



ÜCRETSİZ ÖRNEK · KİTAPTAN 15 SAYFA

Olimpiyat için C Programlama

TÜBİTAK Ortaokul Bilgisayar 1. aşama sınavının istediği beceriyi — kodu kâğıtta adım adım izlemeyi — öğreten basılı çalışma kitabı.

- ✓ 400 sayfa · 10 modül · 40 ders
- ✓ 211 soru, her biri adım adım çözümlü
- ✓ 45 çıkmış benzeri soru, 2023-2026 sınav kalıplarıyla

BU DOSYADA

- s. 5-6** İçindekiler — kitabın tamamı
- s. 70-75** Döngüler modülünden bir dersin tamamı: konu anlatımı, izleme tabloları ve 5 sınav sorusu
- s. 306-307** O 5 sorunun adım adım çözümleri
- s. 180-183** İşaretçiler dersinin konu anlatımı: kutu ve ok çizimi, sık yapılan hatalar
- s. 43** Akış diyagramı soruları — çözümleri bu dosyada değil, kitapta

Nasıl okumalı? Soruları önce kendin çöz, çözüm sayfasına sonra bak. Kodu kafadan değil, kâğıda izleme tablosu çizerek izle — kitap bu alışkanlığı kazandırmak için yazıldı.

Kitabı incele ve sipariş et →

Kargo dahil · 15 gün içinde kargoda · 14 gün koşulsuz iade · 1e1kod.org/olimpiyat-kitabi

İçindekiler

Önsöz	3
Kitabı Nasıl Kullanmalısın?	7
1. Aşama Sınavını Tanıyalım	9
Modül 1 · Temeller ve Operatörler	11
Ders 1 Bir C Programının İskeleti	12
Ders 2 Ekrana Yazdırma, Okuma ve Sabitler	17
Ders 3 Artırma, Azaltma ve Operatör Önceliği	22
Ders 4 Tip Dönüşümü, Ternary ve Matematik Kütüphanesi	28
Modül Tekrar Testi	33
Modül 2 · Akış Diyagramları ve Karar	36
Ders 1 Akış Diyagramları: Sıralı, Şartlı, Döngülü	37
Ders 2 Şartsal ve Mantıksal İşleçler	44
Ders 3 if / else if / İç İç Koşul	50
Ders 4 switch ve Fall-through	55
Modül Tekrar Testi	61
Modül 3 · Döngüler	64
Ders 1 while ve do-while	65
Ders 2 for ve İç İç for	70
Ders 3 break, continue, goto	76
Ders 4 İzleme Tablosu Tekniği ve Sayacın Konumu	81
Modül Tekrar Testi	87
Modül 4 · Döngü Kalıpları ve Diziler	90
Ders 1 Döngü Kalıpları: Biriktirici, Sayaç, En Büyük	91
Ders 2 Diziler: Tanım, İndeks ve Gezinme	98
Ders 3 Dizi Kalıpları: Toplam, Arama, Sayma	104
Ders 4 İç İç Döngü ve Şekil Basma	110
Modül Tekrar Testi	117
Modül 5 · Matrisler ve Dizgiler	120
Ders 1 İki Boyutlu Diziler: Satır ve Sütun	121
Ders 2 Matris Kalıpları: Köşegen, Satır ve Sütun	127
Ders 3 Karakterler, ASCII ve char Aritmetiği	133
Ders 4 Dizgiler: '\0', strlen ve Dizgi Dizileri	139
Modül Tekrar Testi	145
Modül 6 · Fonksiyonlar ve Özyineleme	148
Ders 1 Fonksiyonlar: Tanım, Parametre, Değer Kopyası	149
Ders 2 Özyineleme: Taban Durum ve Çağrı Zinciri	157
Ders 3 Özyinelemede Yazdırma Sırası	163
Ders 4 Özyineleme Kalıpları: İkili Çağrı ve İç İç Fonksiyon	169
Modül Tekrar Testi	175

