

Kitapçık Kodu :



**TÜRKİYE BİLİMSEL VE TEKNOLOJİK ARAŞTIRMA KURUMU
BİLİM İNSANI DESTEK PROGRAMLARI BAŞKANLIĞI**

**33. BİLİM OLİMPİYATLARI -2025
BİRİNCİ AŞAMA SINAVI**

ÖĞRENCİ - ÇÖZÜMLER

ORTAOKUL BİLGİSAYAR

Soru Kitapçığı Türü

A

17 Mayıs 2025 Cumartesi, 09.30 – 11.00

ADAYIN ADI SOYADI :
T.C. KİMLİK NO :
OKULU / SINIFI :
SINAVA GİRDİĞİ İL :

SINAVLA İLGİLİ UYARILAR:

- Bu sınav çoktan seçmeli 30 adet sorudan oluşmaktadır, süre 90 dakikadır.
- Her sorunun sadece bir doğru cevabı vardır. Doğru cevabınızı, cevap kâğıdınızdaki ilgili kutucuğu **tamamen karalayarak** işaretleyiniz. Soru kitapçığındaki hiç bir işaretleme değerlendirmeye alınmayacaktır.
- Her soru eşit değerde olup, dört yanlış cevap bir doğru cevabı götürcektir.** Boş bırakılan soruların değerlendirmede olumlu ya da olumsuz bir etkisi olmayacaktır.
- Sorular zorluk sırasında DEĞİLDİR. Dolayısıyla yanıtlamaya geçmeden önce bütün soruları gözden geçirmeniz önerilir.
- Sınavda herhangi bir yardımcı materyal, elektronik hesap makinesi ya da karalama kağıdı kullanılması yasaktır. Soru kitapçığındaki boşlukları karalama için kullanabilirsiniz.
- Sınav süresince görevlilerle konuşulması ve soru sorulması, öğrencilerin birbirlerinden kalem, silgi vb. şeyler istemeleri yasaktır.
- Sınava giren adayın bir soruya itiraz etmek istemesi durumunda, sınav soruları ve cevap anahtarı TÜBİTAK'ın internet sayfasında (<https://bilimolimpiyatlari.tubitak.gov.tr>) yayımlandıktan sonra 7 işgünü içerisinde, kanıtları ile birlikte, TÜBİTAK'a başvurusu gerekir. Bu tarihten sonra yapılacak başvurular işleme konmayacaktır. Sadece sınava giren adayın sorulara itiraz hakkı vardır, üçüncü kişilerin sınav sorularına itirazı işleme alınmayacaktır.
- TÜBİTAK Bilim Olimpiyatı-Birinci Aşama Sınavı'nda sorulan soruların üçüncü kişiler tarafından kullanılması sonucunda doğacak olan hukuki sorunlardan TÜBİTAK ve Atatürk Üniversitesi sorumlu tutulamaz. Atatürk Üniversitesi, bu tip durumlarda sorular ile ilgili görüş bildirmek zorunda değildir.
- Sınav sırasında kopya çeken, çekmeye teşebbüs eden ve kopya verenlerin kimlikleri sınav tutanağına yazılacak ve bu kişilerin sınavları geçersiz sayılacaktır. Görevliler kopya çekmeye veya vermeye kalkışanları uyarmak zorunda değildir, sorumluluk size aittir.
- Sınav başladıktan sonraki ilk yarım saat içinde sınav salonundan ayrılmak yasaktır.
- Sınav süresince sınava giriş belgenizi ve geçerli bir kimlik belgesini masanızın üzerinde bulundurunuz.
- Sınav salonundan ayrılmadan önce cevap kâğıdınızı ve soru kitapçığını görevlilere teslim etmeyi unutmayınız.

Başarılar dileriz.

SORU 1

Bir mecliste yer alan 3 partinin sandalye sayıları, kadın ve erkek vekiller cinsinden aşağıdaki tabloda verilmiştir:

	A Partisi	B Partisi	C Partisi
Erkek	20	10	4
Kadın	10	5	6

Bu mecliste temsili bir komisyon oluşturulması, bu komisyonda her bir partinin meclisteki dağılıma eş şekilde temsil edilmesi, her partinin temsilcilerinde kadın ve erkek sayılarının eşit olması, her partinin en az 5 kişi ile temsil edilmesi kararlaştırılmıştır. Bu komisyon en az kaç kişi ile oluşturulabilir?

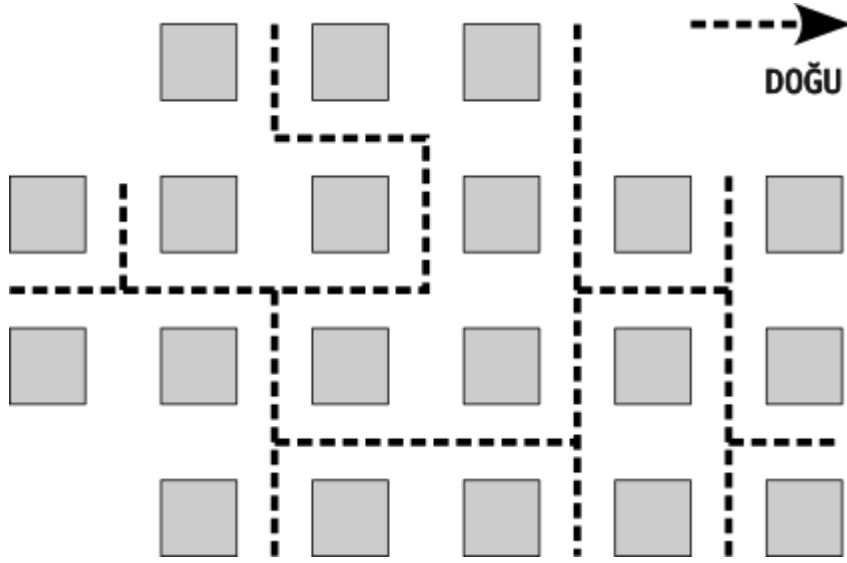
- A) 55
- B) 44
- C) 33
- D) 22
- E) 11

ÇÖZÜM

Bu sorunun İPTAL edilmesine karar verilmiştir.

CEVAP: İPTAL

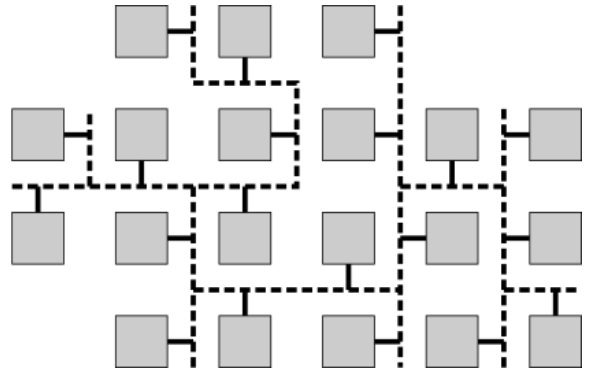
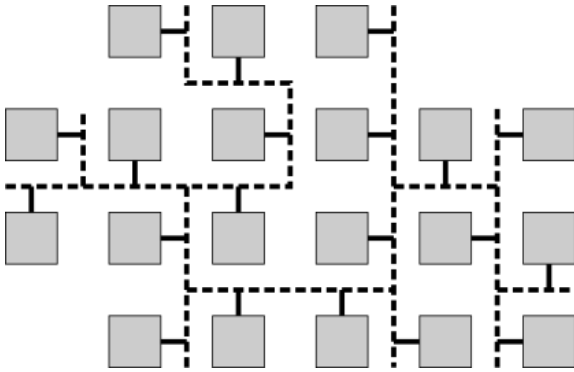
SORU 2



Şekilde yeni inşa edilmekte olan bir mahalledeki evler (kareler) ve yollar (kesikli çizgiler) görülmektedir. Mahalledeki her evin yanından geçen bir yola bakan bir kapısı olacaktır. Her evin en fazla bir kapısı olabilir. Bir yolun karşılıklı iki yanında olan iki evin kapıları birbirini görecek şekilde yapılamaz. Bir başka deyişle bir yolun bir noktasına yalnızca bir evin kapısı bağlanabilir. Bu şartlar altında, bu mahalledeki bütün kapılar yapıldığında kaç tanesi DOĞU yönüne bakacaktır?

- A) 9
- B) 7
- C) 5
- D) 4
- E) 3

ÇÖZÜM



Olası iki çözüm şekilde verilmiştir. Her ikisinde de 9 adet kapı doğu yönüne bakmaktadır.

