

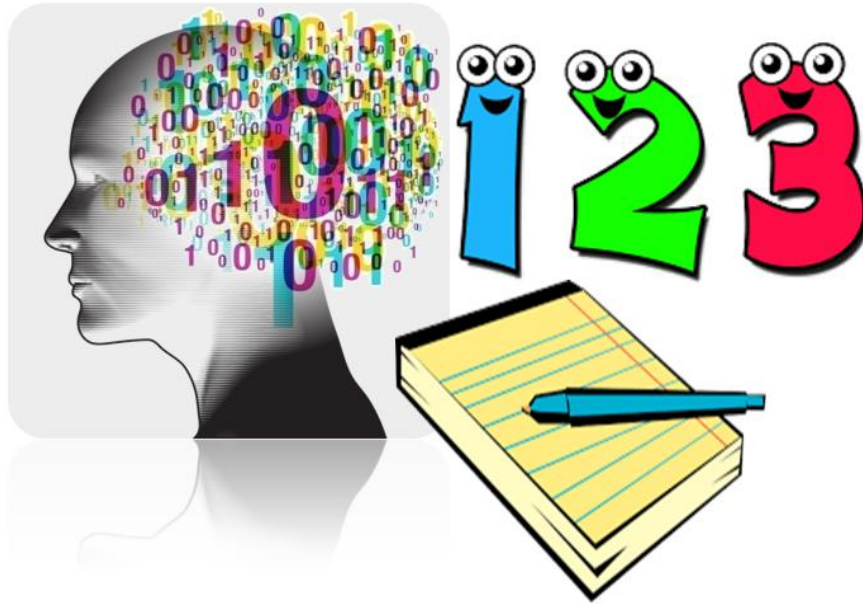
SINAVA YÖNELİK ÇALIŞMA SORULARI 6



TOBB ETÜ

Ekonomi ve Teknoloji Üniversitesi

SORU 1 –SAYI OYUNU



Bu soruda bir oyun tasarlamamız istenmektedir. Sayı oyunu isimli bu oyun şu şekilde oynanmaktadır:

- ▶ Bilgisayar **RAKAMLARI BİRBİRİNDEN FARKLI** ve **İLK RAKAMI 0 OLMAYAN** (basamak sayısı önceden belirlenmiş, örneğin 4 basamaklı) bir sayı tutar.
- ▶ Tahminde bulunursunuz.
- ▶ Bilgisayar tahmininiz ile kendi sayısını şu şekilde karşılaştırır:
 - ▶ iki rakam eşleştiyorsa ama sıra noları farklı ise $\rightarrow -1$
 - ▶ iki rakam eşleştiyorsa ve sıra noları da aynı ise $\rightarrow +1$
 - ▶ örn 1687 ve 2346 karşılaştırması -1 üretir çünkü 6 rakamı 1.sayıda 2.sırada, 2.sayıda 4.sırada bulunmaktadır.
 - ▶ örn 1687 ve 2690 karşılaştırması $+1$ üretir çünkü 6 rakamı 1.sayıda 2.sırada, 2.sayıda da 2.sırada bulunmaktadır.
- ▶ Sonuçlar $+2$, -2 , $+3$, -1 ... gibi olabilir. $+$ ve $-$ değerleri birbirini götürmeden ifade edilir.
- ▶ Bu sonuçları inceleyerek bir sonraki tahmininizi şekillendirirsiniz ve bu şekilde bilgisayarın tuttuğu sayıyı bulmaya çalışırsınız.

ÖRNEK:

Oyun 4 basamaklı oynanıyor olsun. Bilgisayar **1234** sayısını tutmuş olsun.

- ▶ Tahmininiz: 5678
- ▶ Bilgisayarın cevabı: 0
- ▶ Tahmininiz: 4321
- ▶ Bilgisayarın cevabı: -4
- ▶ Tahmininiz: 1243
- ▶ Bilgisayarın cevabı: +2-2
- ▶ Tahmininiz:1234
- ▶ Bilgisayarın cevabı: TEBRİKLER +4!

Bu çalışmanın ödev olarak verildiği bir döneme ait açıklama videosunu inceleyebilirsiniz:

<https://youtu.be/ut0TzSheHog>

Oyunu iki şekilde tasarlayabilirsiniz:

- TEMEL: Sayı oyununu 4 basamaklı oynanacak şekilde tasarlayınız.
- GELİŞMİŞ: Sayı oyununun kaç basamaklı oynanacağı kodun başında **#define BASAMAK ?** ile kod çalıştırılmadan belirlenebilsin.

