılımlar gözükmemekte. Kodunuzu bu değerlerle test edin.

<http://www.ucl.ac.uk/~ucahhwi/dhondt.pdf>

### LÜTFEN ÇÖZÜMÜNÜZÜ İYİCE KONTROL EDİNİZ!

- **Soru metnini** tamamen(kısmen değil) okuyup, sizden istenenleri anladığınıza emin misiniz?
- Bu kodda **fonksiyon ve döngüleri etkin şekilde** kullanabildiniz mi?
- Kodunuzda "**goto**" ya da **küresel değişken** kullanımı olmadığına emin misiniz?