Modül 7 · İşaretçiler	178
Ders 1 İşaretçi Nedir: Adres, & ve *	179
Ders 2 İşaretçi Aritmetiği: p + 1 Nereye Gider	185
Ders 3 İşaretçi ve Dizi: İki Yüzü Olan Aynı Şey	190
Ders 4 İşaretçiyle Değiştirme ve Karma Sorular	196
Modül Tekrar Testi	203
Modül 8 · Bellek, Yapılar ve Sınav Kampı	206
Ders 1 Bellek: Statik, Dinamik ve Tip Boyutları	207
Ders 2 Yapılar: struct, union, enum ve typedef	214
Ders 3 Bit İşlemleri ve İkili Gösterim	220
Ders 4 Sınav Kampı: Strateji ve Karma Tekrar	225
Modül Tekrar Testi	231
Modül 9 · Algoritma I — Simülasyon ve İzleme	234
Ders 1 İzleme Tablosu ve Adım Adım Simülasyon	235
Ders 2 Izgara ve Robot Simülasyonu	240
Ders 3 Durum Makineleri ve Kural Tabanlı Sistemler	245
Ders 4 Kontrol Basamağı ve Doğrulama Algoritmaları	250
Modül Tekrar Testi	255
Modül 10 · Algoritma II — Arama, Sıralama, Strateji	258
Ders 1 Arama: Doğrusal ve İkili	259
Ders 2 Sıralama Algoritmaları	265
Ders 3 Yığın (Heap) ve Öncelikli Ekleme	271
Ders 4 Yinelemeli Arama, Strateji ve Kapanış	277
Modül Tekrar Testi	284
Ek A · 2023–2026 Çıkmış Benzeri Soru Dizini	287
Ek B · Cevap Anahtarı	289
Ek C · Sınav Formatındaki Soruların Çözümleri	291
Ek D · Müfredat Tamamlayıcı	379
Ek E · C Hızlı Başvuru Kartı	382
Dizin	386

MODÜL 3 · DERS 2

for ve iç içe for



for, döngünün en derli toplu hâlidir: başlangıç, koşul ve adım tek satırda durur. Sınavda en çok bu döngü çıkar — özellikle de iç içe yazılmış hâli. Bu derste kaç tur döndüğünü saymayı ve iç içe döngülerde toplam işlem sayısını hesaplamayı öğreniyorsun.

for Döngüsünün Anatomisi

```
for (başlangic; koşul; adım) {
    gövde
}
```

Örnek:

```
for (int i = 1; i <= 3; i++) {
    printf("%d", i);
}
```

Çalışma sırası:

1. başlangic bir kez çalışır (i = 1)
2. koşul kontrol edilir (i <= 3)
3. doğruysa gövde çalışır
4. adım çalışır (i++)
5. 2. adıma dönülür

ÇIKTI 123

En önemli ayrıntı: **adım, gövdeden SONRA çalışır**. Gövde içinde i'yi kullanırken henüz artmamış değeri görürsün.

Adım Farklı Olabilir

for (i = 1; i <= 5; i += 2)	i = 1, 3, 5 (3 tur)
for (i = 10; i > 0; i -= 3)	i = 10, 7, 4, 1 (4 tur)
for (i = 1; i <= 100; i *= 2)	i = 1, 2, 4, 8, 16, 32, 64 (7 tur)

İç İçe for

Bir for'un gövdesine başka bir for konabilir:

```
for (int i = 1; i <= 3; i++) {
    for (int j = 1; j <= 2; j++) {
        printf("*");
    }
}
```

Dış döngü 3 tur, her turda iç döngü 2 tur → toplam $3 \times 2 = 6$ yıldız.



KURAL

iç içe döngülerde toplam = dış tur $\times$ iç tur.

