

BU SAYFA BOŐ BIRAKILMIŐTIR.

SORU 1

Satırları a,b,c,d,e harfleriyle, sütunları ise 1,2,3,4,5 rakamlarıyla adreslenmiş olan 5 satır ve 5 sütuna bölünmüş tahta üzerinde oynanan bir oyunda oyuncunun piyonu tahtanın bir hücresine konulmakta ve ardından zar atılarak piyon ilerletilmektedir. Atılan zarda gelen her değer farklı bir anlamı vardır. Bunlar aşağıda verilmiştir. Piyon tahtanın kenarından çıkarsa karşı kenardan tekrar girer. Örneğin e satırında iken 2 kare aşağı gitmesi gereken bir piyon tahtanın dışına çıkmış olduğundan tahtanın üst kısmından ve aynı sütundan tekrar girer ve b satırına ilerler. Bir oyuncu sırasıyla 3,2,6,5,2,3,4,1,4,5,3,1 zarlarını atarak c4 karesinde ve yukarı yönde oyunu bitirmiştir. Bu durumda bu oyuncu oyuna hangi karede ve hangi yönde başlamıştır?

	1	2	3	4	5
a					
b					
c				↑	
d					
e					

Zarların karşılıkları:
1: 2 kare ilerle,
2: saat yönünde 90 derece dön,
3: 1 kare geri git
4: saatin tersi yönünde 90 derece dön,
5: bekle
6: 1 kare ilerle

- A) d2, sağ
B) d1, yukarı
C) b5, sol
D) c3, aşağı
E) e1, yukarı

ÇÖZÜM

c4 ve yukarı pozisyonundan geriye doğru atılan zarlar takip edildiğinde (yönleri kuzey, güney, doğu, batı olarak kodlarsak):

(d1,k)>3>(e1,k)>2>(e1,d)>6>(e2,d)>5>(e2,d)>2>(e2,g)>3>(d2,g)>4>(d2,d)>1>(d4,d)>4>(d4,k)
>5>(d
4,k)>3>(e4,k)>1>(c4,k)

oyuncu d1 hücresinde ve yukarı yönde oyuna başlamıştır.

CEVAP: B

SORU 2

Bir öğrenci a,b,c,d,e,f,i harflerini kullanarak yazabileceği tüm 3 harfli kelimelerin sayısını merak etmektedir. Bu kelimeleri türetirken öğrencinin takip etmesi gereken kurallar şunlardır:

1. kelimedede en az 1 en fazla 2 sesli harf olabilir
2. 1 sesli harf var ise, sesli harf ortadadır
3. 2 sesli harf var ise sesli harfler komşu değildir.

Bu kurallar dahilinde öğrencinin yazabileceği kaç farklı kelime vardır?

- A) 48
- B) 64
- C) 36
- D) 27
- E) 84

ÇÖZÜM

Verilen iki duruma göre incelersek:

1. 1 sesli harf ile yazılabilen $4 \times 3 \times 4 = 48$ kelime vardır (sessiz x sesli x sessiz)
 2. 2 sesli harf ile yazılabilen $3 \times 4 \times 3 = 36$ kelime vardır (sesli x sessiz x sesli)
- Toplamda 84 kelime yazılabilir.

CEVAP: E

SORU 3

Bir ağacın gövdesinden 3 adet dal çıkmaktadır. Her bir daldan tekrar 3er adet dal çıkmaktadır. Sonra bu dallardan da 3er adet dal çıkmaktadır. Her dal, kendisinin çıktığı dalın kütlesinin yarısı kadar kütleyle sahiptir. Gövdeden çıkan dalların herbiri de gövdenin yarısı kadar kütleyle sahiptir. Ağacın gövdesi 1 birim kütleyle sahip ise, ağacın toplamı kaç birim kütleyle sahiptir?

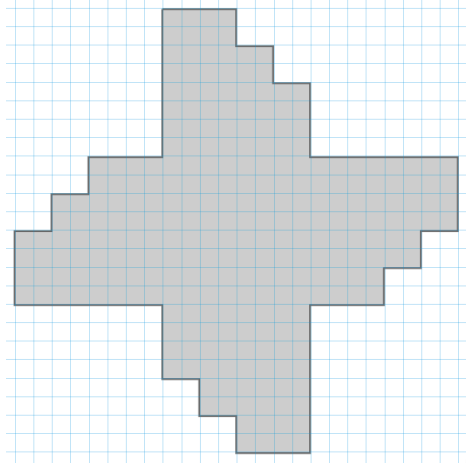
- A) $65/8$
- B) 4
- C) 15
- D) $57/8$
- E) $19/4$

ÇÖZÜM

Ağacın gövdesi 1 birim olduğuna göre, ilk seviyede çıkan dallar $1/2$, sonraki seviyede $1/4$, sonraki seviyede ise her bir dal $1/8$ ağırlığında olacaktır. İlk seviyede 3 dal olduğundan $3 \times 1/2 = 3/2$, ikinci seviyede $3 \times 3 \times 1/4 = 9/4$ ve son seviyede $3 \times 3 \times 3 \times 1/8 = 27/8$ birim kütle olacaktır. Hepsini topladığımızda $65/8$ sonucuna ulaşırız.

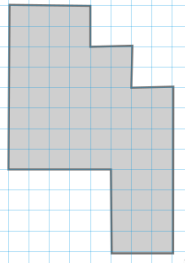
CEVAP: A

SORU 4

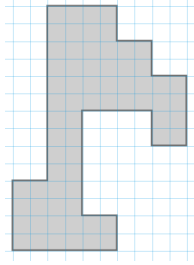


Şekildeki figürün dört eş parçaya bölünmesi sonucunda aşağıdaki hangi figür elde edilebilir?

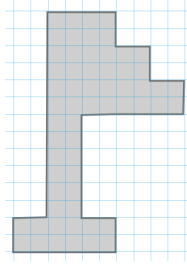
A)



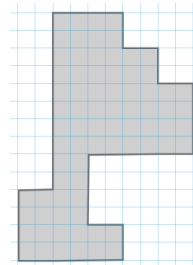
B)



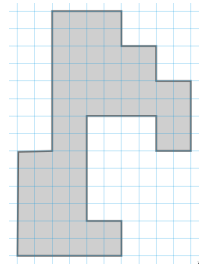
C)



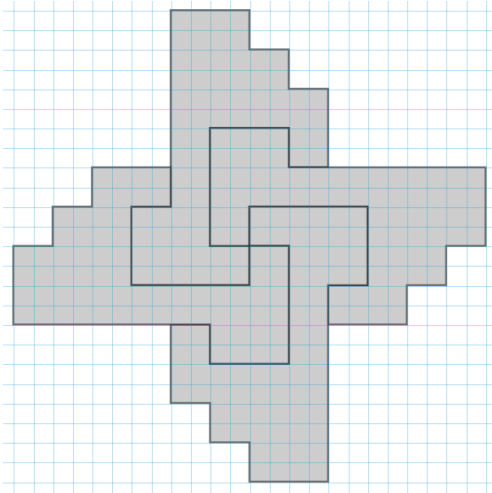
D)



E)



ÇÖZÜM



CEVAP: B

SORU 5

4 kardeş eşit yaş aralıklarıyla doğmuşlardır ve toplam yaşları 68'dir. En büyük kardeş en fazla kaç yaşında olabilir? (tüm yaşlar tam sayı kabul edilir)

- A) 20
- B) 23
- C) 26
- D) 32
- E) 35

ÇÖZÜM

Kardeşlerin yaşlarını x , $x+a$, $x+2a$, $x+3a$ olarak kodlarsak, $4x+6a = 68$ olur. Bu denklemin tam sayı çözümleri $(14,2)$, $(11,4)$, $(8,6)$, $(5,8)$ ve $(2,10)$ şeklinde bulunur. Bu durumda en büyük kardeş en fazla $(2,10)$ seçeneğine uygun olarak $2+3 \times 10 = 32$ yaşında olabilir.

CEVAP: D

[Soru 6-7 için açıklama]

Başlangıçta n adet tam sayı bir kuyrukta soldan sağa dizili olsun. Kuyruğun uzunluğu 1 olana kadar aşağıdaki algoritma uygulanıyor.

1. Kuyruğun önündeki iki sayıyı sırayla x ve y olarak al ve kuyruktan çıkar.
2. Eğer $x < y$ ise, $y - x$ sayısını kuyruğun sonuna ekle.
3. Aksi halde, $x + y$ sayısını kuyruğun sonuna ekle.

Örnek olarak başlangıç kuyruğu $\{3, 1, 4\}$ ise önce 3 ve 1 alınır, bunların yerine 4 eklenir ve kuyruk $\{4, 4\}$ olur. Sonraki adımda geriye 8 kalır.

Aşağıdaki iki soruyu bu kurallara göre cevaplayınız.

SORU 6

A: $\{10, 2, 9, 1, 8, 3, 7, 4, 6\}$ olduğunda algoritma sonlandığında kuyrukta kalan sayı ne olur?

- A) 12
- B) 14
- C) 16
- D) 18
- E) 20

ÇÖZÜM

Başlangıç:

$\{10, 2, 9, 1, 8, 3, 7, 4, 6\}$

10 ve 2 alınır. $10 < 2$ yanlış, $10 + 2 = 12$ eklenir:

$\{9, 1, 8, 3, 7, 4, 6, 12\}$

9 ve 1 alınır, 10 eklenir:

$\{8, 3, 7, 4, 6, 12, 10\}$

8 ve 3 alınır, 11 eklenir:

$\{7, 4, 6, 12, 10, 11\}$

7 ve 4 alınır, 11 eklenir:

$\{6, 12, 10, 11, 11\}$

6 ve 12 alınır. $6 < 12$ olduğu için $12 - 6 = 6$ eklenir:

$\{10, 11, 11, 6\}$

10 ve 11 alınır, 1 eklenir:

$\{11, 6, 1\}$

11 ve 6 alınır, 17 eklenir:

$\{1, 17\}$

1 ve 17 alınır, 16 eklenir:

$\{16\}$

Sonuç 16.

CEVAP: C

SORU 7

A: {5, 12, 4, 9, 2, 11, 3, 8} olduğunda algoritma sonlandığında kuyrukta kalan sayı ne olur?

- A) 0
- B) 2
- C) 4
- D) 6
- E) 8

ÇÖZÜM

Başlangıç:

{5, 12, 4, 9, 2, 11, 3, 8}

5 ve 12 alınır. $5 < 12$ olduğu için $12 - 5 = 7$ eklenir:

{4, 9, 2, 11, 3, 8, 7}

4 ve 9 alınır, 5 eklenir:

{2, 11, 3, 8, 7, 5}

2 ve 11 alınır, 9 eklenir:

{3, 8, 7, 5, 9}

3 ve 8 alınır, 5 eklenir:

{7, 5, 9, 5}

7 ve 5 alınır. $7 < 5$ yanlış, $7 + 5 = 12$ eklenir:

{9, 5, 12}

9 ve 5 alınır, 14 eklenir:

{12, 14}

12 ve 14 alınır, 2 eklenir:

{2}

Sonuç 2.