CEVAP: A

SORU 3

Bir odanın, bir kenarı 9 birim olan kare tabanı, boyutları 2×3 ve 3×5 birim olan dikdörtgen fayanslar ile kaplanacaktır. Kaplama esnasında fayansların kırılarak küçültülmesi veya tabanda boş yer bırakılması kabul edilmemektedir. Bu kaplama işi, kullanılacak fayansların sayısı açısından kaç farklı şekilde yapılabilir? (Aynı sayıda fayans ile yapılan farklı kaplamalar 1 kere sayılacaktır)

- A) Yapılamaz
- B) 1
- C) 2
- D) 3
- E) 4

ÇÖZÜM

Alanlar arasındaki ilişkiyi birinci dereceden iki bilinmeyenli bir denklemle ifade edersek $81 = 6x + 15y$ olmalıdır. Bu denklemi çözen 3 farklı x, y ikilisi vardır: $(1, 5)$, $(6, 3)$ ve $(11, 1)$. Bunlardan $(6, 3)$ ve $(11, 1)$ in kaplamada kullanılabileceği kolayca gösterilebilir. $(1, 5)$ ise mümkün değildir. Çünkü, 5 adet 3×5 lik parça tabana koyulduğunda, kalan 6 birimlik alan dağınık halde olmaktadır. Bu yüzden taban, fayans sayıları açısından iki farklı şekilde doldurulabilir.

CEVAP: C

SORU 4

10 katlı bir apartmanın her katında farklı sayıda oda vardır. Her katın alanı 10 birimdir. Apartmandaki en küçük odanın alanı 1 birimdir. Her kattaki en küçük odalar kırmızı ile boyanacaktır. Bir odanın boyanması için 1 birim alanda 1 birim boya kullanılması gerekiyorsa, toplamda en az kaç birim boya kullanılması gerekir? (Bir katta en küçük alana sahip odadan birden fazla varsa, hepsi boyanacaktır.)

- A) 10
- B) 19
- C) 55
- D) 45
- E) 44

ÇÖZÜM

Başta verilen şartlar gereği katlardaki oda sayıları 1 den 10 a kadar değişmektedir. En küçük odanın alanını ve adedini minimize etmek aşağıdaki şekilde mümkündür. Boyanacak odalar kalın olarak gösterilmiştir.

- 1 oda olan kat: **10**
- 2 oda olan kat: 9, **1**
- 3 oda olan kat: 7, 2, **1** (8,1,1 değil)
- 4 oda olan kat: 5, 2, 2, **1**
- 5 oda olan kat: 3, 2, 2, 2, **1**
- 6 oda olan kat: 2, 2, 2, 2, **1, 1**
- 7 oda olan kat: 2, 2, 2, **1, 1, 1, 1**
- 8 oda olan kat: 2, 2, **1, 1, 1, 1, 1, 1**
- 9 oda olan kat: 2, **1, 1, 1, 1, 1, 1, 1, 1**
- 10 oda olan kat: **1, 1, 1, 1, 1, 1, 1, 1, 1, 1**

Böylece toplam boyanacak oda alanı $10+1+1+1+1+2+4+6+8+10 = 44$ birimdir.

CEVAP: E

SORU 5

Güneye göç etmekte geciken bir kaz kafasını kaldırdığında üzerinden geçmekte olan bir kaz sürüsünü görür. Kaz kanunlarına göre bir sürüde en fazla 40 kaz olabildiğinden bu sürüye katılıp katılamayacağını anlamak için sürünün liderine seslenir: “Ey uçan kazlar! Siz kaç kazsınız?”. Kazların lideri bu soruya bir bulmaca ile cevap verir: “Ey tembel kaz! Bizim kadar daha olsa, sonra yarımız kadar daha olsa, sonra yarımımızın yarısı kadar daha olsa, bir de sen olsan 100 kaz ederiz!”. Tembel kaz bunu duyunca bir süre düşünür ve sonra bu soruyu çözmektense kış soğuklarında titremeyi tercih ederek uçan kazlara el sallayıp dönüp yürümeye başlar. Peki gerçekten sürüde kaç kaz vardır?

- A) 33
- B) 34
- C) 35
- D) 36
- E) 37

ÇÖZÜM

Bir sürü artı bir sürü artı yarım sürü artı yarımın yarısı sürü toplamda 2 tam $\frac{3}{4}$ sürü eder. Bunu $1\frac{1}{4}$ sürü olarak da yazabiliriz. Yerdeki kaz da eklendiğinde 100 kaz olduğuna göre, $1\frac{1}{4}$ sürü 99 kaza eşittir. Bu da sürüde 36 kaz olduğunu gösterir.

CEVAP: D**SORU 6**

Elinde 6 adet 10cm uzunluğunda çubuk olan bir kişi, bu çubuklarla düz bir zeminde **kapalı** bir alan oluşturmak istemektedir. Bu alan en fazla kaç santimetrekare olabilir?

- A) $25\sqrt{3}$
- B) $50\sqrt{3}$
- C) $75\sqrt{3}$
- D) $150\sqrt{3}$
- E) $250\sqrt{3}$

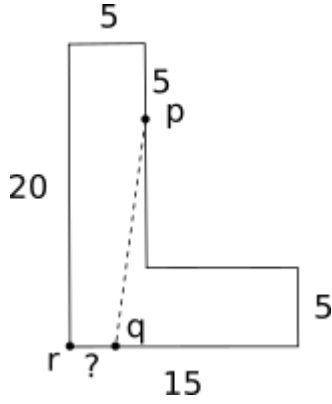
ÇÖZÜM

En fazla alan 6 çubuk bir düzgün altıgenin kenarları olacak şekilde dizildiğinde elde edilir. Bu durumda altıgen 6 adet eşkenar üçgenden oluşur. Her bir eşkenar üçgenin alanı, kenar uzunluğu 10cm olduğundan, $25\sqrt{3}$ olarak hesaplanır. 6 üçgen için bu $150\sqrt{3}$ olarak bulunur.

CEVAP: D

SORU 7

Aşağıdaki şekilde verilen uzun kenarı 20cm, kısa kenarı 15cm olan L şeklindeki tahta p ve q noktalarını birleştiren bir doğru parçası boyunca kesilecektir. Ortaya çıkan iki parça eşit alana sahip olacağına göre q ve r noktaları arasındaki mesafe ne kadar olmalıdır?



- A) 5
- B) 3
- C) $5/3$
- D) $3/5$
- E) 0

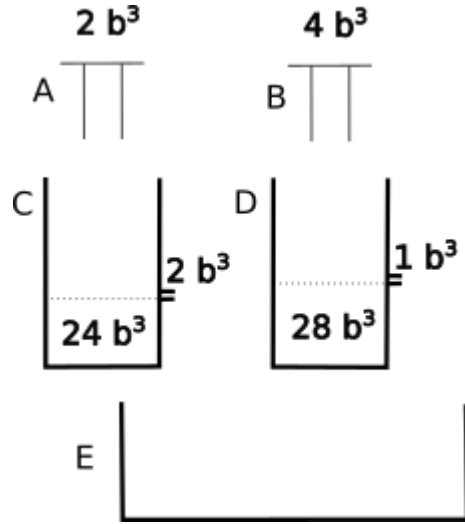
ÇÖZÜM

Şeklin toplam alanı $5 \cdot 20 + 10 \cdot 5 = 150$ cm karedir. Yarısı 75 olacaktır. L nin ucundaki $5 \cdot 5$ lik alanı çıkartırsak pqr noktalarını içeren yamuğun alanı 50cmkare olmalıdır. Bu alanın formülü: $(5+x)/2 \cdot 15$ tir. Yani x $5/3$ olarak bulunur.

CEVAP: C

SORU 8

Aşağıdaki düzenekte A ve B musluklarından sırasıyla C ve D kaplarına saniye 2 ve 4 birim küp su akmaktadır. C kabında 24 birimküp seviyesinde, D kabında ise 28 birim küp seviyesinde bir delik bulunmaktadır. C kabındaki delik saniyede 2 birim küp su boşaltmakta, D kabındaki ise saniyede 1 birim küp su boşaltmaktadır. Bu deliklerden akan sular da E kabında birikmektedir. Tüm kaplar boş iken, aynı anda A ve B muslukları açılırsa, kaç saniye sonra E kabında eşit miktarda A ve B musluklarından gelen su bulunur?



- A) 5
- B) 7
- C) 12
- D) 15
- E) 17

ÇÖZÜM

C kabı 12 saniye, D kabı ise 7 saniye sonra akıtmaya başlar. Yani 12. saniyeye gelindiğinde E kabında $5 \cdot 1 = 5$ birimküp B musluğundan gelen su birikmiş olur. Bir x saniye daha geçtiğinde kaptaki su miktarı $2x$ (A'dan gelen) + $5+x$ (B'den gelen) şeklinde olur. Bu durumda eşit olmaları için $2x = 5+x$ denkleminde $x = 5$ bulunur. Yani $12+5 = 17$. saniyede farklı musluklardan gelen su miktarı eşitlenir.