İç Sınır Dışa Bağlıysa

Asıl ilginç durum, iç döngünün sınırı dış değişkene bağlı olduğunda:

```
for (int i = 1; i <= 3; i++) {
    for (int j = 1; j <= i; j++) {
        printf("*");
    }
}
```

Artık her turda iç döngü farklı sayıda döner:

i	iç tur	yıldız
1	1	*
2	2	**
3	3	***

Toplam: $1 + 2 + 3 = 6$ yıldız

Bu artık çarpma değil TOPLAMA problemidir: $1 + 2 + \dots + n = n(n+1)/2$

$n = 3$ için $3 \times 4 / 2 = 6 \checkmark$

Sayma Alıştırması

```
for (i = 0; i < n; i++)
    for (j = 0; j < n; j++)
        sayac++;
```

Toplam işlem: $n \times n = n^2$

```
for (i = 0; i < n; i++)
    for (j = i; j < n; j++)
        sayac++;
```

Toplam: $n + (n-1) + \dots + 1 = n(n+1)/2$

Bu iki kalıbı tanımak, "kaç işlem yapılır" sorularını saniyede çözer.

Çıktı Nerede?

printf'in hangi döngünün içinde olduğuna çok dikkat et:

```
for (i = 1; i <= 3; i++) {
    for (j = 1; j <= 3; j++) printf("*");
    printf("\n");           // ← DIŞ döngüde: her satırdan sonra bir kez
}
```

Satır sonu iç döngünün DIŞINDA olduğu için 3 satır × 3 yıldız çıkar. İçeri alsaydın 9 satır olurdu.

Sık yapılan hatalar

X for (i=1; i<=3; i++) döngüsünü 3 değil 4 tur sanmak

✓ i = 1, 2, 3 → 3 tur; i = 4 olduğunda koşul bozulur

X İç sınırı i olan döngüde toplamı $3 \times 3 = 9$ hesaplamak

✓ $1 + 2 + 3 = 6$; her turda iç döngü farklı sayıda döner

X Satır sonunu iç döngüye koymak

✓ printf("\n") dış döngüde olmalı ki her satırdan sonra bir kez çalışsın

X Gövdede i'nin artmış değerini görmek

✓ Adım gövdeden sonra çalışır; gövdede eski değer geçerlidir



HATIRLA

- for'da adım, gövdeden sonra çalışır.
- İç içe döngüde toplam tur = dış tur × iç tur.
- İç sınır dış değişkene bağlıysa toplam $1+2+\dots+n = n(n+1)/2$ olur.
- printf'in hangi döngünün içinde olduğu çıktının şeklini belirler.
- Adım i++ olmak zorunda değil: i+=2, i-=3, i*=2 de olabilir.





Sınav Formatında Sorular

Önerilen süre: 15 dakika

1

ÇIKMIŞ BENZERİ · 2026 S16

Aşağıdaki programın ekran çıktısı nedir?

```
int main() {  
    int s = 0;  
    for (int k = 20; k >= 5; k -= 4) {  
        s += k;  
    }  
    printf("%d\n", s);  
    return 0;  
}
```

A) 60

B) 56

C) 50

D) 36

E) 44

2

SINAV FORMATINDA

Aşağıdaki program ekrana kaç tane * basar?

```
int main() {  
    for (int i = 1; i <= 3; i++) {  
        for (int j = 1; j <= 3; j++) {  
            printf("*");  
        }  
    }  
    return 0;  
}
```

A) 3

B) 6

C) 12

D) 27

E) 9

3

ÇIKMIŞ BENZERİ · 2026 S23

Aşağıdaki programın ekran çıktısı nedir?

```
int main() {  
    int adet = 0;  
    for (int i = 4; i >= 1; i--) {  
        for (int j = i; j <= 4; j++) {  
            adet++;  
        }  
    }  
    printf("%d\n", adet);  
    return 0;  
}
```