CEVAP: B

SORU 8

Başlangıçta sadece "A" harfinden oluşan bir diziye her adımda aşağıdaki kurallar **eşzamanlı olarak** uygulanmaktadır.

1. Dizideki her "A" harfini "AB" harf grubuna dönüştür.
2. Dizideki her "B" harfini "CA" harf grubuna dönüştür.
3. Dizideki her "C" harfini "B" harfine dönüştür.

Örnek olarak 1. adımın sonunda dizi "AB" olurken, 2. adımın sonunda dizi "ABCA" şekline gelmektedir.

Bu algoritma kurallarına göre çalıştırıldığında, **5. adımın** tamamlanmasıyla elde edilen yeni dizilimde toplam kaç tane "B" harfi bulunur?

- A) 4
- B) 6
- C) 9
- D) 10
- E) 23

ÇÖZÜM

Bu tür sorularda uzun dizileri simüle etmek hata payını artıracığından, algoritma matematiksel bir durum geçiş (state transition) modeline dönüştürülmelidir. Harflerin diziliş sırası miktar üzerinde etkili olmadığından sadece sayıları dikkate alınır.

A_n , B_n , C_n değerleri n . adımdaki harf sayılarını göstermek üzere, bir sonraki adıma geçiş kuralları şu formüllerle ifade edilir:

- $A_{n+1} = A_n + B_n$ (Her A ve B harfi yeni bir A üretir)
- $B_{n+1} = A_n + C_n$ (Her A ve C harfi yeni bir B üretir)
- $C_{n+1} = B_n$ (Mevcut B harfleri C harfine dönüşür)

Başlangıç (Adım 0): $A=1, B=0, C=0$

- **1. Adım:** $A=(1+0)=1, B=1, C=0$
- **2. Adım:** $A=(1+1)=2, B=1, C=1$
- **3. Adım:** $A=(2+1)=3, B=3, C=1$
- **4. Adım:** $A=(3+3)=6, B=4, C=3$
- **5. Adım:** $A=(6+4)=10, B=9, C=4$

Görüldüğü üzere **5. adım** sonunda toplam "B" harfi sayısı **9** olur. Seçeneklerdeki 4 sayısı 'C' harfini, 10 sayısı 'A' harfini ve 23 sayısı ise toplam karakter uzunluğunu ($10+9+4$) temsil eden çeldiricilerdir.

CEVAP: C

[Soru 9-10 için açıklama]

Bir ağ sisteminde, gelen görevleri düzenlemekle sorumlu bir Görev Dağıtıcı ve bu görevleri işleyen **5** adet sunucu (**S1, S2, S3, S4, S5**) bulunmaktadır. Başlangıçta tüm sunucuların yükü **0**'dir.

Sisteme sırayla numaralandırılmış görevler gelmektedir. Her görev aşağıdaki kurallara göre dağıtılır:

- Eğer gelen görevin numarası **3**'ün katı ise, görev yalnızca **S3, S4, S5** sunucularından birine atanabilir.
- Aksi halde **5** sunucunun tamamı uygundur.
- Görev her zaman, uygun olan sunucular arasında o anki yükü en az olan sunucuya atanır.
- Eğer en az yüke sahip birden fazla uygun sunucu varsa, numarası en küçük olan sunucu seçilir.
- Atanan görevin numarası çift ise seçilen sunucunun yükü **2** artar.
- Atanan görevin numarası tek ise seçilen sunucunun yükü **1** artar.

Aşağıdaki iki soruyu bu kurallara göre cevaplayınız.

SORU 9

Sisteme sırasıyla numaraları **1, 2, 3, 4, 5, 6, 7** olan **7** görev gelmektedir.

Tüm görevler (1'den 7'ye kadar) kurallara uygun biçimde dağıtıldıktan sonra, sistemde **en yüksek yüke** sahip olan sunucu veya sunucular aşağıdakilerden hangisidir?

- A) Yalnızca **S1**
- B) Yalnızca **S3**
- C) Yalnızca **S4**
- D) **S1** ve **S3**
- E) **S3** ve **S5**

ÇÖZÜM

Bu soru, "Görev Dağıtıcı" algoritmasının **7** görevlik bir süreç boyunca koşullu durum takibi ve eşitlik bozma kuralları çerçevesinde titizlikle simüle edilmesini gerektirmektedir.

Algoritmanın temel çalışma prensiplerini şu şekilde özetleyebiliriz:

- Görev numarası 3'ün katı ise sadece S3, S4, S5 sunucuları; aksi halde tüm sunucular uygundur.
- Görev, uygun sunucular arasından mevcut yükü en az olana atanır.
- Yüklerin eşitliği durumunda, numarası (indeksi) en küçük olan sunucu tercih edilir.
- Çift numaralı görevler yükü +2, tek numaralı görevler ise +1 birim artırır.

Başlangıç: {S1:0, S2:0, S3:0, S4:0, S5:0}

- **Görev 1:** (Tek, 3'ün katı değil). Tüm sunucular uygun olduğundan en küçük indeksli S1'e atanır. (S1 yükü = 1)
Durum: [1, 0, 0, 0, 0]
- **Görev 2:** (Çift, 3'ün katı değil). Uygun sunucular arasından en az yüke sahip olan S2'ye atanır. (S2 yükü = 2)
Durum: [1, 2, 0, 0, 0]
- **Görev 3:** (Tek, 3'ün katı). Sadece {S3, S4, S5} uygun olup en küçük indeksli S3'e atanır. (S3 yükü = 1)
Durum: [1, 2, 1, 0, 0]
- **Görev 4:** (Çift, 3'ün katı değil). Tüm sunucular uygun olduğundan en az yüklü S4'e atanır. (S4 yükü = 2)
Durum: [1, 2, 1, 2, 0]
- **Görev 5:** (Tek, 3'ün katı değil). En az yüke sahip olan S5'e atanır. (S5 yükü = 1)
Durum: [1, 2, 1, 2, 1]
- **Görev 6:** (Çift, 3'ün katı). Sadece {S3, S4, S5} uygun olup en az yüklü S3 ve S5 arasından en küçük indeksli S3'e atanır. (S3 yükü: 1+2 = 3)
Durum: [1, 2, 3, 2, 1]
- **Görev 7:** (Tek, 3'ün katı değil). Tüm sunucular uygun olup en az yüklü S1 ve S5 arasından S1 seçilir. (S1 yükü: 1+1 = 2)
Son Durum: [2, 2, 3, 2, 1]

Algoritma sonlandığında en yüksek yük değeri 3 olup bu yüke sahip tek sunucu S3 sunucusudur. D seçeneği, 6. adımda kısıt kuralını gözden kaçıranlar için bir çeldiricidir.

CEVAP: B

SORU 10

Sisteme sırasıyla numaraları **1, 2, 3, 4, 5, 6, 7, 8, 9** olan 9 görev gelmektedir.

Tüm görevler (1'den 9'a kadar) kurallara uygun biçimde dağıtıldıktan sonra, sistemde **en yüksek yüke** sahip olan sunucu veya sunucular aşağıdakilerden hangisidir?

- A) Yalnızca S3
- B) S3 ve S4
- C) S3, S4 ve S5
- D) S1, S3 ve S5
- E) S1, S2, S3, S4 ve S5

ÇÖZÜM

Bu soru, "Görev Dağıtıcı" algoritmasının 9 görevlik bir süreç boyunca koşullu durum takibi (state tracking) ve eşitlik bozma (tie-breaker) kuralları çerçevesinde titizlikle simüle edilmesini gerektirmektedir.

Algoritmanın temel çalışma prensiplerini şu şekilde özetleyebiliriz:

- Görev numarası **3**'ün katı ise sadece **S3, S4, S5** sunucuları; aksi halde tüm sunucular uygundur.
- Görev, uygun sunucular arasından mevcut yükü en az olana atanır.
- Yüklerin eşitliği durumunda, numarası (indeksi) en küçük olan sunucu tercih edilir.
- Çift numaralı görevler yükü **+2**, tek numaralı görevler ise **+1** birim artırır.

Başlangıç: {S1:0, S2:0, S3:0, S4:0, S5:0}

- **Görev 1:** (Tek, 3'ün katı değil). Tüm sunucular uygun, en küçük indeksli **S1** seçilir.
Durum: [1, 0, 0, 0, 0]
- **Görev 2:** (Çift, 3'ün katı değil). En az yüklü (0) sunucular içinden **S2** seçilir.
Durum: [1, 2, 0, 0, 0]
- **Görev 3:** (Tek, 3'ün katı). Sadece {S3, S4, S5} uygun, en küçük indeksli **S3** seçilir.
Durum: [1, 2, 1, 0, 0]
- **Görev 4:** (Çift, 3'ün katı değil). En az yüklü (0) **S4** sunucusuna atanır.
Durum: [1, 2, 1, 2, 0]
- **Görev 5:** (Tek, 3'ün katı değil). En az yüke (0) sahip olan **S5** seçilir.
Durum: [1, 2, 1, 2, 1]
- **Görev 6:** (Çift, 3'ün katı). {S3, S4, S5} arasından en az yüklü S3 ve S5 (yük 1) içinden **S3** seçilir.
Durum: [1, 2, 3, 2, 1]

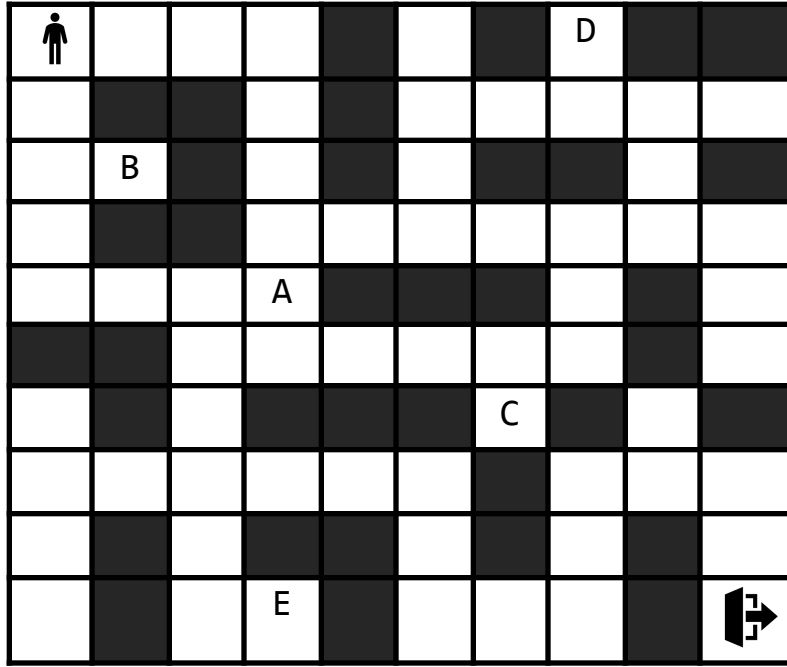
- **Görev 7:** (Tek, 3'ün katı değil). En az yüklü S1 ve S5 (yük 1) arasından **S1** tercih edilir.
Durum: [2, 2, 3, 2, 1]
- **Görev 8:** (Çift, 3'ün katı değil). En az yüke (1) sahip tek sunucu olan **S5** seçilir.
Durum: [2, 2, 3, 2, 3]
- **Görev 9:** (Tek, 3'ün katı). {S3, S4, S5} içinden en az yüke (2) sahip **S4** sunucusu seçilir.