CEVAP: E

SORU 9

Bir kitap satıcısının elinde sınırsız sayıda farklı kitap olduğunu varsayalım. Bu kitap satıcısı insanların daha fazla kitap okuyabilmesi için bir kampanya başlatmıştır. Kitap satıcısı, okuyup satıcıya vereceğiniz her 7 kitap için size elinizdekenden farklı 1 kitabı ücretsiz olarak vermektedir. Bu kitapları da okuduktan sonra tekrar kitap satıcısına verebilmektesiniz. Başlangıçta elinizde 121 farklı kitabınız var ise, bu kitapları da dahil ederek ek bir ücret ödmeden toplamda okuyabileceğiniz farklı kitap sayısı en fazla kaç olur?

- A) 141
- B) 142
- C) 143
- D) 144
- E) 145

ÇÖZÜM

Öncelikle kitap sayısı 121'dir. Bu kitaplardan 119'u satıcıya verildiğinde size $119 / 7 = 17$ yeni kitap geri verecektir. Ancak $121 \% 7 = 2$ kitap daha önce kullanılmadığı için elde $17 + 2 = 19$ kitap vardır. Bu kitaplardan da okunmayanlar okunduktan sonra 14 kitap satıcıya verildiğinde satıcı $14 / 7 = 2$ yeni kitap geri verecektir. Eldeki $19 \% 7 = 5$ kitap ile birlikte kampanyada kullanılabilecek $2 + 5 = 7$ toplamda kitap vardır. Bu kitaplardan da okunmayanlar okunduktan sonra 7 kitap satıcıya verilir ve $7 / 7 = 1$ yeni kitap alınır. Bundan sonra başka yeni kitap almak mümkün değildir.
 $121 + 17 + 2 + 1 = 141$ kitap okunabilir.

CEVAP: A

SORU 10

Bir pazarcı satmakta olduğu elmalar için bir kampanya yapmıştır. Pazarcı 1 kilo elmayı 2 TL'ye satmaktadır ancak elmalar 5 kilo olarak alındığında bunun ücreti 8 TL'dir. Toplamda 58 kilo elma almanız gerekmektedir. Daha hesaplı olması durumunda fazla elma almanızda bir sakınca yoktur. Buna göre 58 kilo elmayı en az kaç TL ödeyerek satıcıdan alabilirsiniz?

- A) 92
- B) 93
- C) 94
- D) 95
- E) 96

ÇÖZÜM

5 kilo elmayı kampanya ile alırsak 8 TL olmaktadır. Ancak 1 kilo elma 2TL olduğundan kampanyasız alırsak $5 \times 2 = 10$ TL olmaktadır. Bu yüzden 5'er kilo alabildiğimiz tüm durumlarda kampanyalı fiyatı tercih etmemiz gerekir.

55 kiloyu kampanyalı fiyattan alırsak $55 / 5 = 11$ olacaktır. $11 \times 8 = 88$ TL ücret olacaktır. Kalan 3 kilo elma için ise iki seçeneğe de bakılmalıdır.

5 kiloluk kampanya ile alırsak 8 TL ücreti olacaktır. 1 kilo x 3 şeklinde alınırsa $= 2 \text{ TL} \times 3 = 6 \text{ TL}$ ücreti olacaktır. O yüzden kalan 3 kiloyu kampanyasız fiyattan almak daha ucuz olacaktır.

$88 \text{ TL} + 6 \text{ TL} = 94 \text{ TL}$ ödenecek en uygun fiyattır.

CEVAP: C

SORU 11

Ahmet ve Zeynep bir sayı tahmini oyunu oynamaya karar verirler. Zeynep aklından 1 ile 100 dahil olacak şekilde bu sayılar arasında bir sayı tutacaktır. Ahmet Zeynep'in aklından tuttuğu sayının bu aralıkta olduğunu bilmektedir. Ahmet tahminde bulunduğunda Zeynep Ahmet'e sadece aklından tuttuğu sayının Ahmet'in söylediği sayıdan daha küçük olduğunu, daha büyük olduğunu ya da tahmin ettiği sayı olduğunu söyleyebilir. Oyunda yanlış bilgi verilmemektedir. Zeynep, Ahmet'e kaç tahmin hakkı istediğini sorar. Ahmet sayıyı tahmin etmek için en etkili yöntemi bulmuştur ve kaç tahmin hakkı isterse her koşulda Zeynep'in aklında tuttuğu sayıyı kesin olarak tahmin edebilir?

- A) 6
- B) 7
- C) 8
- D) 9
- E) 10

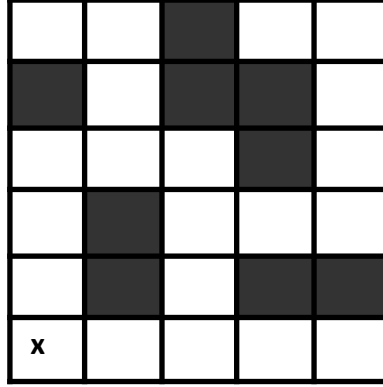
ÇÖZÜM

Ahmet'in elinde sadece Zeynep'in aklındaki sayının tahminden daha küçük, daha büyük ve tahmine eşit olduğu bilgisi vardır bu yüzden en akıllı strateji her zaman tam ortadaki sayıyı söylemektir. Böylece Ahmet'in tüm tahminleri yanlış olduğunda bile alacağı sonuç şöyle olacaktır.

1. 50 sayısını tahmin ettiğinde daha küçük ya da daha büyük bilgisi fark etmeksizin 50 sayıdan kurtulmuş ve geriye 50 sayı kalmıştır.
2. Kalan 50 sayı içerisinde ortadaki sayıyı söylediğinde tahmini yanlış olsa da 25 sayıyı daha eleyecektir.
3. Kalan 25 sayı içerisinde ortadaki sayıyı söylediğinde tahmini yanlış olsa da 13 sayıyı daha eleyecektir.
4. Kalan 12 sayı içerisinde ortadaki sayıyı söylediğinde tahmini yanlış olsa da 6 sayıyı daha eleyecektir.
5. Kalan 6 sayı içerisinde ortadaki sayıyı söylediğinde tahmini yanlış olsa da 3 sayıyı daha eleyecektir.
6. Kalan 3 sayı içerisinde ortadaki sayıyı söylediğinde tahmini yanlış olsa da 2 sayıyı daha eleyecektir.
7. Kalan son sayı mecburen Zeynep'in aklından tuttuğu sayı olacaktır.

Böylece 7 tahmin hakkı verilirse Ahmet tüm sayıları doğru tahmin edecektir.

CEVAP: B

[12-13 soruları için açıklama]

Yukarıdaki Can'ın odasının haritası verilmiştir. Siyah ile gösterilen kareler engelleri, beyaz ile gösterilen kareler boş alanları temsil etmektedir. Can boş alanları temizlemek için bir robot süpürge tasarlamış ve robot süpürgesinin algoritmasını aşağıdaki şekilde oluşturmuştur:

- Robot süpürge sadece 1 birim yukarı, aşağı, sola veya sağa gidebilmekte ancak çapraz gidememektedir.
- Robot her bir birim için o birimin kaç kez ziyaret edildiği bilgisini hafızasında saklar ve başlangıçta tüm birimlerin ziyaret değeri sıfırdır.
- Robot süpürge öncelikle gidebileceği birimler içerisinde hiç ziyaret edilmemiş bir birime ilerler ve o birimin ziyaret sayısını 1 artırır ve o birimi temizler, eğer birden fazla ziyaret edilmemiş birim varsa:
 - o Öncelik sırası öncelikliden başlayarak: *Yukarı > Sağ > Sol > Aşağı* şeklindedir.
- Eğer robotun gidebileceği tüm birimler daha önceden ziyaret edilmiş birimlerse, robot daha önce en az sayıda ziyaret ettiği birimi tercih eder ve o birimin ziyaret sayısını 1 artırır. Burada eşitlik olması durumunda:
 - o Öncelik sırası öncelikliden başlayarak: *Aşağı > Sol > Sağ > Yukarı* şeklindedir.
- Süpürge önce başlatıldığı birimi temizlemekte ve o birimin ziyaret sayısını bir artırmaktadır.
- Tüm kareler en az 1 kez ziyaret edildiğinde robot süpürge temizliği bitirmektedir.

SORU 12

Can robot süpürgesi temizliği bitirdikten sonra algoritmasının verimliliğini ölçmek istemektedir. Buna göre Can robot süpürgeyi en sol sütunda en alttaki (x ile işaretli olan) birimden çalışmaya başlattığında, Can'ın süpürgesinin algoritmasına göre temizlik bittiğinde tüm karelerin ziyaret sayıları toplamı kaç olur?