A) 16

B) 10

C) 4

D) 6

E) 12

4

SINAV FORMATINDA

Aşağıdaki kodun çıktısının 9 olması için boş bırakılan yere hangi ifade gelmelidir?

```
int main() {  
    int t = 0;  
    for (int i = 1; i <= 5; _____) {  
        t += i;  
    }  
    printf("%d", t);  
    return 0;  
}
```

A) i++

B) i += 2

C) i += 3

D) i--

E) i *= 2

5

SINAV FORMATINDA

Aşağıdaki program çalıştığında ekranda nasıl bir görüntü oluşur?

```
int main() {  
    for (int i = 1; i <= 3; i++) {  
        for (int j = 1; j <= 3; j++)  
            printf("*");  
        printf("\n");  
    }  
    return 0;  
}
```

A) 3 satır, her satırda 3 yıldız

B) 1 satır, 9 yıldız yan yana

C) 9 satır, her satırda 1 yıldız

D) 3 satır, sırayla 1, 2 ve 3 yıldız

E) Hiçbir şey yazılmaz

Adım adım çözümler: Ek C

5

SINAV FORMATINDA

D) 10

İzleme tablosu:

tur	n	n%10	t	n sonrası
1	1234	4	4	123
2	123	3	7	12
3	12	2	9	1
4	1	1	10	0
—	0	—	—	döngü biter

ÇIKTI 10

Bu kod ne yapıyor? Sayının BASAMAKLARINI TOPLUYOR: $1 + 2 + 3 + 4 = 10$

İki işlecin birlikte çalışması bu kalıbın anahtarı:

- $n \% 10 \rightarrow$ son basamağı verir

$n /= 10$ son basamağı siler

Bu ikili, M4'te sürekli karşına çıkacak. Bir sayıyı basamaklarına ayırmanın standart yoludur.

Çeldiriciler:

- 4 $\rightarrow$ sadece ilk turu hesaplamak
- 4321 $\rightarrow$ sayıyı ters çevirdiğini sanmak (o farklı bir kalıptır)

Doğru cevap: D) 10

Modül 3 · Ders 2 — for ve iç içe for

1

ÇIKMIŞ BENZERİ · 2026 S16

B) 56

k 20'den başlıyor, her turda 4 AZALIYOR, $k \geq 5$ olduğu sürece dönüyor.

tur	k	$k \geq 5?$	s
1	20	evet	20
2	16	evet	36
3	12	evet	48
4	8	evet	56
—	4	hayır	döngü biter

$$s = 20 + 16 + 12 + 8 = 56$$

Dikkat: $4 < 5$ olduğu için son değer (4) toplama GİRMEZ.

Çeldiriciler:

- A) 60 $k = 4$ 'ü de toplamak
- D) 36 iki turda durmak

Doğru cevap: B) 56

Kaynak notu: Bu soru, TÜBİTAK 2026 1. aşama Ortaokul Bilgisayar sınavının 16. sorusundaki kalıbı ölçmek için 1e1kod tarafından yazılmış özgün bir sorudur. Orijinal soru TÜBİTAK Bilim Olimpiyatları sitesinde yayımlanmıştır.

2

SINAV FORMATINDA

E) 9

Her iki döngünün de sınırı sabit:

Dış döngü: $i = 1, 2, 3$

3 tur

İç döngü: her dış turda $j = 1, 2, 3$

3 tur

Toplam = $3 \times 3 = 9$ yıldız**KURAL**sabit sınırlı iç içe döngüde toplam = dış tur $\times$ iç tur.

Çeldiriciler:

- 6 $\rightarrow$ iç sınırın dışı bağlı olduğunu sanmak (o zaman $1+2+3 = 6$ olurdu)
- 27 $\rightarrow$ üç döngü varmış gibi 3^3 hesaplamak

Bir sonraki soruda iç sınırın dışı bağlı hâlini göreceksin — çıktı farklı olacak.