TEMEL VERSİYON	GELİŞMİŞ VERSİYON
1. Kod rakamları birbirinden farklı olan, kullanıcının tahmin etmesi istenen 4 basamaklı sayıyı üretir. 2. Kullanıcıdan tahmin girişlerini alır. 3. Bilgisayar alınan tahminle kendi sayısını karşılaştırır. 4. Karşılaştırma sonucuna göre uygun cevap (+1, +2-2, 0 vs. gibi) verilir. 5. 2.,3. ve 4. maddeyi kullanıcı doğru sayıyı girene kadar tekrarlar. 6. Kullanıcı doğru sayıyı girdikten sonra tebriklerini sunar.	1. Kodun başındaki #define BASAMAK ? ifadesindeki ? yerine yazılan sayıya bağlı olarak kodun kaç basamaklı oynanacağı belirlenir. • #define BASAMAK 6 ise kod çalışınca bilgisayar 185462 gibi bir sayı tutar ve kullanıcı da 6 basamaklı değerler girer 2. <i>Ancak basamak 1 ise şöyle yap, basamak 2 ise şöyle yap, ... basamak 10 ise şöyle yap...</i> şeklinde KÖTÜ bir kod yazmayınız. Bu tür bir kullanım oldukça acemidir. Parametrik düşünerek dikkatlice kodlayınız.

- Derste öğrendiğimiz şeylerle hoşunuza gidecek şekilde bir çalışma ortaya çıkartın.
- Kodun başına #define DEBUG 1 ve main'in içerisine de

```
if(DEBUG)
{
    printf("Bilgisayarın tuttuğu sayı ... //bu sayıyı ekranda gösteren kod
    getch());
}
```

koyun, böylece DEBUG 1 iken kodun başında bilgisayarın tuttuğu değer gözüksün.

- *if(a==b || a==c || a==d...*

tarzı gibi elle tüm koşulları yazma amatörlüğünden artık **döngü/dizi kullanma aşamasına** geçiniz. Örneğin, for döngüsü içerisinde *if(dizi[i] == dizi[i+1])* gibi kullanımlarla algoritmanızı koda dökmeye çalışın. Özellikle gelişmiş versiyonu yaparken bu şekilde çalışmanın avantajlarını gözlemlersiniz.

- Fonksiyon kullanmadan da tüm kodu main'e yığarak işinizi yapabilirsiniz. Bu, bir şirketin tüm işlerini genel müdüre yaptırmaya benzer... **300 satırlık kodda kahramanlık yapmak kolaydır.** Mesele 300 satırlık kod yazarak öğrendiğiniz programlama dili dersimizde, ileride 30000 satırlık kod yazarken kullanabileceğiniz bakış açısı, yetenekler ve alışkanlıklar edinebilmek... **SONUÇ: Fonksiyonları kullanmaya çalışın!**
- **Kodunuzu geliştirin...**



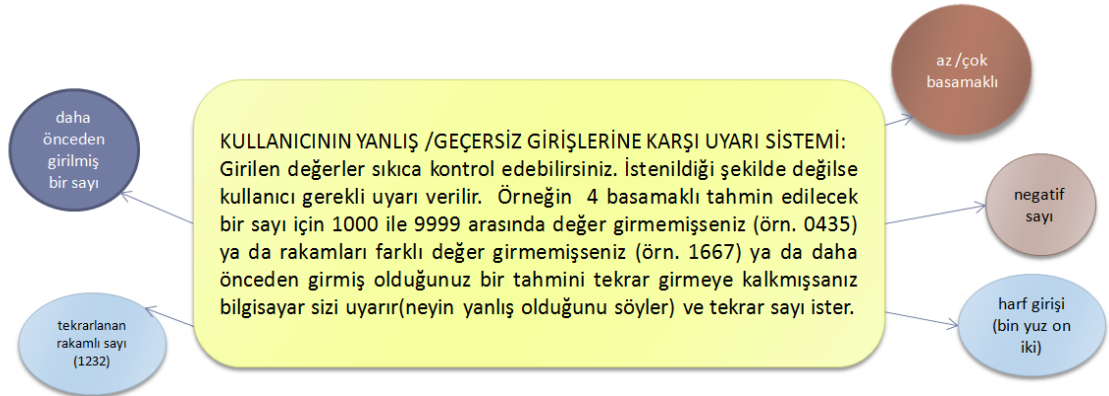
Şu üç konuyu etkin olarak kullanmaya çalışın.