- A) 30
B) 31
C) 32
D) 33
E) 34

ÇÖZÜM

<table border="1"> <tr><td>1</td><td>2</td><td></td><td></td><td></td></tr> <tr><td></td><td>2</td><td></td><td></td><td></td></tr> <tr><td>1</td><td>2</td><td>1</td><td></td><td>1</td></tr> <tr><td>1</td><td></td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> </table>	1	2					2				1	2	1		1	1		1	1	1	1					1					<table border="1"> <tr><td>1</td><td>2</td><td></td><td></td><td></td></tr> <tr><td></td><td>2</td><td></td><td></td><td>1</td></tr> <tr><td>1</td><td>2</td><td>1</td><td></td><td>1</td></tr> <tr><td>1</td><td></td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> </table>	1	2					2			1	1	2	1		1	1		1	1	1	1					1					<table border="1"> <tr><td>1</td><td>2</td><td></td><td></td><td>1</td></tr> <tr><td></td><td>2</td><td></td><td></td><td>1</td></tr> <tr><td>1</td><td>2</td><td>1</td><td></td><td>1</td></tr> <tr><td>1</td><td></td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> </table>	1	2			1		2			1	1	2	1		1	1		1	1	1	1					1					<table border="1"> <tr><td>1</td><td>2</td><td></td><td>1</td><td>1</td></tr> <tr><td></td><td>2</td><td></td><td></td><td>1</td></tr> <tr><td>1</td><td>2</td><td>1</td><td></td><td>1</td></tr> <tr><td>1</td><td></td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> </table>	1	2		1	1		2			1	1	2	1		1	1		1	1	1	1					1					<table border="1"> <tr><td>1</td><td>2</td><td></td><td>1</td><td>2</td></tr> <tr><td></td><td>2</td><td></td><td></td><td>1</td></tr> <tr><td>1</td><td>2</td><td>1</td><td></td><td>1</td></tr> <tr><td>1</td><td></td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> </table>	1	2		1	2		2			1	1	2	1		1	1		1	1	1	1					1				
1	2																																																																																																																																																									
	2																																																																																																																																																									
1	2	1		1																																																																																																																																																						
1		1	1	1																																																																																																																																																						
1																																																																																																																																																										
1																																																																																																																																																										
1	2																																																																																																																																																									
	2			1																																																																																																																																																						
1	2	1		1																																																																																																																																																						
1		1	1	1																																																																																																																																																						
1																																																																																																																																																										
1																																																																																																																																																										
1	2			1																																																																																																																																																						
	2			1																																																																																																																																																						
1	2	1		1																																																																																																																																																						
1		1	1	1																																																																																																																																																						
1																																																																																																																																																										
1																																																																																																																																																										
1	2		1	1																																																																																																																																																						
	2			1																																																																																																																																																						
1	2	1		1																																																																																																																																																						
1		1	1	1																																																																																																																																																						
1																																																																																																																																																										
1																																																																																																																																																										
1	2		1	2																																																																																																																																																						
	2			1																																																																																																																																																						
1	2	1		1																																																																																																																																																						
1		1	1	1																																																																																																																																																						
1																																																																																																																																																										
1																																																																																																																																																										
<table border="1"> <tr><td>1</td><td>2</td><td></td><td>1</td><td>2</td></tr> <tr><td></td><td>2</td><td></td><td></td><td>2</td></tr> <tr><td>1</td><td>2</td><td>1</td><td></td><td>1</td></tr> <tr><td>1</td><td></td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> </table>	1	2		1	2		2			2	1	2	1		1	1		1	1	1	1					1					<table border="1"> <tr><td>1</td><td>2</td><td></td><td>1</td><td>2</td></tr> <tr><td></td><td>2</td><td></td><td></td><td>2</td></tr> <tr><td>1</td><td>2</td><td>1</td><td></td><td>2</td></tr> <tr><td>1</td><td></td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> </table>	1	2		1	2		2			2	1	2	1		2	1		1	1	1	1					1					<table border="1"> <tr><td>1</td><td>2</td><td></td><td>1</td><td>2</td></tr> <tr><td></td><td>2</td><td></td><td></td><td>2</td></tr> <tr><td>1</td><td>2</td><td>1</td><td></td><td>2</td></tr> <tr><td>1</td><td></td><td>1</td><td>1</td><td>2</td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> </table>	1	2		1	2		2			2	1	2	1		2	1		1	1	2	1					1					<table border="1"> <tr><td>1</td><td>2</td><td></td><td>1</td><td>2</td></tr> <tr><td></td><td>2</td><td></td><td></td><td>2</td></tr> <tr><td>1</td><td>2</td><td>1</td><td></td><td>2</td></tr> <tr><td>1</td><td></td><td>1</td><td>2</td><td>2</td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> </table>	1	2		1	2		2			2	1	2	1		2	1		1	2	2	1					1					<table border="1"> <tr><td>1</td><td>2</td><td></td><td>1</td><td>2</td></tr> <tr><td></td><td>2</td><td></td><td></td><td>2</td></tr> <tr><td>1</td><td>2</td><td>1</td><td></td><td>2</td></tr> <tr><td>1</td><td></td><td>2</td><td>2</td><td>2</td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> </table>	1	2		1	2		2			2	1	2	1		2	1		2	2	2	1					1				
1	2		1	2																																																																																																																																																						
	2			2																																																																																																																																																						
1	2	1		1																																																																																																																																																						
1		1	1	1																																																																																																																																																						
1																																																																																																																																																										
1																																																																																																																																																										
1	2		1	2																																																																																																																																																						
	2			2																																																																																																																																																						
1	2	1		2																																																																																																																																																						
1		1	1	1																																																																																																																																																						
1																																																																																																																																																										
1																																																																																																																																																										
1	2		1	2																																																																																																																																																						
	2			2																																																																																																																																																						
1	2	1		2																																																																																																																																																						
1		1	1	2																																																																																																																																																						
1																																																																																																																																																										
1																																																																																																																																																										
1	2		1	2																																																																																																																																																						
	2			2																																																																																																																																																						
1	2	1		2																																																																																																																																																						
1		1	2	2																																																																																																																																																						
1																																																																																																																																																										
1																																																																																																																																																										
1	2		1	2																																																																																																																																																						
	2			2																																																																																																																																																						
1	2	1		2																																																																																																																																																						
1		2	2	2																																																																																																																																																						
1																																																																																																																																																										
1																																																																																																																																																										
<table border="1"> <tr><td>1</td><td>2</td><td></td><td>1</td><td>2</td></tr> <tr><td></td><td>2</td><td></td><td></td><td>2</td></tr> <tr><td>1</td><td>2</td><td>1</td><td></td><td>2</td></tr> <tr><td>1</td><td></td><td>2</td><td>2</td><td>2</td></tr> <tr><td>1</td><td></td><td>1</td><td></td><td></td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td></tr> </table>	1	2		1	2		2			2	1	2	1		2	1		2	2	2	1		1			1					<table border="1"> <tr><td>1</td><td>2</td><td></td><td>1</td><td>2</td></tr> <tr><td></td><td>2</td><td></td><td></td><td>2</td></tr> <tr><td>1</td><td>2</td><td>1</td><td></td><td>2</td></tr> <tr><td>1</td><td></td><td>2</td><td>2</td><td>2</td></tr> <tr><td>1</td><td></td><td>1</td><td></td><td></td></tr> <tr><td>1</td><td></td><td>1</td><td></td><td></td></tr> </table>	1	2		1	2		2			2	1	2	1		2	1		2	2	2	1		1			1		1																																																																																															
1	2		1	2																																																																																																																																																						
	2			2																																																																																																																																																						
1	2	1		2																																																																																																																																																						
1		2	2	2																																																																																																																																																						
1		1																																																																																																																																																								
1																																																																																																																																																										
1	2		1	2																																																																																																																																																						
	2			2																																																																																																																																																						
1	2	1		2																																																																																																																																																						
1		2	2	2																																																																																																																																																						
1		1																																																																																																																																																								
1		1																																																																																																																																																								
<table border="1"> <tr><td>1</td><td>2</td><td></td><td>1</td><td>2</td></tr> <tr><td></td><td>2</td><td></td><td></td><td>2</td></tr> <tr><td>1</td><td>2</td><td>1</td><td></td><td>2</td></tr> <tr><td>1</td><td></td><td>2</td><td>2</td><td>2</td></tr> <tr><td>1</td><td></td><td>1</td><td></td><td></td></tr> <tr><td>1</td><td></td><td>2</td><td>2</td><td>1</td></tr> </table>	1	2		1	2		2			2	1	2	1		2	1		2	2	2	1		1			1		2	2	1	<table border="1"> <tr><td>1</td><td>2</td><td></td><td>1</td><td>2</td></tr> <tr><td></td><td>2</td><td></td><td></td><td>2</td></tr> <tr><td>1</td><td>2</td><td>1</td><td></td><td>2</td></tr> <tr><td>1</td><td></td><td>2</td><td>2</td><td>2</td></tr> <tr><td>1</td><td></td><td>1</td><td></td><td></td></tr> <tr><td>1</td><td>1</td><td>2</td><td>2</td><td>1</td></tr> </table>	1	2		1	2		2			2	1	2	1		2	1		2	2	2	1		1			1	1	2	2	1																																																																																													
1	2		1	2																																																																																																																																																						
	2			2																																																																																																																																																						
1	2	1		2																																																																																																																																																						
1		2	2	2																																																																																																																																																						
1		1																																																																																																																																																								
1		2	2	1																																																																																																																																																						
1	2		1	2																																																																																																																																																						
	2			2																																																																																																																																																						
1	2	1		2																																																																																																																																																						
1		2	2	2																																																																																																																																																						
1		1																																																																																																																																																								
1	1	2	2	1																																																																																																																																																						

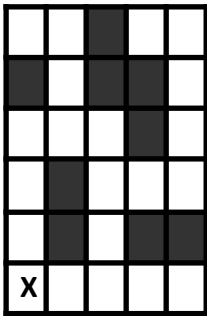
Toplamda ziyaret değeri 32 olacaktır.