Son Durum: [2, 2, 3, 3, 3]

Simülasyon neticesinde sunucuların nihai yükleri **S1=2, S2=2, S3=3, S4=3, S5=3** olarak belirlenmiştir.

En yüksek yük değeri olan **3** birime **S3, S4 ve S5** sunucuları birlikte ulaşmıştır.

CEVAP: C

SORU 11

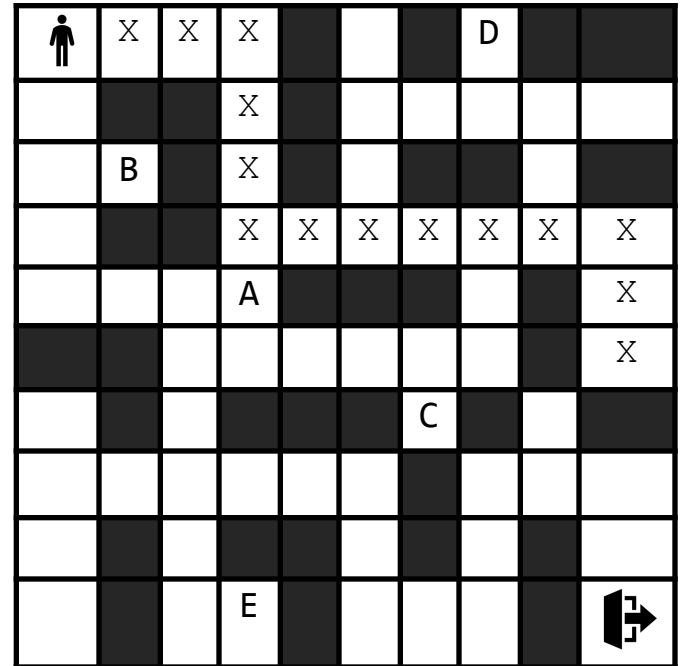
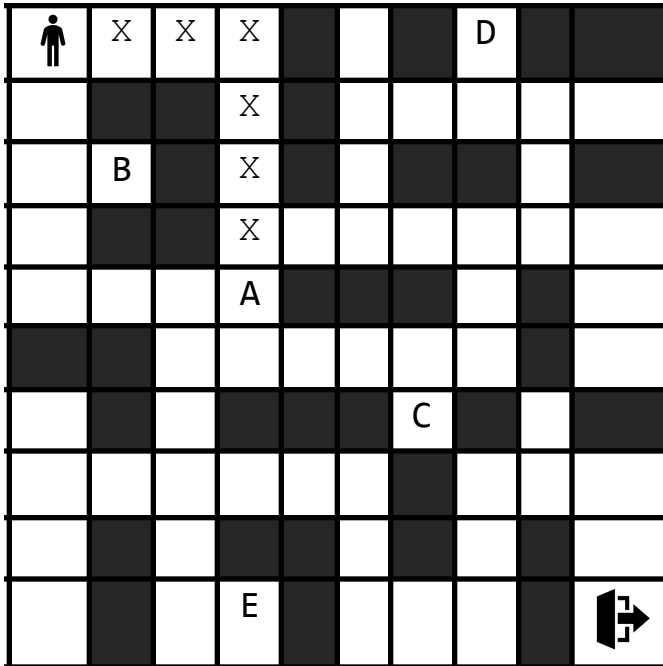
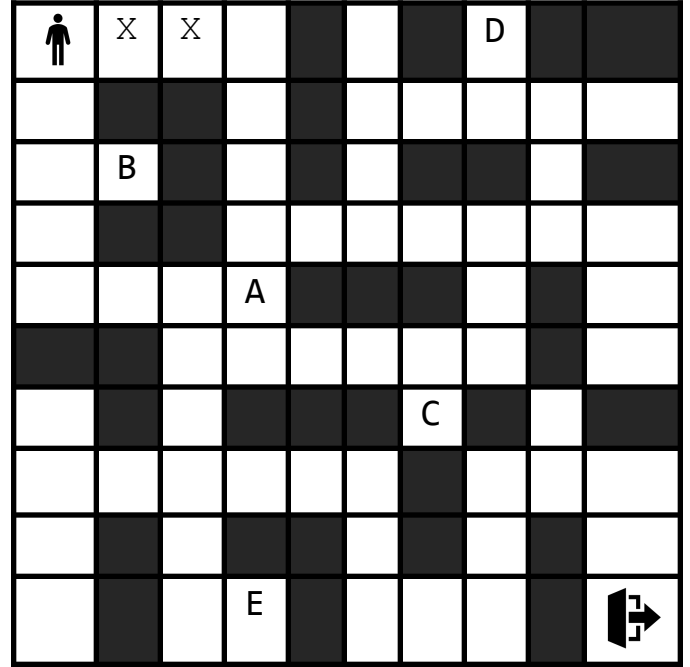
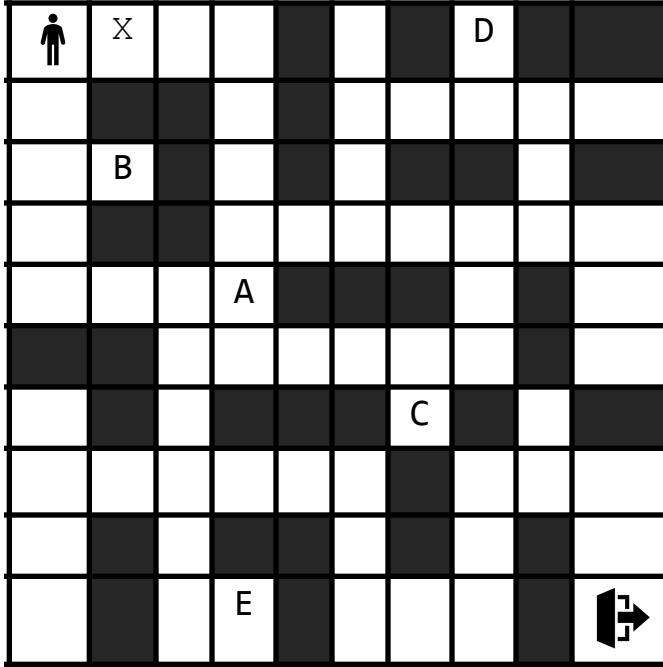
Can, bir labirentin çıkışını bulmak amacıyla bir kural dizisi ile bir arama stratejisi izlemektedir. Can'ın hareket kuralları aşağıda verilmiştir:

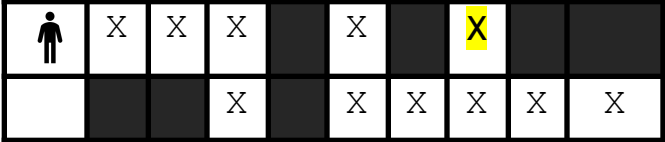
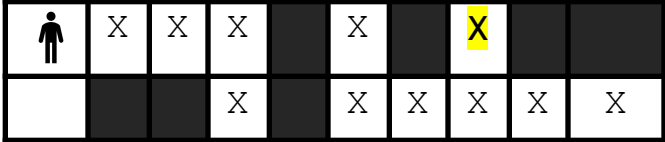
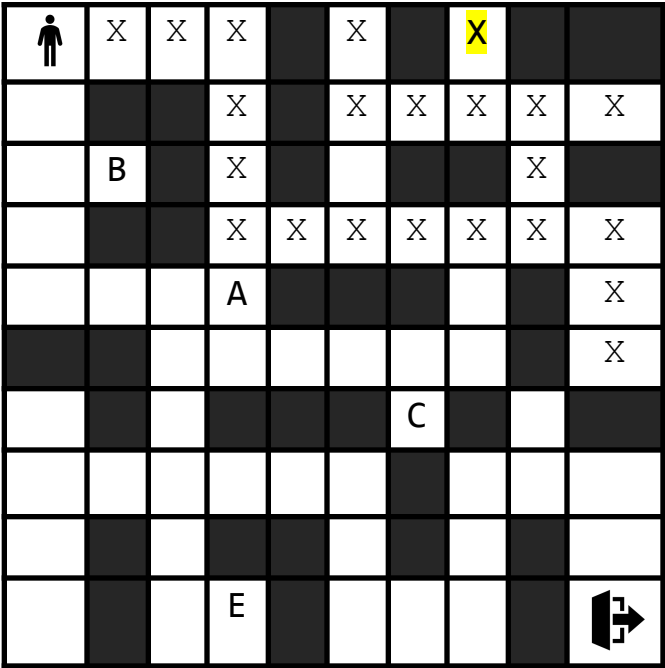
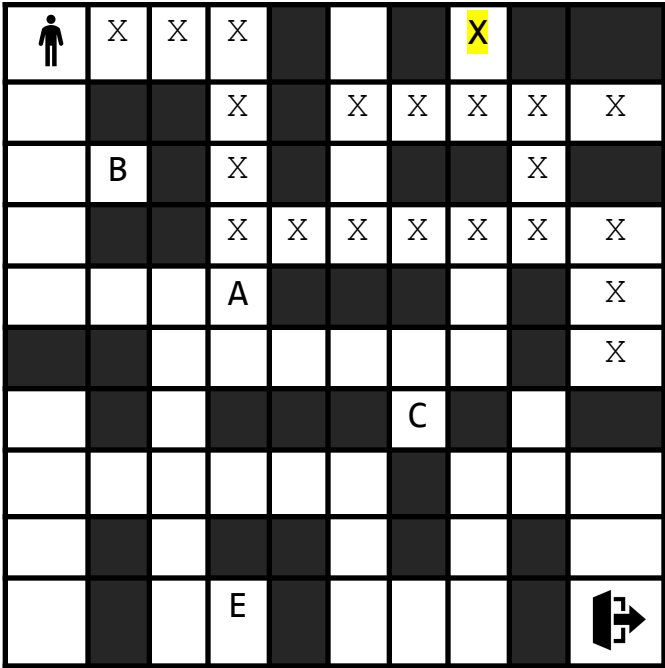
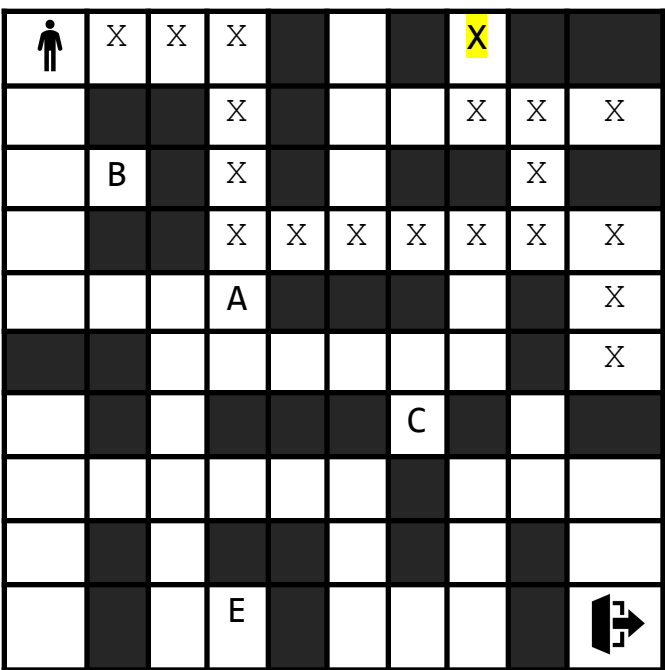
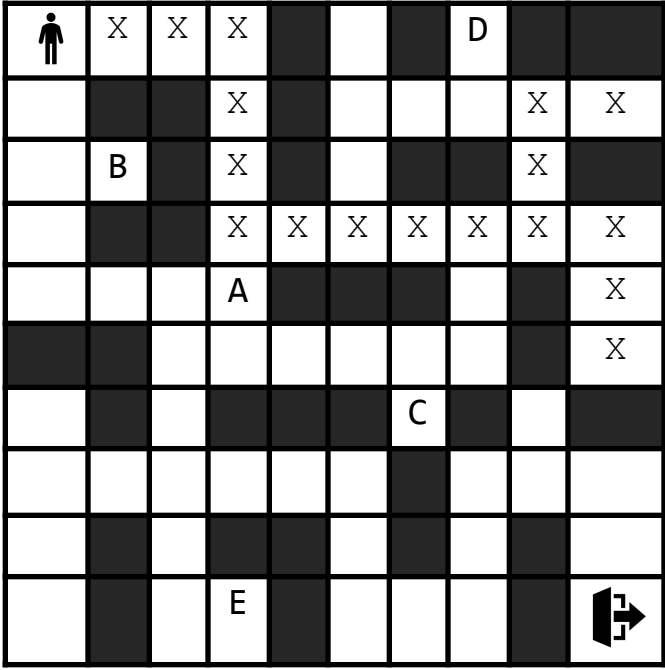
- Labirentte yalnızca yatay ve dikey hareket edilebilir; çapraz ilerlemek mümkün değildir.
- Can üzerine bastığı her karenin zeminine "ziyaret edildi" işareti bırakmaktadır.
- Can, bulunduğu kareden yeni bir adıma geçerken çevresini daima şu sabit öncelik sırasına göre tarar: 1. SAĞ, 2. AŞAĞI, 3. SOL, 4. YUKARI.
- Bu tarama sırasında karşılaştığı ilk "duvar olmayan" ve "daha önce ziyaret edilmemiş" kareye adım atar. (Örneğin; sağındaki kare uygunsa, diğer yönlere bakmaksızın doğrudan sağa ilerler).
- Eğer etrafındaki tüm komşu kareler duvarla kaplıysa veya daha önceden ziyaret edilmişse (çıkılmaz sokak durumu), Can geldiği yoldan bir adım geriye döner. Döndüğü bu karede, yön taramasına kaldığı (denemediği) öncelik sırasından devam eder.
- Labirentin etrafının tamamen duvar olduğu ve o kısımlara ilerlenemeyeceği kabul edilmiştir.