Doğru cevap: E) 9

3

ÇIKMIŞ BENZERİ · 2026 S23

B) 10

İç döngü $j = i$ 'den 4'e kadar dönüyor — başlangıcı dış döngüye bağlı, her turda DEĞİŞİYOR.

i	j değerleri	iç tur
4	4	1
3	3, 4	2
2	2, 3, 4	3
1	1, 2, 3, 4	4

Toplam = $1 + 2 + 3 + 4 = 10$ Sınır değişken olduğu için ÇARPMA ($4 \times 4 = 16$) değil TOPLAMA yapılır. Kısa yol: $1 + 2 + \dots + n = n(n+1)/2 \rightarrow 4 \times 5 / 2 = 10$.

Çeldiriciler:

A) 16

iç döngünün her seferinde 4 kez döndüğünü sanmak

C) 4

yalnız dış döngüyü saymak

Doğru cevap: B) 10

Kaynak notu: Bu soru, TÜBİTAK 2026 1. aşama Ortaokul Bilgisayar sınavının 23. sorusundaki kalıbı ölçmek için 1e1kod tarafından yazılmış özgün bir sorudur. Orijinal soru TÜBİTAK Bilim Olimpiyatları sitesinde yayımlanmıştır.

4

SINAV FORMATINDA

B) $i += 2$

Her şıkkı deneyelim:

A) $i++$ $i = 1, 2, 3, 4, 5 \rightarrow t = 15$ B) $i += 2$ $i = 1, 3, 5 \rightarrow t = 1+3+5 = 9 \checkmark$ C) $i += 3$ $i = 1, 4 \rightarrow t = 5$ D) $i--$ $i = 1, 0, -1, \dots$ koşul hep doğru $\rightarrow$ SONSUZ DÖNGÜE) $i *= 2$ $i = 1, 2, 4 \rightarrow t = 7$

D şıkkı önemli bir uyarı: adım yanlış yöne giderse koşul hiç bozulmaz ve program durmaz. Sınavda "sonsuz döngü" şıkkı bazen doğru cevaptır.

Doğru cevap: B) $i += 2$

MODÜL 7 · DERS 1

İşaretçi Nedir: Adres, & ve *



Şimdiye kadar değişkenin İÇİNDEKİ değeri kullandın. İşaretçi ise değişkenin NEREDE durduğunu tutar. Bu iki soruyu birbirinden ayırabilen öğrenci sınavın işaretçi sorularını kaybetmez: "içinde ne var?" ve "nerede duruyor?"

Her Değişkenin Bir Adresi Var

Bellek, numaralı kutulardan oluşur. Bir değişken tanımladığında C ona bir kutu ayırır ve o kutunun bir numarası (adresi) olur.

```
int a = 42;
```

adres	1000	1004	1008
içerik	42	?	?

a'nın DEĞERİ 42, ADRESİ 1000.

& — Adresini Al

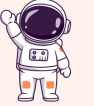
```
printf("%d", a);           // 42 → değeri
printf("%p", (void *)&a); // 1000 → adresi
```

& işareti "bunun adresi" demektir. Türkçesi: "a'nın nerede durduğunu ver."



TUZAK

Adres **%p** ile basılır ve adres **(void *)** ile verilir. **%d** ile adres basmak YANLIŞTIR: tanımsız davranıştır ve 64 bit sistemde adresin yalnızca bir kısmı çıkar. Gerçek adres 0x7ffd5e8c3a4c gibi uzun bir sayıdır ve programı her çalıştırdığında değişir. Bu yüzden sınav soruları adresi 1000 gibi uydurma bir sayıyla verir; buradaki 1000 de öyle.



İşaretçi Tanımlama

```
int *p;           // p, bir int'in adresini tutabilen değişken
p = &a;           // p artık a'yı gösteriyor
```

Yıldız tanımda "bu bir işaretçidir" anlamına gelir. Tipin de önemi vardır: **int *p** yalnızca int adresi tutar, **double *q** yalnızca double adresi.