CEVAP: C

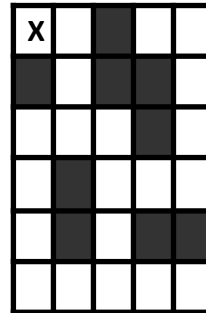
SORU 13

Can'ın algoritmasına göre süpürgeler tüm birimler ziyaret edilmemiş şekilde ve işaretli birimlerden temizliğe başlayıp ilgili birimin ziyaret edilme sayısını bir artırarak temizliğe başladığında, Can robot süpürgeyi hangi birimden başlatırsa temizlik bittiğinde birimlerin toplam ziyaret edilme sayısı en fazla olur?

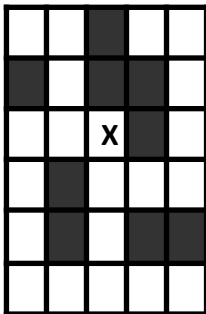
A)



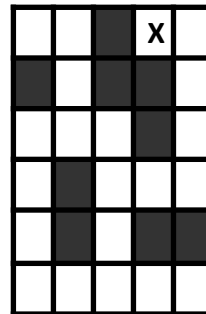
B)



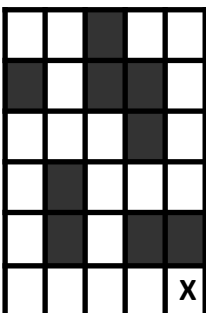
C)



D)



E)



ÇÖZÜM

1	2		1	2
	2			2
1	2	1		2
1		2	2	2
1		1		
1	1	2	2	1

A) 32

1	1		1	2
	1			2
1	1	1		2
1		2	2	2
1		1		
1	1	2	2	1

B) 29

1	2		1	2
	2			2
1	2	1		2
1		2	2	2
1		2		
1	1	2	1	1

C) 32

1	2		1	1
	2			1
1	2	1		1
1		1	1	1
1		1		
1	1	2	1	1

D) 25

1	2		1	1
	2			1
2	2	1		1
3		2	1	1
3		2		
3	2	3	3	2

E) 38

CEVAP: E

[14-15 soruları için açıklama]

Bir okulda 110 öğrenci İngilizce, Fransızca ve Almanca dillerinden en az birini öğrenmektedir. Size aşağıdaki bilgiler veriliyor:

- 65 öğrenci İngilizce öğreniyor.
- 40 öğrenci Almanca öğreniyor.
- 25 öğrenci hem İngilizce hem Fransızca öğreniyor.
- 15 öğrenci hem İngilizce hem Almanca öğreniyor.
- 10 öğrenci hem Fransızca hem Almanca öğreniyor.
- 5 öğrenci üç dili birden öğreniyor.

SORU 14

Sadece İngilizce öğrenen kaç öğrenci vardır?

- A) 25
- B) 30
- C) 35
- D) 40
- E) 45

ÇÖZÜM

5 öğrenci 3 dili birden öğrendiğine göre sadece İngilizce ve Fransızca öğrenen öğrenci sayısı $25 - 5 = 20$ 'dir. Benzer şekilde sadece İngilizce ve Almanca öğrenen öğrenci sayısı $15 - 5 = 10$ 'dur. Öyleyse sadece İngilizce öğrenen öğrenci sayısı $65 - 20 - 10 - 5 = 30$ 'dur.

CEVAP: B**SORU 15**

Aşağıdakilerden hangisi yanlıştır?

- A) Almanca öğrenenlerin sayısı en az iki dil öğrenenlerin sayısını eşittir
- B) Sadece Almanca öğrenenlerin sayısı sadece Fransızca öğrenenlerin sayısına eşittir
- C) Fransızca öğrenenlerin çoğu başka bir dil daha öğrenmektedir
- D) Almanca öğrenmeyenlerin çoğu Fransızca öğrenmektedir
- E) Fransızca veya Almanca öğrenenlerin yarısı İngilizce öğrenmektedir

ÇÖZÜM

30 öğrenci sadece İngilizce, 20 öğrenci sadece Fransızca, 20 öğrenci sadece Almanca öğrenmektedir. Sadece İngilizce ve Almanca öğrenen 10, sadece İngilizce ve Fransızca öğrenen 20, sadece Almanca ve Fransızca öğrenen 5 öğrenci vardır. 5 öğrenci ise 3 dili de öğrenmektedir. Bu bilgilere göre Fransızca veya Almanca öğrenen $20 + 20 + 10 + 20 + 5 + 5 = 80$ öğrenci vardır. Bu öğrencilerden $65 - 30 = 35$ tanesi İngilizce öğrenmektedir. Bu yüzden E şıkında verilen ifade yanlıştır. Diğer ifadelerin tamamı doğrudur.

CEVAP: E

SORU 16

Bir saat, her 15 dakikada bir özel bir sinyal veriyor. Sabah saat 9:00'da başlayan bir deneyde, bu saat toplam 250 sinyal veriyor. Deney sona erdiğinde saat kaç olur?

- A) 7:15
- B) 9:45
- C) 12:15
- D) 19:30
- E) 23:15

ÇÖZÜM

İlk ve son sinyal arasında 249 aralık vardır. Deney $249 \times 15 = 3735$ dakika sürmüştür. Bu da 62 saat 15 dakika eder. Bu süreden 2 günü (48 saat) çıkartırsak saatimiz 14 saat 15 dakika ilerlemiştir. Deney bittiğinde saat 23:15'dir.

CEVAP: E**SORU 17**

Bir tiyatro sahnesinde 5 kadın ve 3 erkek sırayla dizilecektir. İki erkek yanyana gelmeyecek şekilde kaç farklı dizilim gerçekleştirilebilir?

- A) 2400
- B) 14400
- C) 19200
- D) 28800
- E) 40320

ÇÖZÜM

5 kadın kendi aralarında $5! = 120$ şekilde dizilir. Kadınlar dizildikten sonra başta 1 boşluk, arada 4 boşluk ve sonda 1 boşluk olmak üzere toplam 6 boşluk olur. Erkekler bu 6 boşluktan 3 tanesine yerleşecekleridir. Erkekler boşluklara $6! / 3! = 120$ şekilde yerleştirilebilir. Bu yüzden cevap $120 \times 120 = 14400$ 'tür.

CEVAP: B

[18-19 soruları için açıklama]

Ayşe, Berk, Can ve Derya adlı 4 çocuk koşu yarışması yapmaya karar veriyorlar. Yarışmanın sonucuna ilişkin çocuklar aşağıdaki tahminlerde bulunuyorlar.

- Ayşe diyor ki: "Ben Can'dan önce geleceğim."
- Berk diyor ki: "Derya birinci olacak."
- Can diyor ki: "Ben üçüncü olacağım."
- Derya diyor ki: "Berk benden daha kötü bir sırada olacak."

SORU 18

Bütün çocuklar doğru tahminde bulundularsa yarışı kim son sırada tamamlamıştır?

- A) Ayşe
- B) Berk
- C) Can
- D) Derya
- E) Tahminler arasında çelişki vardır

ÇÖZÜM

Tahminler arasında bir çelişki yoktur ve olarası tek sıralama DACB şeklindedir. Yarışı sor sırada tamamlayan Berk'tir.

CEVAP: B**SORU 19**

Çocukların yaptığı 4 tahminden sadece 1 tanesinin yanlış olduğunu varsayalım. Yarışmanın sonucunda çocuklar kaç farklı şekilde sıralanmış olabilirler?