Can, bu algoritmik yaklaşımı eksiksiz bir şekilde uygulayarak labirentin çıkışını bulmaya çalışmaktadır. Can labirentin sol üst kısmından bu strateji ile çıkışı aramaktadır. Çıkış ise sağ alt tarafta bulunmaktadır. Buna göre Can çıkışı ararken yukarıda belirtilen karelerden hangilerini ziyaret etmiştir?


- A) Yalnız A
- B) A ve B
- C) A ve C
- D) C ve D
- E) C ve E



ÇÖZÜM








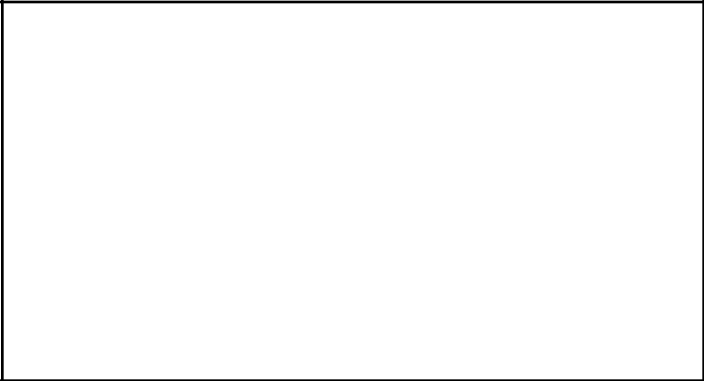
	B		X		X			X	
			X	X	X	X	X	X	X
			A				X		X
						X	X		X
						C			
			E						

	B		X		X			X	
			X	X	X	X	X	X	X
			A				X		X
						X	X		X
						X			
			E						

	X	X	X		X		X		
			X		X	X	X	X	X
	B		X		X			X	
			X	X	X	X	X	X	X
			A				X		X
		X	X	X	X	X	X		X
						X			
			E						

	X	X	X		X		X		
			X		X	X	X	X	X
	B		X		X			X	
			X	X	X	X	X	X	X
			A				X		X
		X	X	X	X	X	X		X
		X					X		
		X							
			E						

	X	X	X		X		X		
			X		X	X	X	X	X
	B		X		X			X	
			X	X	X	X	X	X	X
			A				X		X

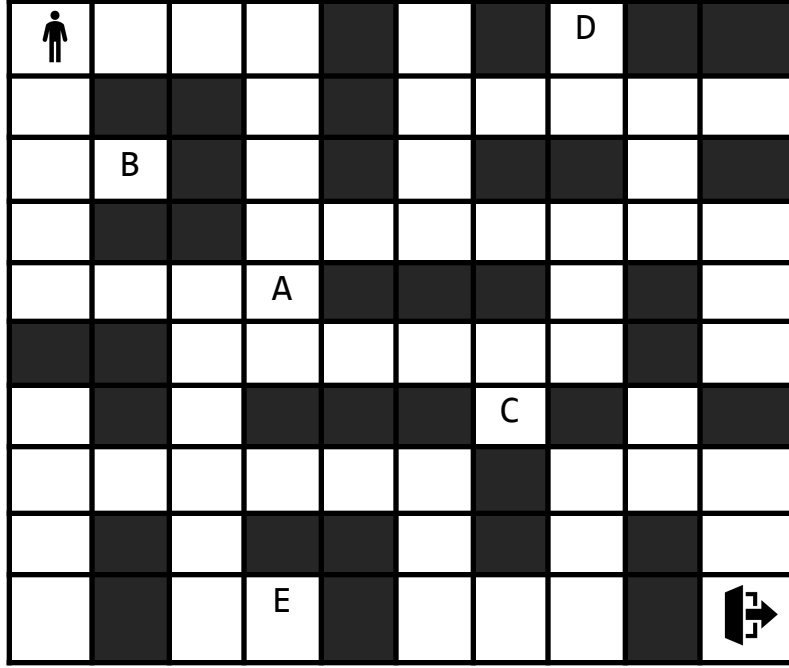


		X	X	X	X	X	X		X
		X				X			
		X	X	X	X		X	X	X
					X		X		X
			E		X	X	X		X

Sadece C ve D kareleri ziyaret edilmiştir.

CEVAP: D

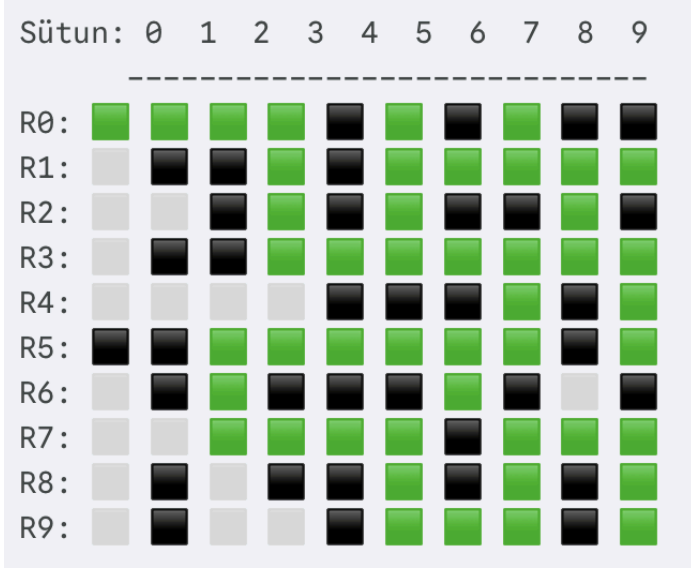
SORU 12



Soru 11’de verilen açıklamalara göre başlangıç ve bitiş kareleri de sayıldığında toplam kaç kare ziyaret edilmiştir?

- A) 33
- B) 46
- C) 47
- D) 54
- E) 65

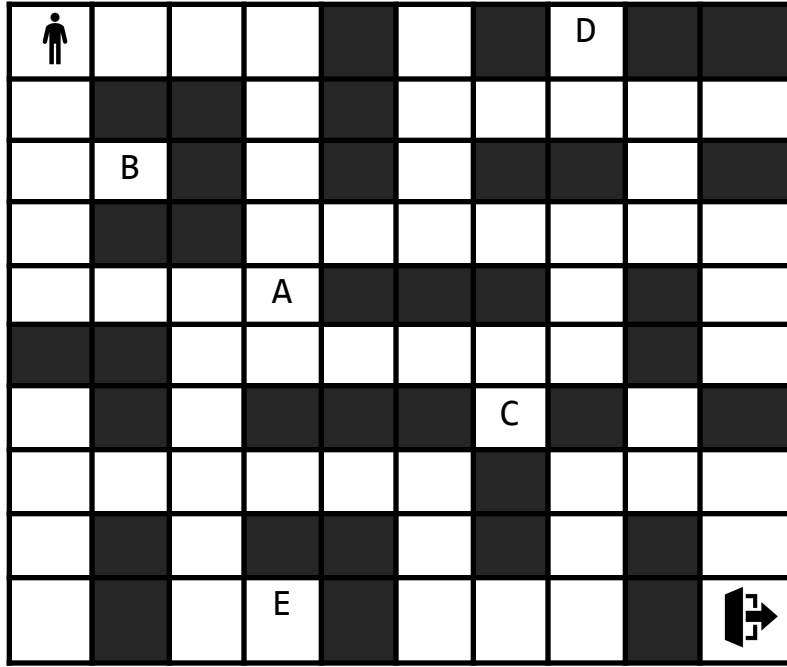
ÇÖZÜM



47 kare ziyaret edilmiştir.