- ▶ **Fonksiyonlar:** Girilen sayıyı parçalayıp diziye yerleştirme vs. gibi kod içinde tekrarlayan işlemleri fonksiyon haline getirerek, kodunuzu daha akıcı ve daha kolay geliştirilir hale getirebilirsiniz.
- ▶ **Döngüler:** Döngüler ile değer kontrol, sayı ile tahmini kıyaslama gibi işlemleri yapabilirsiniz.
- ▶ **Diziler:** Sayıları dizide saklarsanız döngüler ile işlemleri yapmanız kolaylaşır.



- ▶ **İPUCU:** Tahmin edilecek sayının her bir basamağının birbirinden farklı oluşması için algoritma önerisi:
 - Girilen basamak değeri kapasitesinde bir dizi üretilir. ($dizi[i]$)
 - Dizinin ilk basamağına ($dizi[0]$) rastgele değer üretilir.
 - ilk basamak sıfır olamaz.
 - $dizi[1]$ üretilir.
 - $dizi[1]$, $dizi[0]$ ile **aynı olduğu sürece** $dizi[1]$ tekrar üretilir.
 - $dizi[2]$ üretilir.
 - $dizi[2]$, $dizi[0]$ veya $dizi[1]$ ile **aynı olduğu sürece** $dizi[2]$ tekrar üretilir.
 - ...
 - $dizi[k]$ üretilir.
 - $dizi[k]$, $dizi[0]...$ veya $dizi[k-1]$ ile **aynı olduğu sürece** $dizi[k]$ tekrar üretilir.

- Kodunuza gelişmiş değer kontrolü özelliği ekleyebilir, böylece kullanıcı kaynaklı hatalara karşı kodunuzun çökmesinin önüne geçersiniz.



- Benzer satırları sürekli tekrar etmez
 - Döngüler, diziler, fonksiyonlar ile...**kod tekrarlarının önüne geçer.**
- Tüm işi main'e yaptırmak yerine toplu fonksiyonlara atar ve main'den onları **kumanda eder.**
- Koddaki detaylar, kodun başında yorum bloğu ile ve satır aralarındaki **yorum** satırları ile açıkça ifade edilir.
- İyi bir kod, değişik girdilerle birçok defa **test edilmiş** ve eksikleri giderilmiş koddur.

LÜTFEN ÇÖZÜMÜNÜZÜ İYİCE KONTROL EDİNİZ!

- Kodun başına detaylı bir yorum ile kodun özelliklerini(yazan kişi, tarih, kodun özellikleri, artıları, ileride geliştirilmesi gereken yanları, tasarımı vs.) anlattınız mı?
- Kodunuzdaki satırlara, fonksiyonların başlarına vs. bol bol açıklama yorumu eklediniz mi? Öyle ki *“kodu yıllar sonra açsam birkaç dakikada nasıl çalıştığını hatırlayabilirim”* diyebiliyor musunuz?
- Kodunuzla sayı oyunu düzgün şekilde oynanabiliyor mu?
- Oyununuzu iyi bir şekilde (en az 5 kez oynayarak) test ettiniz mi?
- Kodunuz oyunun başında RAKAMLARI BİRBİRİNDEN FARKLI sayı üretebiliyor mu?
- DEBUG değeri 1 atandığında oyunun başında bilgisayarın tuttuğu sayı gösteriliyor mu?
- Kodunuz +1 değeri (bu durumda +1 -0 yazmamasına özen gösteriniz) üretebiliyor mu? (Örn. 1234 değerini tahmin ederken 1987 girdiğinizde)
- Kodunuz +1 -1 (toplayıp da 0 yazarak değil) değeri üretebiliyor mu? (Örn. 1234 değerini tahmin ederken 1487 girdiğinizde)
- 4 basamaklı modda iken kodunuz +3-1 değeri üretebiliyor mu? (üretmemesi lazım)
- Derste öğrendiğimiz konuları kodunuza yansıtmaya çalıştınız mı?
- Döngüleri etkin şekilde kullandınız mı? Döngü kullanarak kısaltabileceğiniz yerlerde hep döngü kullandınız mı?
- Fonksiyonları etkin şekilde kullandınız mı? Fonksiyonları fonksiyon kullanmış olmak için değil, **etkin şekilde kullanarak** kodun tamamını kısa ve geliştirilebilir kıldığınızı düşünüyor musunuz?
- Dizileri etkin şekilde kullandınız mı? *“Dizi kullanmak mümkünken, onlarca değişkenle kodlama yapmaya çalışmadım. Dizileri etkin kullandığımı düşünüyorum.”* diyebiliyor musunuz?
- Kodunuzda goto ya da küresel değişkenler gibi sakıncalı kullanımlar kullanmamaya özen gösterdiniz mi?