- A) 3
- B) 4
- C) 5
- D) 6
- E) 8

ÇÖZÜM

Ayşe'nin tahmini yanlışsa sıralama DBCA şeklinde olmak zorundadır. Berk'in tahmini yanlışsa sıralama ADCB şeklinde olmak zorundadır. Can'ın tahmini yanlışsa sıralama DBAC ve DABC şeklinde olabilir. Derya'nın tahminin yanlış olduğunu varsayalım. Berk Derya'dan daha iyi sırada olmak zorundadır. Ancak Berk'in doğru tahmininden dolayı da Derya birinci olmak zorundadır. Bu da bir çelişkidir. Bu yüzden olası 4 durum vardır.

CEVAP: B

SORU 20

5 sarı ve 6 yeşil top bulunan bir torbadan rastgele 3 top çekiliyor. Tam olarak 1 sarı top çekme olasılığı nedir?

- A) 5/11
- B) 6/11
- C) 5/22
- D) 10/33
- E) 3/11

ÇÖZÜM

11 toptan 3 top 165 şekilde seçilebilir. Bir sarı top 5 şekilde ve 2 yeşil top 15 şekilde seçilebilir. Bu yüzden cevap $5 \times \frac{15}{165} = \frac{5}{11}$ 'dir.

CEVAP: A

[Soru 21-30 İçin Açıklama]

- Soruları C programlama dili çerçevesinde cevaplayınız.
- Derleyici olarak gcc kullanıldığını varsayınız.
- Gerekli tüm başlık (header) dosyalarının verilen programa dahil edildiğini varsayınız.

SORU 21

```
int main() {
    int x = 12, y = 6;
    x = (--x) / (y++) * (++y) - x % y;
    printf("%d", x);
    return 0;
}
```

Yukarıdaki kod parçasığı çalıştırıldığında ekrana yazdığı değer ne olur?

- A) 0
- B) 4
- C) 5
- D) 11
- E) 15

ÇÖZÜM

Öncelikle $x = 12$ ve $y = 6$ olarak tanımlanmıştır. Sonraki satırda öncelikli olarak $(--x) / (y++)$ işlemi gerçekleşecektir ancak bölme işleminden önce parantez içlerindeki $-$ ve $++$ işlemleri gerçekleşir. $(--x)$ olduğundan x 'in değeri 1 azaltılarak 11 olarak işleme alınır. $(y++)$ işlemi ise y değerini 6 olarak kullandıktan sonra değeri 1 artırarak 7 yapmış olur. Bu yüzden işlem $11/6$ olarak yapılır ve tam sayı bölme işlemi C programlama dilinde kalansız olarak yapıldığı için bu işlemin sonucu 1 olur.

İşlemin devamında önceki kısmın devamı olan $1 * (++y)$ işlemi yapılacaktır. Yine işlem önceliği parantez içerisinde olacaktır. Bir önceki işlemde y değeri 7 olmuştur ancak burada $++$ işareti y 'nin önünde olduğundan değerin artırılması işlemden önce yapılacaktır. Bu yüzden $1 * 8 = 8$ olacaktır.

Sonrasında $8 - x \% y$ işlemi kalmıştır. Burada $\%$ işleminin önceliği vardır. Bu aşamadaki x değeri 11 y değeri ise 8 dir o yüzden $x \% y$ işlemi sonucu $11 \bmod 8 = 3$ olacaktır.

Son aşamada $8 - 3 = 5$ olacaktır. İşlem sonucu 5 olmalıdır.

CEVAP: C

SORU 22

```
int main() {  
    int i, j;  
    for (i = 1; i <= 5; i++) {  
        for (j = 1; j <= 5; j++) {  
            if (i + j == 7) {  
                break;  
            }  
            else if(i + j == 4){  
                continue;  
            }  
            else if(j == i){  
                i++;  
            }  
            printf("%d ", i * j);  
        }  
    }  
}
```

Yukarıdaki kod parçacığı çalıştırıldığında programın çıktısı aşağıdakilerden hangisi olur?

- A) 1 2 8 10 9 16 25
- B) 2 2 8 10 9 16 25
- C) 2 4 8 6 12 20 30
- D) 2 6 8 6 12 20 30
- E) 2 6 8 10 12 20 30

ÇÖZÜM

i: 1, j:1 değeri için else if(j == i) bloğu çalışır ve i 1 artar $2*1 = 2$ olur, 2 değeri ekrana yazar.

i değeri 2 olmuş olur içteki döngüde bittiği için j değeri de 2 olur.

i: 2, j:2 için else if (i + j == 4) bloğu çalışır ve program içteki döngünün 1 iterasyonunu pas geçerek j'yi 1 artırır ve 3 yapar. Tekrar içteki döngüye devam eder.

i: 2, j:3 için hiçbir if bloğu çalışmaz ve $2*3 = 6$ olduğundan ekrana 6 yazar. j 1 artırılarak içteki döngü başa döner.

i: 2, j:4 için hiçbir if bloğu çalışmaz ve $2*4 = 8$ olduğundan ekrana 8 yazar. j 1 artırılarak içteki döngü başa döner.

i: 2, j:5 için if (i + j == 7) bloğu çalıştığı için içteki döngüyü kırarak çıkar ve dıştaki döngüye devam ederek i değeri 1 artırılır ve daha sonra program tekrar içteki döngüye i değeri 1 için başlar.

i: 3, j:1 --> için else if(i + j == 4) bloğu çalışır ve program içteki döngünün 1 iterasyonunu pas geçerek j'yi 1 artırır ve 2 yapar. Tekrar içteki döngüye devam eder.

i: 3, j:2 --> 6 için hiçbir if bloğu çalışmaz ve $3*2 = 6$ olduğundan ekrana 6 yazar. j 1 artırılarak içteki döngü başa döner.

i: 3, j:3 --> değeri için else if(j == i) bloğu çalışır ve i 1 artar $4*3 = 12$ olur, 12 değeri ekrana yazar. Bir sonraki iterasyonda j 4 olur.

i: 4, j:4 --> değeri için else if(j == i) bloğu çalışır ve i 1 artar $5*4 = 20$ olur, 20 değeri ekrana yazar. Bir sonraki iterasyonda j 5 olur.

i: 5, j:5 --> değeri için else if(j == i) bloğu çalışır ve i 1 artar $6*5 = 30$ olur, 30 değeri ekrana yazar. Bir sonraki iterasyonda j 6 olur. İçteki döngü sonlandırılır. Daha sonrasında i değeri de 7 olur ve dıştaki döngüde sonlanmış olur.

CEVAP: D

SORU 23

```

int f1(int n) {
    if (n == 0)
        return 1;
    return n * f1(n - 1);
}
int f2(int n) {
    if (n <= 1)
        return 1;
    return f1(n / 2) - f2(n - 2) * (n/3);
}
int main() {
    int result = f2(9);
    printf("%d", result);
    return 0;
}

```

Yukarıdaki kod parçası hangi çıktıyı üretir?

- A) 12
- B) 24
- C) 56
- D) 6
- E) 18

ÇÖZÜM

f1 fonksiyonu faktöriyel hesabı yapmaktadır, f2(9) çağırısı yapıldığında, adımlar şu şekilde işler:

$$f2(9) = f1(9 / 2) - f2(9 - 2) * (9/3) = f1(4) - f2(7) * 3$$

$$f1(4) = 24$$

$$f2(9) = 24 - f2(7) * 3 \text{ (bulmamız gereken sonuç bu denklemdir.)}$$

$$f2(7) = f1(7 / 2) - f2(7 - 2) * (7/3) = f1(3) - f2(5) * 2$$

$$f1(3) = 6$$

$$f2(7) = 6 - f2(5) * 2$$

$$f2(5) = f1(5 / 2) - f2(5 - 2) * (5/3) = f1(2) - f2(3) * 1$$

$$f1(2) = 2$$

$$f2(5) = 2 - f2(3) * 1$$

$$f2(3) = f1(3 / 2) - f2(3 - 2) * (3/3) = f1(1) - f2(1) * 1$$

$$f1(1) = 1$$

$$f2(3) = 1 - f2(1) * 1$$

f2(1) = 1, yukarıya doğru eksik fonksiyon değerleri tamamlandığında

$$f2(3) = 1 - f2(1) * 1 = 1 - 1 * 1 = 0$$

$$f2(5) = 2 - f2(3) * 1 = 2 - 0 * 1 = 2$$

$$f2(7) = 6 - f2(5) * 2 = 6 - 2 * 2 = 2$$

$$f2(9) = 24 - f2(7) * 3 = 24 - 2 * 3 = 18$$

CEVAP: E

SORU 24

```

int main() {
    char *word = "olimpiyat";
    int len = (int)strlen(word);
    for(int i=1; i<=len-3; i++){
        switch (i) {
            case 1:
                printf("%c", word[len-i++]);
            case 2:
                printf("%c", word[len-i*i]);
            case 3:
                printf("%c", word[i]);
                break;
            case 4:
                printf("%c", word[len-i+1]);
                break;
            case 5:
                printf("%c", word[len-1]);
            default:
                printf("%c", word[1]);
                break;
        }
    }
    return 0;
}

```

Yukarıdaki kod parçası çalıştığında ekrana yazdırılan mesaj ne olur?