CEVAP: C









































































































































































































SORU 13



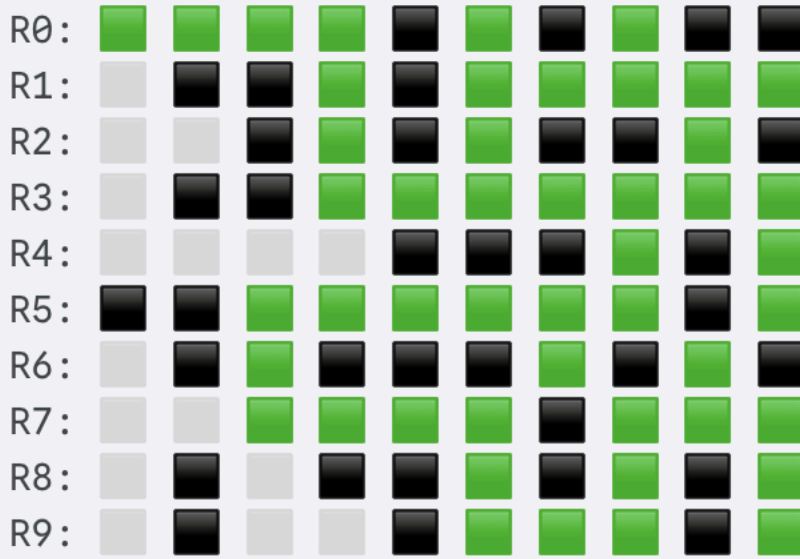
11. Soru düşünülürken kurallar içerisinde sadece komşu karelere bakma önceliğimizin sırasını en öncelikliден en az öncelikliye göre yazacak olursak, aşağıdaki stratejilerden hangisi çözüme en az kareyi ziyaret ederek ulaşır?

- A) SAĞ, YUKARI, SOL, AŞAĞI
- B) SOL, YUKARI, SAĞ, AŞAĞI
- C) YUKARI, SOL, AŞAĞI, SAĞ
- D) AŞAĞI, SOL, YUKARI, SAĞ
- E) AŞAĞI, SAĞ, YUKARI, SOL

ÇÖZÜM

<p>Sütun: 0 1 2 3 4 5 6 7 8 9</p> <p>-----</p> <p>R0:          </p> <p>R1:          </p> <p>R2:          </p> <p>R3:          </p> <p>R4:          </p> <p>R5:          </p> <p>R6:          </p> <p>R7:          </p> <p>R8:          </p> <p>R9:          </p>	<p>SAĞ YUKARI SOL AŞAĞI</p> <p>74 ADIM</p>
<p>Sütun: 0 1 2 3 4 5 6 7 8 9</p> <p>-----</p> <p>R0:          </p> <p>R1:          </p> <p>R2:          </p> <p>R3:          </p> <p>R4:          </p> <p>R5:          </p> <p>R6:          </p> <p>R7:          </p> <p>R8:          </p> <p>R9:          </p>	<p>SOL YUKARI SAĞ AŞAĞI</p> <p>74 ADIM</p>

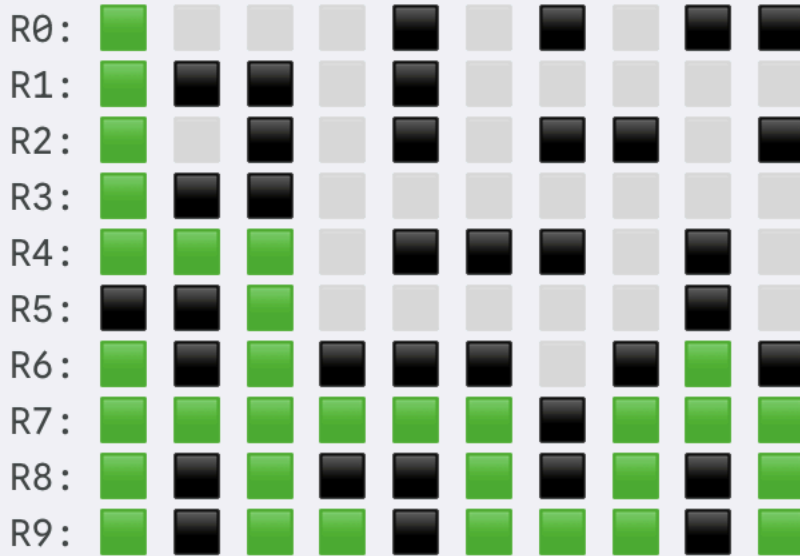
Sütun: 0 1 2 3 4 5 6 7 8 9



YUKARI SOL AŞAĞI SAĞ

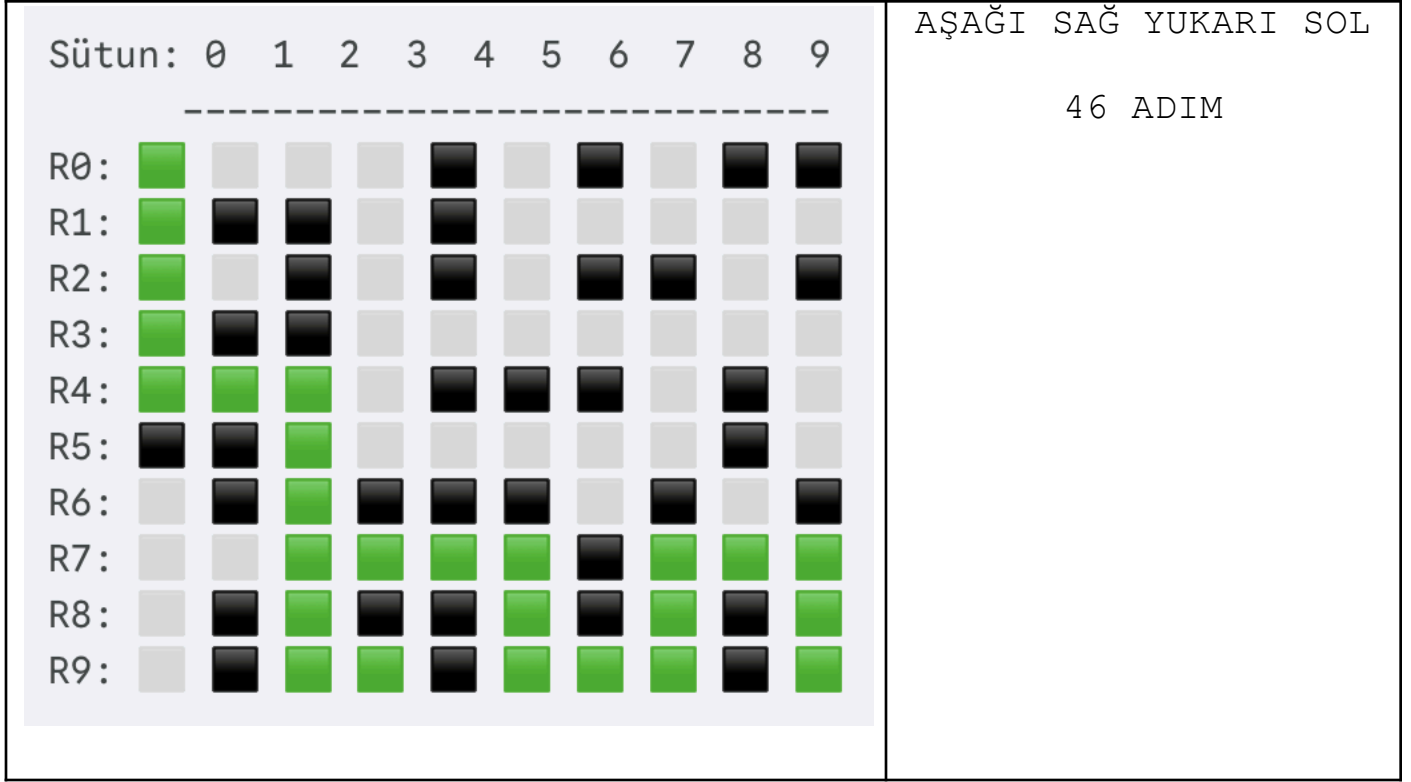
74 ADIM

Sütun: 0 1 2 3 4 5 6 7 8 9



AŞAĞI SOL YUKARI SAĞ

52 ADIM



Bu problem için en iyi strateji aşağı, sağ, yukarı, sol şeklindedir.

CEVAP: E

SORU 14

Bir uzay istasyonundaki arıza sonucunda radyasyon sızıntısı olmuştur ve istasyonun acilen tahliye edilmesi gerekmektedir. Radyasyon sızıntısı, personelin mahsur kaldığı "Sistem Odası" ile güvenli bölge olan "Çıkış Odası" arasındaki bağlantı koridorunu kaplamıştır. Sistem odasındaki 6 personelin tahliye edilebilmesi için uyulması gereken kurallar şunlardır:

- Ellerin radyasyon dalgalarını saptıran yalnızca bir adet taşınabilir koruyucu kalkan bulunmaktadır.
- Kalkanın etki alanı dar olduğu için koridordan geçerken aynı anda en fazla 2 kişiyi koruyabilmektedir.
- Kalkan uzaktan fırlatılamaz; güvenli bölgeye (Çıkış Odasına) geçen personelden birinin, geride kalanları almak için kalkanı Sistem Odasına yürüyerek geri getirmesi şarttır.

Personelin koridoru geçme süreleri birbirinden farklıdır. İki kişi birlikte geçerken mecburen daha yavaş olanın hızında ilerlemek zorundadırlar. Personelin tek başlarına koridoru geçiş süreleri şu şekildedir:

- Pilot: 1 dakika
- Sistem Mühendisi: 2 dakika
- Bilim İnsanı: 5 dakika
- Zırhlı Güvenlik Görevlisi: 8 dakika
- Yaşlı Personel: 10 dakika
- Yaralı Personel: 15 dakika

Buna göre, kusursuz bir tahliye planı yapıldığında, tüm personelin çıkış odasına ulaşması **en az** kaç dakika sürer?

- A) 33
- B) 35
- C) 37
- D) 41
- E) 44

ÇÖZÜM

Bu soruda kural şudur: **En yavaşları grupla, feneri en hızlılara taşı!** 6 kişi olduğumuz için bu "fedakarlık" döngüsünü tam 2 kez ardaya arkaya kurmamız (hem 10-15 ikilisini, hem de 5-8 ikilisini kendi içlerinde eritmemiz) gerekmektedir.

Bölüm 1: En Ağır İkiliyi (10 ve 15) Karşıya Atmak

1. **A (1) ve B (2) birlikte geçer.** Süre: **2 dk.**
2. **A (1) kalkanı geri getirir.** Süre: **1 dk.** (*Şu an kalkan başlangıçta, karşıda B var*)
3. **E (10) ve F (15) birlikte geçer.** Süre: **15 dk.** (*En büyük yükü 15'in içinde erittik!*)

4. **B (2) kalkanı geri getirir.** Süre: **2 dk.** (10 ve 15 karşıda güvende. A ve B tekrar başlangıçta!) İlk aşamanın toplam maliyeti: 20 Dakika.

Bölüm 2: Orta Ağırlıktaki İkiliyi (5 ve 8) Karşıya Atmak Şimdi başlangıç noktasında A(1), B(2), C(5) ve D(8) kaldı. Aynı taktiği birebir tekrarlıyoruz: 5. **A (1) ve B (2) birlikte geçer.** Süre: **2 dk.** 6. **A (1) feneri geri getirir.** Süre: **1 dk.** 7. **C (5) ve D (8) birlikte geçer.** Süre: **8 dk.** (Büyük yükü yine erittik!) 8. **B (2) kalkanı geri getirir.** Süre: **2 dk.** İkinci aşamanın toplam maliyeti: 13 Dakika.

Bölüm 3: Son Kapanış Başlangıç noktasında sadece kahramanlarımız A (1) ve B (2) kaldı. Fener de onlarda. 9. **A (1) ve B (2) birlikte geçer.** Süre: **2 dk.**

Toplam Minimum Süre: $20 + 13 + 2 = 35$ Dakika.