* — Gösterdiği Yerdeki Değer

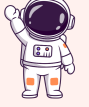
```
printf("%p", (void *)p); // 1000 → p'nin kendi değeri, yani adres
printf("%d", *p);       // 42  → p'nin GÖSTERDİĞİ değer
```



TUZAK

Yıldızın iki farklı işi var ve bu sınavda tuzak olur:

- **Tanımda** `int *p;` → "p bir işaretçidir"
- **Kullanımda** `*p` → "p'nin gösterdiği yerdeki değer"



Aynı sembol, iki farklı anlam. Hangi bağlamda olduğuna bak.

Gösterdiği Yere Yazmak

```
*p = 99; // a'nın kutusuna 99
yazar
printf("%d", a); // 99
```

p'nin kendisine dokunmadık; p hâlâ aynı kutuyu gösteriyor. Ama o kutunun İÇİNİ değiştirdik.

Bu, işaretçilerin en önemli gücüdür: **bir fonksiyona adres verirsen, o fonksiyon senin değişkenini gerçekten değiştirebilir.** (M7-L4'te, İşaretçiyle Değiştirme dersinde buna döneceğiz.)

İki Farklı Atama

Dikkat, bu ikisi apayrı şeydir:

```
p = &b; // p artık BAŞKA bir kutuyu gösteriyor
*p = 7; // p'nin gösterdiği kutunun İÇİNE 7 yazıldı
```

Birincisi işaretçiye hareket ettirir, ikincisi hedefi değiştirir. Sınav soruları bu ikisini arka arkaya kullanıp izlemeni zorlaştırır.

İzleme Yöntemi

Her işaretçi sorusunda kâğıda İKİ şey çiz:

1. Kutular ve içindekiler
2. Okla: hangi işaretçi hangi kutuyu gösteriyor

kutu	a	b
değer	42	7

p → a

Her satırda oku ya da kutunun içeri güncelle. Bu iki katmanlı çizim olmadan işaretçi soruları izlenemez.

NULL — Hiçbir Yeri Göstermemek

```
int *p = NULL;
```

NULL, "şu an hiçbir yeri göstermiyorum" demektir. NULL bir işaretçiyi yıldızla açmak (*p) programı çökertir.

İşaretçiye değer atamadan kullanmak da tehlikelidir:

```
int *p;      // nereyi gösteriyor? BELİRSİZ
*p = 5;      // ✗ rastgele bir belleğe yazar
```

Sınav Refleksi

1. p mi yazıyor *p mi? Biri adres, diğeri değer.
2. p = ... mı *p = ... mı? Biri oku taşır, diğeri kutunun içeri değiştirir.
3. & gördüysen: adres alınıyor.
4. Kâğıda kutuları ve okları çiz — kafadan izleme.

Sık yapılan hatalar

✗ p ile *p'yi karıştırmak

✓ p adrestir, *p o adresteki değerdir

✗ p = 7; yazarak değeri değiştirdiğini sanmak

✓ Bu işaretçiyi 7 numaralı adrese taşır; değer için *p = 7 gerekir

✗ Tanımdaki yıldız da 'değeri al' diye okumak

✓ Tanımda yıldız yalnızca 'bu bir işaretçidir' demektir

✗ İşaretçiye adres atamadan *p kullanmak

✓ Rastgele belleğe yazılır; önce bir adres atanmalıdır

✗ Soruyu kafadan izlemeye çalışmak

✓ Kutuları ve okları kâğıda çiz; her satırda güncelle



HATIRLA

- Her değişkenin bir değeri ve bir adresi vardır; & adresi verir.
- int *p tanımı p'nin bir int adresi tutacağını söyler.
- p işaretçinin kendi değeridir (adres), *p gösterdiği yerdeki değerdir.
- p = &b işaretçiyi başka kutuya taşır; *p = 7 gösterdiği kutunun içeri değiştirir.
- Yıldız tanımda 'işaretçidir', kullanımda 'gösterdiği değer' anlamına gelir.
- NULL hiçbir yeri göstermez; NULL veya atanmamış işaretçiyi açmak programı çökertir.