- A) tiimytll
- B) ltiimytll
- C) oapliiao
- D) apliiaoo
- E) tmytl

ÇÖZÜM

strlen(word) fonksiyonu kelimenin uzunluğunu bulur. len değeri 9 olmuş olur. Böylece döngü i=1 değeri için başlayıp i<=6 olana kadar devam eder.

switch(i) değeri ilk iterasyonda switch(1) olarak çağırılır. Case 1 çalışır ancak case içerisinde break olmadığı için case 1, 2 ve 3 çalışmış olur.

word[len-i++] = word[9-1] olur ve i değeri 2'ye yükseltilmiş olur.

word[8] kelimenin 8. harfidir ve C dilinde karakter sıraları 0 dan başladığı için word[8] aslında 9. kelimeyi göstermektedir.

word[8] = t

Case 2 için word[len-i*i] = word[9-2*2] = word[9-4] = word[5] yani kelimenin 6. harfidir.

word[5] = i

Case 3 için word[i] = word[2] yani kelimenin 3. harfidir.

word[2] = i

Daha sonra break ifadesi çalışır ve for döngüsünün ilk iterasyonu bitmiş olur. i değeri 1 artırılarak 3 olur. Böylece Case 3 çalışmış olur. Break ifadesi olduğu için sadece case 3 çalışır.

Case 3: word[i] = word[3] yani kelimenin 4. harfi

word[3] = m

Daha sonra break ifadesi çalışır ve for döngüsünün ikinci iterasyonu bitmiş olur. i değeri 1 artırılarak 4 olur. Böylece Case 4 çalışmış olur. Break ifadesi olduğu için sadece case 4 çalışır.

Case 4: word[len-i+1] = word[9-4+1] = word[6] yani kelimenin 7. harfi

word[6] = y

Daha sonra break ifadesi çalışır ve for döngüsünün üçüncü iterasyonu bitmiş olur. i değeri 1 artırılarak 5 olur. Böylece Case 5 çalışmış olur, break ifadesi olmadığı için default case aynı zamanda çalışacaktır.

Case 5: word[len-1] = word[9-1] = word[8] yani kelimenin 9. harfi

word[8] = t

Default:

word[1] = l

Daha sonra break ifadesi çalışır ve for döngüsünün üçüncü iterasyonu bitmiş olur. i değeri 1 artırılarak 6 olur. Böylece Default case çalışmış olur ve döngü sonlanır.

Default:

word[1] = l

CEVAP: A

SORU 25

```

int main() {
    int n = 9;
    for(int i=0; i<n; i++)
        printf("*");
    printf("\n");
    for(int i=1; i<n; i+=2){
        for(int j=i/2; j<n/2; j++)
            printf("*");
        for(int j=1; j<=i; j++)
            printf(" ");
        for(int j=i/2; j<n/2; j++)
            printf("*");
        printf("\n");
    }
    return 0;
}

```

Yukarıdaki kod parçası hangi çıktıyı üretir?

- A) *
- ```


```
- B) \*\*\*\*\*
- ```

*****
*****
*****
***
*

```
- C) *****
- ```

*

```
- D) \*\*\*\*\*
- ```

***** *****
***      ***
**       **
*        *

```
- E) *****
- ```

*** * ***
** * **
* * *

```

**ÇÖZÜM**

Programın başındaki for döngüsü öncelikle 9 tane yıldız aynı satıra yazdırdıktan sonra döngü sonrasındaki `printf("\n")` komutu ile bir alt satıra geçer.

\*\*\*\*\*

İkinci for döngüsü 1 ile başlayıp ikiye ikiye artacak bir i değerinin 9 a kadar gideceği şekilde ayarlanmıştır. Bu yüzden 1,3,5,7 olmak üzere 4 kez daha çalışacaktır.

**i=1 için**

Bu döngünün içerisindeki `for(int j=i/2; j<n/2;j++)`  $i/2 = 1/2$  yani 0 olacaktır.  $n/2 = 9/2$  yani 4 olacaktır. Bu yüzden 4 kez çalışacaktır ve 4 yıldız ekrana yazar.

`for(int j=1; j<=i; j++)` döngüsü i değeri ilk iterasyonda 1 olduğu için 1 kez çalışır ve 1 boşluk yazdırır.

`for(int j=i/2; j<n/2;j++)`  $i/2 = 1/2$  yani 0 olacaktır.  $n/2 = 9/2$  yani 4 olacaktır. Bu yüzden 4 kez çalışacaktır ve 4 yıldız ekrana yazar.

Daha sonrasında `printf("\n")` alt satıra geçeceğinden satır aşağıdaki şekilde oluşmuş olur ve i değeri 3'e yükselir.

\*\*\*\* \*

**i=3 için**

`for(int j=i/2; j<n/2;j++)` döngüsü  $i/2 = 3/2$  yani 1 olacaktır.  $9/2 = 9/2$  yani 4 olacaktır. Bu yüzden 3 kez çalışacaktır ve 3 yıldız ekrana yazar.

`for(int j=1; j<=i; j++)` döngüsü i 3 olduğundan j 1 olduğu için 3 kez çalışır ve 3 boşluk yazdırır.

`for(int j=i/2; j<n/2;j++)` döngüsü  $i/2 = 3/2$  yani 1 olacaktır.  $9/2 = 9/2$  yani 4 olacaktır. Bu yüzden 3 kez çalışacaktır ve 3 yıldız ekrana yazar.

Daha sonrasında `printf("\n")` alt satıra geçeceğinden satır aşağıdaki şekilde oluşmuş olur ve i değeri 5'e yükselir.

\*\*\* \*\*

**i=5 için**

`for(int j=i/2; j<n/2;j++)` döngüsü  $i/2 = 5/2$  yani 2 olacaktır.  $9/2 = 9/2$  yani 4 olacaktır. Bu yüzden 2 kez çalışacaktır ve 2 yıldız ekrana yazar.

`for(int j=1; j<=i; j++)` döngüsü i 5 olduğundan j 1 olduğu için 5 kez çalışır ve 5 boşluk yazdırır.

`for(int j=i/2; j<n/2;j++)` döngüsü  $i/2 = 5/2$  yani 2 olacaktır.  $9/2 = 9/2$  yani 4 olacaktır. Bu yüzden 2 kez çalışacaktır ve 2 yıldız ekrana yazar.

Daha sonrasında `printf("\n")` alt satıra geçeceğinden satır aşağıdaki şekilde oluşmuş olur ve i değeri 7'e yükselir.

\*\* \*\*

**i=7 için**

`for(int j=i/2; j<n/2; j++)` döngüsü  $i/2 = 7/2$  yani 3 olacaktır.  $9/2 = 9/2$  yani 4 olacaktır. Bu yüzden 1 kez çalışacaktır ve 1 yıldız ekrana yazar.

`for(int j=1; j<=i; j++)` döngüsü i 7 olduğundan j 1 olduğu için 7 kez çalışır ve 7 boşluk yazdırır.

`for(int j=i/2; j<n/2; j++)` döngüsü  $i/2 = 7/2$  yani 3 olacaktır.  $9/2 = 9/2$  yani 4 olacaktır. Bu yüzden 1 kez çalışacaktır ve 1 yıldız ekrana yazar.

Daha sonrasında `printf("\n")` alt satıra geçeceğinden satır aşağıdaki şekilde oluşmuş olur ve i değeri 7'e yükselir.

\*           \*

Daha sonrasında i değeri 9 olur ve döngü koşulunu sağlamadığı için program sonlanır. Programın çıktısı aşağıdaki gibi olur:

```

**** ****
*** ***
** **
* *

```

**CEVAP:D**

**SORU 26**

Aşağıda verilen *metod* fonksiyonu, bir sayının son hanesinin geçerli bir kontrol basamağı olup olmadığını kontrol etmeyi amaçlamaktadır. Örneğin, 1523 sayısı geçerli kontrol basamağına sahiptir. Buna göre, verilen *isle* fonksiyonu için *metod* fonksiyonunun boş bırakılan kısmına ne gelmelidir?

```
int isle(int n) {
 int m = ((n % 10 + 1) * (n / 10 + 1)) % (n / 2 + 1);
 return m > 9 ? isle(m) : m;
}

bool metod(int x) {
 return _____;
}
```

- A)  $x == isle(x)$
- B)  $x == isle(x / 10)$
- C)  $(x \% 10) == isle(x)$
- D)  $(x \% 10) == isle(x / 10)$
- E)  $isle(x \% 100) == isle(x / 10) \% 10$

**ÇÖZÜM**

Kontrol basamağı, *T.C. Kimlik Numarasında* da kullanılan ve hatalı giriş ya da yanlış veri beyanını kontrol etmeyi amaçlayan yaygın bir yöntemdir. D şıkkında,  $isle(x / 10)$  ile sayının birler basamağı dışındaki kısmı özyinelemeli *isle* fonksiyonuna verilerek tek haneli bir kontrol basamağı üretilmekte ve akabinde bu kontrol basamağının, sayının son hanesine  $(x \% 10)$  eşitliği kontrol edilmektedir. Eğer eşitlik sağlanıyorsa, sayının son hanesi geçerli bir kontrol basamağıdır. Dolayısıyla doğru cevap D şıkkıdır.