CEVAP: B

SORU 15

Bir okulun Zeka Oyunları Kulübü öğretmeni Ayşe, Buse ve Cihat isimli üç öğrencisine ödüllü bir mantık bulmacası hazırlar. Öğrencileri en arkada Ayşe, orta sırada Buse ve ön sırada Cihat olacak şekilde oturtur. Ayşe önündeki Buse ve Cihat'ı görebilmektedir. Buse sadece Cihat'ı görebilmektedir ve Cihat diğerlerini görememektedir. Öğretmen elinde 3 tane kırmızı ve 2 tane beyaz olmak üzere toplam 5 tane yapışkan kağıt olduğunu belirtir. Öğretmen rastgele 3 kağıdı her bir öğrencinin sırtına birer tane olacak şekilde yapıştırır. Öğrenciler kendilerinin kağıtlarını görememekte ve arkalarını dönememektedir. Öğretmen daha sonrasında önce Ayşe'ye sonra Buse'ye ve daha sonrasında Cihat'a sırtlarındaki kağıtların rengini sorar ve eğer bir tahminleri varsa bunu ispatlamaları istenir. Öğrenciler çok zekidir ve sadece sırtlarındaki kağıdın renginden kesin olarak eminlerse tahminde bulunurlar. Sonra cevap veren öğrenciler önceki öğrencilerinde cevaplarını ispatlarında kullanabilirler.

Öğretmen sırasıyla sorduğunda Ayşe "Kesin olarak bilmiyorum" diye cevaplar. Buse "Bende kesin olarak bilmiyorum" diye cevaplar ve bu cevapların ardından Cihat "Benim kağıdımın rengi kesin olarak kırmızıdır" der. Buna göre öğrencilerin kağıtlarının renkleri aşağıdakilerden hangisi veya hangileri olabilir?

	Ayşe	Buse	Cihat
I.	Kırmızı	Beyaz	Kırmızı
II.	Kırmızı	Beyaz	Beyaz
III.	Beyaz	Kırmızı	Kırmızı

- A) Yalnız I
- B) Yalnız II
- C) I ve II
- D) I ve III
- E) II ve III

ÇÖZÜM

Öncül II'nin Analizi: Kırmızı – Beyaz – Beyaz (NEDEN OLAMAZ?)

Eğer durum bu olsaydı, en arkadaki Ayşe önünde duran Buse ve Cihat'ın sırtındaki **iki adet Beyaz kağıt** görecekti. Öğretmenin elinde sadece 2 Beyaz kağıt olduğu için, Ayşe kendi sırtındakinin kesinlikle "Kırmızı" olduğunu anında bulur ve "*Biliyorum, kağıdım Kırmızı!*" derdi. Ancak hikayede Ayşe "**Bilemiyorum**" diyor. Bu yüzden önündeki ikili (Beyaz-Beyaz) olamaz. II. öncül kesinlikle yanlıştır.

Öncül I'in Analizi: Kırmızı – Beyaz – Kırmızı (OLABİLİR Mİ?)

- **Ayşe'nin durumu:** Ayşe öndekilere bakar ve (Beyaz, Kırmızı) görür. İki beyaz görmediği için kendi kağıdının kesin olarak bilemez ve "Bilemiyorum" der. (*Senaryoyla uyumlu*).
- **Buse'in durumu:** Buse, Ayşe'nin iki beyaz görmediğini anlar. Demek ki Buse ve Cihat ikilisi "Beyaz-Beyaz" değildir. Buse önündeki Cihat'a bakar ve "Kırmızı" görür. Kendi

kağıdı Beyaz da olsa Kırmızı da olsa bu kural bozulmayacağı için Buse kendi kağıdından emin olamaz ve "Bilemiyorum" der. (*Senaryoyla uyumlu*).

- **Cihat'ın durumu:** Cihat, Buse'in neden bilemediğini çözer: "*Eğer benim kağıdım Beyaz olsaydı, Buse kendi kağıdının Kırmızı olduğunu anlardı. Buse bilemediğine göre benim kağıdım Beyaz değil!*" der ve Kırmızı olduğunu bulur. (*Senaryoyla uyumlu*).
- **Sonuç:** I. durum mümkündür.

Öncül III'ün Analizi: Beyaz – Kırmızı – Kırmızı (OLABİLİR Mİ?)

- **Ayşe'nin durumu:** Ayşe öndekilere bakar ve (Kırmızı, Kırmızı) görür. Kendi kağıdını bilemez ve "Bilemiyorum" der. (*Senaryoyla uyumlu*).
- **Buse'in durumu:** Buse önündeki Cihat'da "Kırmızı" görür. Yine kendi kağıdından emin olamaz ve "Bilemiyorum" der. (*Senaryoyla uyumlu*).
- **Cihat'ın durumu:** Cihat yine diyaloglardan yola çıkarak kendi kağıdının kesinlikle Kırmızı olduğunu yukarıdaki mantıkla bulur. (*Senaryoyla uyumlu*).
- **Sonuç:** III. durum da mümkündür.

CEVAP: D

[Soru 16-30 İin Aıklama]

- Soruları C programlama dili erevesinde cevaplayınız.
- Derleyici olarak gcc kullanıldığını varsayınız.
- Gerekli tm bařlık (header) dosyalarının verilen programa dahil edildiğini varsayınız.

SORU 16

```
int main() {  
    int i;  
    int toplam = 0;  
    for (i = 1; i <= 5; i+=2) {  
        toplam += i;  
    }  
    printf("%d\n", toplam);  
    return 0;  
}
```

Yukarıdaki programın ekran ıktısı nedir?

- A) 3
- B) 4
- C) 5
- D) 9
- E) Hibiri

ÖZÜM

Kodda toplam deęiřkeni 0 olarak bařlatılıyor ve for dngüsü i=1'den bařlayıp i<=5 kořulu saęlandığı srece her adımda i'yi 2 artırarak ilerliyor. Dngü sırasıyla i=1, i=3 ve i=5 deęerlerini alıyor, her adımda toplam deęiřkenine i deęeri ekleniyor. Bu durumda toplam nce 1, sonra 1+3=4, en son 4+5=9 oluyor. Dngü i=7 olduęunda kořul saęlanmadığı iin sonlanıyor ve printf fonksiyonu ile ekrana 9 yazdırılıyor.

CEVAP: D

SORU 17

```
void arttir(int a) {
    if (a >= 10) a = a + 1;
    else a = a + 2;
}

int main() {
    int a = 10;
    arttir(a);
    printf("%d\n", a);
    return 0;
}
```

Yukarıdaki programın ekran çıktısı nedir?

- A) Kod hata vereceğinden ekran çıktısı oluşmaz.
- B) 10
- C) 11
- D) 12
- E) 13

ÇÖZÜM

Bu kodda main fonksiyonu içinde a değişkenine 10 değeri atanıyor, ardından `arttir(a);` satırıyla a değişkeni `arttir` fonksiyonuna parametre olarak gönderiliyor. Ancak burada önemli nokta, `arttir` fonksiyonunun parametreyi değer ile (call by value) almasıdır. Yani a'nın kendisi değil, sadece değerinin bir kopyası fonksiyona iletilir. Fonksiyon içinde a üzerinde yapılan değişiklikler, main içindeki a değişkenini etkilemez. `arttir` fonksiyonu çağrıldığında a = 10 olur, a >= 10 koşulu sağlandığı için a = a + 1; çalışır ve a=11 olur, ancak bu değişiklik sadece fonksiyonun yerel kopyasında kalır. Fonksiyon bittiğinde main'e geri dönülür ve a hâlâ 10 değerini taşır. Bu nedenle `printf("%d\n", a);` satırı ekrana 10 yazdırır.

CEVAP: B

SORU 18

```
int f(int n) {
    if (n == 1) return 0;
    return (n - 1) * f(n - 1) + 1;
}

int main() {
    printf("%d\n", f(5));
    return 0;
}
```

Yukarıdaki programın ekran çıktısı nedir?

- A) 3
- B) 10
- C) 41
- D) 206
- E) Kod hata vereceğinden ekran çıktısı oluşmaz.

ÇÖZÜM

Bu program, özyinelemeli bir fonksiyon olan f yardımıyla belirli bir hesaplama yapmaktadır. $f(5)$ çağırısı ile başlayan süreçte, $n=5$ olduğu için $(5-1) * f(4) + 1$ yani $4 * f(4) + 1$ hesaplanır. $f(4)$ ise $3 * f(3) + 1$, $f(3)$ ise $2 * f(2) + 1$, $f(2)$ ise $1 * f(1) + 1$ olarak hesaplanır. $f(1)$ base case'tir ve 0 döndürür. Bu durumda $f(2) = 1 * 0 + 1 = 1$, $f(3) = 2 * 1 + 1 = 3$, $f(4) = 3 * 3 + 1 = 10$, $f(5) = 4 * 10 + 1 = 41$ olarak hesaplanır. Sonuçta program 41 çıktısını üretir.

CEVAP: C

SORU 19

```
int main() {  
    int arr[] = {10, 20, 30, 40};  
    int *p = arr;  
    printf("%d ", *(p + 2));  
    printf("%d", p[1]);  
    return 0;  
}
```

Yukarıdaki programın ekran çıktısı nedir?

- A) 10 20
- B) 20 30
- C) 30 20
- D) 30 40
- E) Kod hata vereceğinden ekran çıktısı oluşmaz.

ÇÖZÜM

Kodda arr dizisi {10, 20, 30, 40} olarak tanımlanıyor ve `int *p = arr;` ifadesiyle p işaretçisi dizinin ilk elemanını (`arr[0]`) gösteriyor.

- İlk `printf("%d ", *(p + 2));` ifadesinde `p + 2`, p'den iki adım ilerleyerek dizinin üçüncü elemanını (`arr[2]`) işaret ediyor. `*(p + 2)` ile bu elemanın değeri olan 30 ekrana yazdırılıyor ve ardından bir boşluk bırakılıyor.
- İkinci `printf("%d", p[1]);` ifadesinde `p[1]`, p'nin bir ilerisindeki elemanı (`arr[1]`) gösteriyor ve değeri 20 ekrana yazdırılıyor.

Ekran çıktısı 30 20 şeklinde olur.

CEVAP: C

SORU 20

```
int *ptr = (int*)malloc(5 * sizeof(int));
```

satırı ile ayrılan bellek alanı hakkında hangisi doğrudur?

- A) Bellek alanı otomatik olarak 0 ile doldurulur.
- B) Ayrılan alan stack (yığın) bölgesindedir.
- C) free(ptr) komutu kullanılmazsa da program bitince bellek anında boşalır.
- D) Bu satır derleme hatası verir; calloc kullanılmalıdır.
- E) Bellek yetersizse ptr değişkeni NULL değerini alır.