Sınav Formatında Sorular

Önerilen süre: 15 dakika

1

SINAV FORMATINDA

Aşağıdaki programın ekran çıktısı nedir?

```
int main() {  
    int a = 25;  
    int *p = &a;  
    printf("%d", *p);  
    return 0;  
}
```

A) 25

B) a'nın adresi

C) 0

D) 1

E) Kod hata verir

2

SINAV FORMATINDA

Aşağıdaki programın ekran çıktısı nedir?

```
int main() {  
    int a = 8;  
    int *p = &a;  
    *p = *p * 2;  
    a = a + 1;  
    printf("%d %d", a, *p);  
    return 0;  
}
```

A) 8 16

B) 16 17

C) 17 8

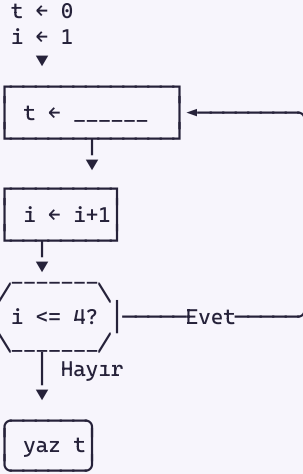
D) 17 17

E) 16 16

4

SINAV FORMATINDA

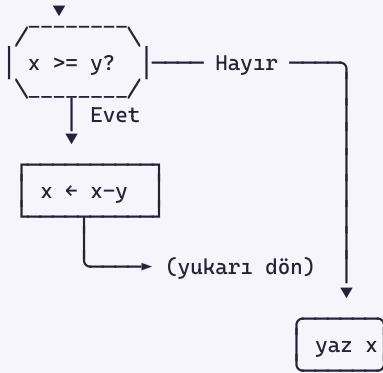
Aşağıdaki akış diyagramının çıktısının 10 olması için boş bırakılan kutuya ne yazılmalıdır?

A) $t+i$ B) $t+1$ C) $t*i$ D) i E) $t+i*2$

5

SINAV FORMATINDA

Aşağıdaki akış diyagramı $x = 17$ ve $y = 5$ için çalıştırıldığında ekrana ne yazılır?



A) 3

B) 5

C) 12

D) 2

E) 0

Adım adım çözümler: Ek C

Bu soruların çözümü bu örnekte yok. Döngüyü adım adım izleme tablosuyla gösteren çözümler kitabın 297. sayfasında.

[Kitabı incele →](#)

ÖRNEK BURADA BİTTİ

Kitabın geri kalanında 385 sayfa daha var

- ✓ 40 dersin tamamı — temellerden işaretçilere, özyinelemeden algoritma sorularına
- ✓ 211 soru, 5 şıklı, her biri adım adım çözümlü
- ✓ 2023-2026 sınav kalıplarıyla 45 çıkmış benzeri soru
- ✓ Her modül sonunda tekrar testi, cevap anahtarı ve dizin
- ✓ Sınav kampı: süre yönetimi, boş bırakma kararı, karma tekrar



s. 43'teki soruyu çözebildin mi? Akış diyagramındaki döngü kaç kez döner, x hangi değerde durur? Adım adım izleme tablosuyla çözümü kitabın **297. sayfasında**.

Kargo dahil

Türkiye'nin her yerine

15 gün içinde

kargoda, takip no e-postana

14 gün iade

gerekçe gerekmez, kargo bizden

Güvenli ödeme

PayTR · faturalı satış

Kitabı sipariş et →

Güncel lansman fiyatı ve sipariş formu sayfada

1e1kod.org/olimpiyat-kitabi · Sorun olursa WhatsApp: 0232 335 35 17

Bu kitap TÜBİTAK'ın resmî yayını değildir; 1e1kod tarafından hazırlanmış bir sınava hazırlık kaynağıdır.