ÇÖZÜM

- A) Yanlış. malloc ile ayrılan bellek alanı ilk değer ataması yapılmaz, içeriğindeki değerler belirsizdir (çöp değer içerir). Oysa calloc ile ayrılan bellek sıfırlanır.
- B) Yanlış. malloc ile ayrılan bellek heap (yığın) bölgesinde yer alır. Stack bölgesi yerel değişkenler ve fonksiyon çağrıları için kullanılır.
- C) Yanlış. Program sonlandığında işletim sistemi genellikle ayrılan belleği geri alır ancak bu anlık ve garanti değildir; ayrıca program uzun süre çalışıyorsa free kullanılmazsa bellek sızıntısı (memory leak) oluşur. Doğru uygulama, dinamik olarak ayrılan belleğin free ile serbest bırakılmasıdır.
- D) Yanlış. malloc doğru bir kullanımdır, derleme hatası vermez. calloc farklı bir işlemdir ancak malloc da geçerlidir.
- E) Doğru. Eğer bellek yetersizse malloc NULL döndürür. Bu durumda ptr değişkeni NULL değerini alır.

CEVAP: E

SORU 21

```
int k = 2;
switch(k) {
    case 1: printf("A");
    case 2: printf("B");
    case 3: printf("C");
    default: printf("D");
}
```

Yukarıdaki programın ekran çıktısı nedir?

- A) B
- B) BCD
- C) BC
- D) BD
- E) D

ÇÖZÜM

Kodda k değişkeninin değeri 2 olduğu için switch yapısı case 2: etiketinden itibaren çalışmaya başlar. C dilinde switch içinde break ifadesi kullanılmadığında, ilgili case'den sonra gelen tüm case'ler ve default da sırayla çalışır. Bu nedenle önce case 2:'deki printf("B"); çalışır, ardından break olmadığı için case 3:'e geçilir ve printf("C"); çalışır, son olarak default: etiketine geçilir ve printf("D"); çalışır. Ekrana sırasıyla B, C ve D yan yana yazdırılır.

CEVAP: B

SORU 22

double *p; tanımı yapılmış bir sistemde, p adresi 1000 ise p + 1 işleminin sonucu hangi adrestir? (double = 8 byte varsayınız)

- A) 1000
- B) 1001
- C) 1004
- D) 1008
- E) 1016

ÇÖZÜM

C dilinde işaretçi aritmetiği yapılırken, bir işaretçiye tam sayı eklendiğinde adres, işaretçinin türünün boyutu kadar artar. Burada p bir double * türündedir ve double değişkeninin boyutu 8 byte olarak verilmiştir.

p adresi 1000 olduğunda, p + 1 işleminin sonucu:

$$1000+1\times 8=1008 \text{ olur.}$$

CEVAP: D

SORU 23

```
int main() {
    int i, j;

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

    return 0;
}
```

Yukarıdaki program ekrana kaç tane * basar?

- A) 3
- B) 4
- C) 5
- D) 6
- E) 7

ÇÖZÜM

Kodda dış döngü $i = 1$ 'den başlayıp $i \leq 3$ koşulu sağlandığı sürece çalışır. İç döngü ise $j = 1$ 'den başlar ve $j \leq i$ olduğu sürece `printf("*");` ile yıldız basar.

Adım adım:

- $i = 1$ için: iç döngü $j = 1$ 'den $j \leq 1$ 'e kadar çalışır \rightarrow 1 yıldız basılır.
- $i = 2$ için: iç döngü $j = 1$ 'den $j \leq 2$ 'ye kadar çalışır \rightarrow 2 yıldız basılır.
- $i = 3$ için: iç döngü $j = 1$ 'den $j \leq 3$ 'e kadar çalışır \rightarrow 3 yıldız basılır.

Toplam yıldız sayısı: $1+2+3=6$

CEVAP: D

SORU 24

```
int main() {
    int a[5] = {1,2,3,4,5};
    int *p = a;
    int sum = 0;
    int i;

    for (i = 0; i < ?; i++) {
        sum += *(p + i);
    }

    printf("%d\n", sum);
    return 0;
}
```

Yukarıdaki kodun çıktısının 15 olması için ? yerine ne yazılmalıdır?

- A) 3
- B) 4
- C) 5
- D) 6
- E) 10

ÇÖZÜM

Kodda a dizisi {1, 2, 3, 4, 5} olarak tanımlanmış ve p işaretçisi dizinin başlangıç adresini göstermektedir. Döngüde *(p + i) ile dizinin elemanları toplanmaktadır. Çıktının 15 olması için toplamın $1 + 2 + 3 + 4 + 5 = 15$ olması gerekir, yani dizinin tüm elemanlarının toplanması gerekmektedir. Bu durumda döngü 5 kez çalışmalıdır, çünkü dizi 5 elemanlıdır.

Döngü değişkeni i 0'dan başladığı için $i < 5$ koşulu ile 5 eleman ($i = 0, 1, 2, 3, 4$) döngüye girer ve toplam 15 olur.

CEVAP: C

SORU 25

```
int main() {  
    int a[4] = {2, 4, 6, 8};  
    int *p = a;  
  
    printf("%d\n", *(p + ?) + *(p + 3));  
    return 0;  
}
```

Aşağıdaki kodun çıktısının 14 olması için ? yerine aşağıdakilerden hangisi yazılmalıdır?

- A) 0
- B) 1
- C) 2
- D) 3
- E) Bilinemez.

ÇÖZÜM

*(p + 3) dizinin 4. elemanı olan 8'i gösterir. Çıktının 14 olması için *(p + ?) ifadesinin 6 vermesi gerekir. Dizide 6 değeri a[2]'dir ve p + 2 ile gösterilir. Bu nedenle ? = 2 olmalıdır.

CEVAP: C

SORU 26

```
int f1() {
    printf("A");
    return 0;
}
int f2() {
    printf("B");
    return 1;
}
int f3() {
    printf("C");
    return 2;
}
int main() {
    int s = f1() && f2() || f3() && f1() || f2() && f3();
    printf(" %d\n", s);
    return 0;
}
```

Yukarıdaki program ekrana aşağıdakilerden hangisi yazar?

- A) A 0
- B) ACA 0
- C) ACBC 1
- D) ACABC 1
- E) ABAC 2

ÇÖZÜM

Çözüm sürecinde dikkat edilmesi gereken iki temel kural vardır:

1. **İşlem Önceliği:** VE (&&) operatörünün VEYA (||) operatöründen önceliği daha yüksektir. Bu nedenle derleyici ifadeyi şu şekilde gruplar: (f1() && f2()) || (f3() && f1()) || (f2() && f3())
2. **Kısa Devre (Short-Circuit):** İfadeler soldan sağa doğru değerlendirilir. && işleminde ilk değer 0 ise ikinci kısım okunmaz. || işleminde ilk değer 1 (veya sıfırdan farklı bir sayı) ise ikinci kısım okunmaz.

Adım Adım Derleyici İzlemesi (Trace):

1. Grup: (f1() && f2()) Değerlendirmesi

- İlk olarak $f1()$ çağrılır. Ekran **A** yazdırır ve 0 döndürür.
- Sol taraf 0 olduğu için VE ($\&\&$) işlemi **kısa devre yapar**. Sağ tarafın sonucunun bir önemi kalmamıştır. $f2()$ fonksiyonu asla çağrılmaz.
- *Bu grubun nihai sonucu: 0.*

2. Grup: ($f3()$ $\&\&$ $f1()$) Değerlendirmesi

- Bir önceki grubun sonucu 0 olduğu için VEYA ($\|\|$) zinciri kopmaz, derleyici okumaya devam etmek zorundadır.
- $f3()$ çağrılır. Ekran **C** yazdırır ve 2 döndürür. C dilinde 0 dışındaki tüm sayılar mantıksal DOĞRU (TRUE) kabul edilir.
- Sol taraf doğru olduğu için VE ($\&\&$) işlemi devam eder. $f1()$ çağrılır. Ekran **A** yazdırır ve 0 döndürür.
- Mantıksal işlem: $2 \&\& 0$. Sonuç 0 (YANLIŞ) olur.
- *Bu grubun nihai sonucu: 0.*

3. Grup: ($f2()$ $\&\&$ $f3()$) Değerlendirmesi

- Önceki grupların toplam sonucu $0 \|\| 0 = 0$ olduğu için VEYA ($\|\|$) zinciri hala kopmamıştır. Son gruba girilir.
- $f2()$ çağrılır. Ekran **B** yazdırır ve 1 döndürür.
- Sol taraf doğru olduğu için sağa geçilir. $f3()$ çağrılır. Ekran **C** yazdırır ve 2 döndürür.
- Mantıksal işlem: $1 \&\& 2$. Her ikisi de DOĞRU olduğu için sonuç 1 (DOĞRU) olur.
- *Bu grubun nihai sonucu: 1.*

Nihai Sonuç ve Ekran Çıktısı:

- Tüm işlemlerin sonucu: $0 \|\| 0 \|\| 1 = 1$. Bu değer `sonuc` değişkenine atanır.
- Fonksiyonlardan sırasıyla gelen çıktılar: **A, C, A, B, C**
- Ana programdaki yazdırma işlemi: Boşluk ile birlikte `sonuc` değeri yazdırılır: **1**
- Tam ekran çıktısı: **ACABC 1**

CEVAP: D

SORU 27

```
// Alfabe Referansı: abcdefghijklmnopqrstuvwxyz

void f1(char c){
    if(c >= 'f'){
        f1(c-5);
    }
    printf("%c", c);
    return;
}

void f2(char c){
    if(c <= 'k'){
        f1(c+7);
    }
    else{
        f2(c-3);
    }
}

int main() {
    f2('r');
    return 0;
}
```

Yukarıdaki program ekrana aşağıdakilerden hangisi yazar?

- A) plfa
- B) afkp
- C) rlif
- D) chmr
- E) Program hata verir.

ÇÖZÜM

Program main metodu içinden `f2('r')` çağrısı ile başlar.

- **`f2('r')` Çağrısı:** 'r' harfi 'k' harfinden sonra gelir (alfabetik olarak daha büyüktür). `c <= 'k'` koşulu *sağlanmaz*. Kod `else` bloğuna girer ve `f2('r' - 3)` çağrısını yapar. 'r' harfinden 3 harf geriye gidersek 'o' elde ederiz.
- **`f2('o')` Çağrısı:** 'o' 'k' koşulu *sağlanmaz*. `f2('o' - 3)` çağrılır Yeni çağrı: `f2('l')`.
- **`f2('l')` Çağrısı:** 'l' 'k' koşulu *sağlanmaz*. `f2('l' - 3)` çağrılır Yeni çağrı: `f2('i')`.

- **f2('i')** Çağrısı: 'i' 'k' koşulu **sağlanır** ('i' harfi 'k'den önce gelir). *İşte burada durum geçişi (phase transition) gerçekleşir.* Program f1 fonksiyonuna atlar. İlgili satır f1('i' + 7) şeklindedir. 'i' harfine 7 eklersek 'p' elde ederiz Yeni çağrı: f1('p').

Daha sonrasında:

- **f1('p')** Çağrısı: 'p' 'f' koşulu sağlanır. f1('p' - 5) çağrılır Yeni çağrı: f1('k'). (*Bu esnada f1('p') beklemeye geçer*).
- **f1('k')** Çağrısı: 'k' 'f' koşulu sağlanır. f1('k' - 5) çağrılır Yeni çağrı: f1('f').
- **f1('f')** Çağrısı: 'f' 'f' koşulu sağlanır. f1('f' - 5) çağrılır Yeni çağrı: f1('a').
- **f1('a')** Çağrısı: 'a' 'f' koşulu **sağlanmaz**. Program if bloğunu atlar. Özyineleme (recursion) burada dibe vurur ve temel duruma (base case) ulaşılır.

Beklemeye alınan fonksiyonlar, yığından en üstten alta doğru (LIFO mantığıyla) çıkmaya başlar. Yazdırma işlemi sadece bu aşamada gerçekleşir.

- **Dönüş 1:** Aktif olan f1('a') fonksiyonu if bloğunu atladıktan sonra alt satıra geçer ve belleğindeki 'a' karakterini yazdırır.
 - Ekran Çıktısı: **a**
- **Dönüş 2:** f1('a') sonlanınca kontrol kendisini çağırın f1('f') fonksiyonuna döner. f1('f') kaldığı yerden (yani if bloğunun hemen altından) devam eder ve belleğindeki 'f' karakterini yazdırır.
 - Ekran Çıktısı: **af**
- **Dönüş 3:** Kontrol f1('k') fonksiyonuna döner. Kaldığı yerden devam edip kendi karakterini yazdırır.
 - Ekran Çıktısı: **afk**
- **Dönüş 4:** Kontrol f1('p') fonksiyonuna döner. Kendi karakterini yazdırır.
 - Ekran Çıktısı: **afkp**

CEVAP: B

SORU 28

```
void fonk(int *dizi, int *e, int x) {
    int i = *e;
    dizi[i] = x;
    (*e)++;
    while (i > 0) {
        int k = (i - 1) / 2;
        if (dizi[k] < dizi[i]) {
            int g = dizi[k];
            dizi[k] = dizi[i];
            dizi[i] = g;
            i = k;
        } else {
            break;
        }
    }
}

int main() {
    int dizi[10];
    int a = 0;
    fonk(dizi, &a, 50);
    fonk(dizi, &a, 80);
    fonk(dizi, &a, 30);
    fonk(dizi, &a, 90);
    fonk(dizi, &a, 70);
    for(int j = 0; j < a; j++) {
        printf("%d ", dizi[j]);
    }
    return 0;
}
```

Yukarıdaki program ekrana hangi çıktıyı yazdırır?

- A) 50 80 30 90 70
- B) 90 80 70 50 30
- C) 30 50 70 80 90
- D) 90 80 30 50 70
- E) 90 70 80 30 50

ÇÖZÜM

Adım Adım İzleme Tablosu:

1. **fonk(50) Ekleme:**
 - Dizi: [50]. (Döngüye girmez çünkü $i = 0$).
2. **fonk(80) Ekleme:**
 - Dizi sonuna eklenir: [50, 80]. $i = 1$.
 - $k = (1 - 1) / 2 = 0$.
 - $dizi[0] < dizi[1]$ yani DOĞRU. Takas yapılır.
 - Dizi: [80, 50]. $i = 0$ olur, döngü biter.
3. **fonk(30) Ekleme:**
 - Eklenir: [80, 50, 30]. $i = 2$.
 - $k = (2 - 1) / 2 = 0$.
 - $dizi[0] < dizi[2]$ yani YANLIŞ. Döngü kırılır.
 - Dizi: [80, 50, 30].
4. **fonk(90) Ekleme:**
 - Eklenir: [80, 50, 30, 90]. $i = 3$.
 - $k = (3 - 1) / 2 = 1$.
 - $dizi[1] < dizi[3]$ yani DOĞRU. Takas yapılır.
 - *Geçici Dizi:* [80, 90, 30, 50]. Yeni $i = 1$.
 - Döngü devam eder: $k = (1 - 1) / 2 = 0$.
 - $dizi[0] < dizi[1]$ yani DOĞRU. Takas yapılır.
 - Dizi: [90, 80, 30, 50]. $i = 0$ olur, döngü biter.
5. **fonk(70) Ekleme:**
 - Eklenir: [90, 80, 30, 50, 70]. $i = 4$.
 - $k = (4 - 1) / 2 = 1$.
 - $dizi[1] < dizi[4]$ yani YANLIŞ. Döngü kırılır.
 - Son Dizi: [90, 80, 30, 50, 70].

Nihai ekran çıktısı **90 80 30 50 70** olur.

CEVAP: D

SORU 29

```
int main() {
    int m[6][6] = {
        {12, 24, 38, 45, 56, 68},
        {18, 29, 42, 50, 61, 73},
        {25, 33, 49, 58, 65, 80},
        {31, 40, 54, 62, 72, 88},
        {39, 47, 60, 69, 78, 93},
        {46, 55, 67, 75, 84, 99}
    };

    int v = 60;
    int i = 0, j = 5;
    int c = 0;
    while(i < 6 && j >= 0) {
        c++;
        if (m[i][j] == v) {
            break;
        }
        else if (m[i][j] > v) {
            j--;
        }
        else {
            i++;
        }
    }
    printf("%d\n", c);
    return 0;
}
```

Yukarıdaki program ekrana hangi çıktıyı yazdırır?

- A) 6
- B) 8
- C) 12
- D) 15
- E) 36

ÇÖZÜM

Adım Adım İzleme Tablosu:

- Başlangıç: $i = 0, j = 5$, Hedef değer $v = 60$. Sayaç $c = 0$.
- **Adım 1:** $c=1$. Konum: $m[0][5]$ Değer: **68**. olduğu için $j--$ çalışır. Yeni konum: $(0, 4)$.
- **Adım 2:** $c=2$. Konum: $m[0][4]$ Değer: **56**. olduğu için $i++$ çalışır. Yeni konum: $(1, 4)$.
- **Adım 3:** $c=3$. Konum: $m[1][4]$ Değer: **61**. olduğu için $j--$ çalışır. Yeni konum: $(1, 3)$.
- **Adım 4:** $c=4$. Konum: $m[1][3]$ Değer: **50**. olduğu için $i++$ çalışır. Yeni konum: $(2, 3)$.
- **Adım 5:** $c=5$. Konum: $m[2][3]$ Değer: **58**. olduğu için $i++$ çalışır. Yeni konum: $(3, 3)$.
- **Adım 6:** $c=6$. Konum: $m[3][3]$ Değer: **62**. olduğu için $j--$ çalışır. Yeni konum: $(3, 2)$.
- **Adım 7:** $c=7$. Konum: $m[3][2]$ Değer: **54**. olduğu için $i++$ çalışır. Yeni konum: $(4, 2)$.
- **Adım 8:** $c=8$. Konum: $m[4][2]$ Değer: **60**. Koşul $m[i][j] == v$ sağlanır ($60 == 60$). Döngü `break` ile kırılır.

Hedef değere **8** döngü iterasyonu sonucunda ulaşılmıştır.

CEVAP: B

SORU 30

```
int main() {
    int arr[8] = {0, 1, 1, 0, 0, 1, 0, 1};
    int arr2[8];
    int j, i;
    for(j = 0; j < 3; j++) {
        for(i = 0; i < 8 - j; i++) {
            int a = (i == 0) ? 0 : arr[i - 1];
            int b = (i == 7) ? 0 : arr[i + 1];
            arr2[i] = a ^ b;
        }
        for(i = 0; i < 8; i++) {
            arr[i] = arr2[i];
        }
    }
    int s = 0;
    for(i = 0; i < 8; i++) {
        s += arr[i];
    }

    printf("%d\n", s);
    return 0;
}
```

Yukarıdaki program ekrana aşağıdakilerden hangisini yazar?

- A) 3
- B) 4
- C) 5
- D) 6
- E) 7

ÇÖZÜM

Adım Adım İzleme:

- **Başlangıç:** arr = [0, 1, 1, 0, 0, 1, 0, 1], arr2 = Çöp değerler (Garbage)

1. İterasyon (j = 0):

- İlk döngü sınırı: $i < 8 - 0$ (Yani i, 0'dan 7'ye kadar tüm diziyi günceller).
- i=0: sol 0, sağ 1 arr2[0] = $0^1 = 1$
- i=1: sol 0, sağ 1 arr2[1] = $0^1 = 1$

- $i=2$: sol 1, sag 0 $arr2[2] = 1^0 = 1$
- $i=3$: sol 1, sag 0 $arr2[3] = 1^0 = 1$
- $i=4$: sol 0, sag 1 $arr2[4] = 0^1 = 1$
- $i=5$: sol 0, sag 0 $arr2[5] = 0^0 = 0$
- $i=6$: sol 1, sag 1 $arr2[6] = 1^1 = 0$
- $i=7$: sol 0, sag 0 $arr2[7] = 0^0 = 0$ (sag sınır 0 kabul edilir)
- **Kopyalama Döngüsü:** $arr2$ 'nin **tamamı** (8 eleman) arr 'ye kopyalanır.
- *Adım 0 Sonu* arr : [1, 1, 1, 1, 1, 0, 0, 0]

2. İterasyon ($j = 1$):

- İlk döngü sınırı: $i < 8 - 1$ (Yani i , 0'dan 6'ya kadar çalışır!). $arr2[7]$ güncellenmez.
- $i=0$: sol 0, sag 1 $arr2[0] = 1$
- $i=1$: sol 1, sag 1 $arr2[1] = 0$
- $i=2$: sol 1, sag 1 $arr2[2] = 0$
- $i=3$: sol 1, sag 1 $arr2[3] = 0$
- $i=4$: sol 1, sag 0 $arr2[4] = 1$
- $i=5$: sol 1, sag 0 $arr2[5] = 1$
- $i=6$: sol 0, sag 0 $arr2[6] = 0$ (sag yani $arr[7]$ önceki adımdan 0'dır)
- **Kritik Nokta:** $arr2[7]$ güncellenmedi! Ancak bellekte bir önceki iterasyondan kalan 0 değerini taşıyor.
- **Kopyalama Döngüsü:** Yine dizinin **tamamı** (8 eleman) kopyalanır.
- *Adım 1 Sonu* arr : [1, 0, 0, 0, 1, 1, 0, 0]

3. İterasyon ($j = 2$):

- İlk döngü sınırı: $i < 8 - 2$ (Yani i , 0'dan 5'e kadar çalışır!). $arr2[6]$ ve $arr2[7]$ güncellenmez.
- $i=0$: sol 0, sag 0 $arr2[0] = 0$
- $i=1$: sol 1, sag 0 $arr2[1] = 1$
- $i=2$: sol 0, sag 0 $arr2[2] = 0$
- $i=3$: sol 0, sag 1 $arr2[3] = 1$
- $i=4$: sol 0, sag 1 $arr2[4] = 1$
- $i=5$: sol 1, sag 0 $arr2[5] = 1$ (sag yani $arr[6]$ 0'dır)
- **Kritik Nokta:** $arr2[6]$ ve $arr2[7]$ güncellenmedi. Eski değerleri olan 0 ve 0'ı koruyorlar.
- **Kopyalama Döngüsü:** Yine 8 eleman kopyalanır.
- *Adım 2 Sonu* arr : [0, 1, 0, 1, 1, 1, 0, 0]

Sonuç: Dizinin son halinde 4 adet 1 bulunur. Çıktı 4'tür.

CEVAP: B

SINAV BİTTİ
Cevaplarımızı kontrol ediniz.

BU SAYFA BOŞ BIRAKILMIŞTIR.