**CEVAP: D**

**SORU 27**

Aşağıda verilen kod için `metod(1234567890, 10)` çağrısının sonucu şıklardan hangisi olabilir? Verilen kodda `&x` ifadesi  $x$  değişkeninin adresini gösterir. Bu adres `metod` fonksiyonu her çağrıldığında farklı olabilir. `(long)(&x)` ifadesi ise  $x$  değişkeninin adresini bir tam sayıya çevirir.

```
int metod(long x, int i)
{
 long k = ((long)(&x) + i) % 10;

 long s = 1;
 for (int i=0; i<k && s < x / 100; i++)
 s *= 10;

 if (i > 0) {
 long z = x / s;
 long y = (z / 10 * 10) * s + (x % s) * 10 + (z % 10);
 return metod(y, i - 1);
 }
 else
 return x;
}
```

- A) 1468037295
- B) 1246803295
- C) 5194860237
- D) 3476821490
- E) 9468037215

**ÇÖZÜM**

Verilen *metod* fonksiyonu bir sayının basamaklarını karıştırma (basamakların yerinin değişmesi) işlevini görmektedir. `metod` fonksiyonu özyinelemeli olarak çalıştırılacağı için `&x` her defasında farklı bir değer üretecektir. For döngüsündeki  $s < x / 100$  koşulu yüzünden  $s$  hiçbir zaman en soldaki basamağın yerini değiştirecek kadar büyüyemez ve dolayısıyla en soldaki basamağın yeri hiçbir zaman değişmez. Bu durumu sağlayan sadece A ve B şıklarıdır. Fakat B şıkında 7 basamağı yok olmuş ve yerine 2 basamağı gelmiş böylece 2 basamağından iki adet oluşmuştur. Fakat verilen *metod* fonksiyonu basamakların sadece yerini değiştirdiği için böyle bir durum oluşamaz. Doğru cevap A şıkkıdır.

**CEVAP: A**

**SORU 28**

$A$  şehri ile  $B$  şehri arasındaki mesafe 1000 km'dir.  $A$  şehrinden  $B$  şehrine giderken en fazla 3 defa yakıt alma hakkınız bulunmaktadır. Benzin istasyonlarının  $A$  şehrine mesafeleri [400, 100, 950, 850, 600, 500, 700, 200] olarak verilmiştir.  $A$  şehrinden tam dolu depo ile yola çıkarak  $B$  şehrine ulaşmak için, aşağıdaki şıklardan hangisi gereken minimum depo kapasitesini en verimli bir şekilde bulmak için kullanılacak algoritmanın bir adımı değildir?

- A) Şehir  $B$ 'nin (1000. km) de istasyon listesine eklenmesi
- B) Benzin istasyonlarının mesafelerini küçükten büyüğe sıralamak
- C) Belirli bir kapasiteyle hedefe ulaşıp ulaşılamayacağını test eden bir fonksiyon yazmak
- D) İkili arama aralığını belirlemek için 8 benzin istasyonuna ait tüm kombinasyonları denemek
- E) İkili arama ile minimum kapasiteyi aramak

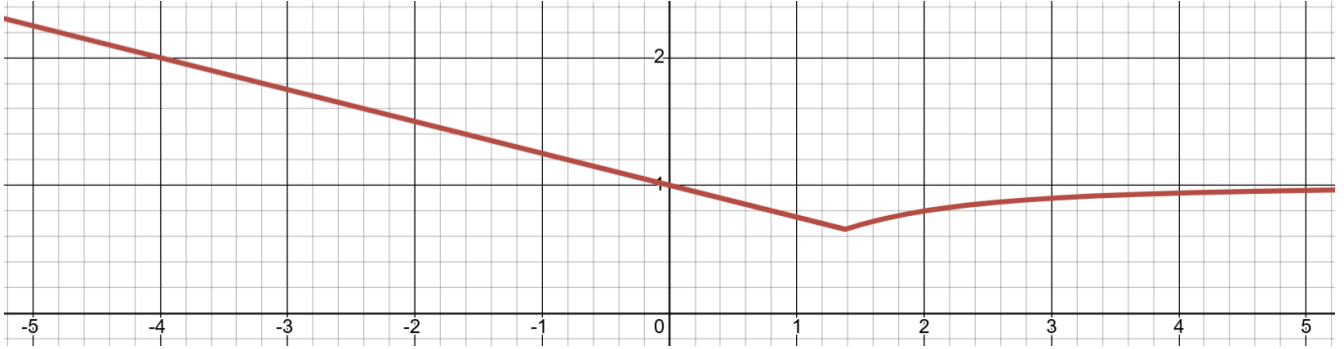
**ÇÖZÜM**

İkili arama için başlangıç değeri genellikle 1 (veya iki nokta arasındaki minimum mesafe) ve bitiş değeri toplam mesafe olan 1000 km olarak alınabilir. D şıkında yer alan "kombinasyonları deneme" yaklaşımı, verimli algoritmaların bir parçası değildir. Diğer adımlar, ikili arama temelli çözümün doğal ve gerekli bileşenleridir. Bu nedenle doğru cevap D şıkkıdır.

**CEVAP: D**

**SORU 29**

Aşağıda verilen fonksiyon için  $[A, B]$  aralığında minimum değeri bulan bir kod verilmiştir.



$[A, B] = [-5, 5]$  aralığında  $f(x) = 1 - \min(\frac{1}{x^2+1}, \frac{x}{4})$  fonksiyonun grafiği

```
double enKucukBul(double A, double B, int miter) {
 double min_val = DBL_MAX;
 double x_opt = 0.0;

 double step = (B - A) / 10.0;
 for (int iter = 0; iter < miter; iter++) {
 for (double x = A; x <= B; x += step) {
 double val = f(x);
 if (val < min_val) {
 min_val = val;
 x_opt = x;
 }
 }

 A = x_opt - step;
 B = x_opt + step;
 step /= 2.0;
 }

 return x_opt;
}
```

Verilen metodun  $A=-5$  ve  $B=5$  için çalıştırılması durumunda ifadelerden hangisi yanlıştır?

- A) Verilen kod herhangi bir fonksiyon için  $[A, B]$  aralığındaki en küçük değeri miter arttırıldıkça daha da azalan bir hata payı ile bulur
- B) miter değeri arttırıldıkça  $f(x)$ 'in en küçük değerine daha çok yaklaşılr
- C) miter=1 için  $f(x)$  fonksiyonu -5 ile 5 arasında 11 değer için hesaplanır
- D) miter=1 için enKucukBul metodu  $x_{opt}$  olarak 1 değerini döndürür
- E) miter=2 için enKucukBul metodu  $x_{opt}$  olarak 1.5 değerini döndürür

**ÇÖZÜM**

Kod, fonksiyonun  $[A, B]$  aralığındaki tüm en küçük değerleri garantili olarak bulmaz. Adım boyunun (step) yetersizliği ve örnekleme yoğunluğunun azlığı nedeniyle, ani değişimlerin olduğu bölgelerde en küçük değerler kaçırılabilir. Doğru cevap A şıkkıdır.

**CEVAP: A**

**SORU 30**

*qsort* metodu, dizileri sıralamak amacıyla kullanılan ve C programlama dilinin standart kütüphanesinde yer alan bir fonksiyondur. *qsort*(points, n, sizeof(Point), cmetod); çağırısı için points dizisi n adet Point tipinde nokta tutmaktadır. *qsort* metoduna yapılan bu çağrı için aşağıdaki ifadelerden hangisi doğrudur?

```
typedef struct {
 double x;
 double y;
} Point;

int signum(double x) {
 return (x > 0) - (x < 0);
}

int cmetod(const void* a, const void* b) {
 Point* p1 = (Point*)a;
 Point* p2 = (Point*)b;

 if (fabs(p1->y - p2->y) < 1e-8)
 return signum(p1->x - p2->x);
 else
 return p1->y < p2->y ? -1 : 1;
}
```

- A) Noktalar önce x değerine göre artan sırayla, eşit x'lerde y değerine göre artan sırayla sıralanır.
- B) Noktalar sadece x değerine göre azalan sırayla sıralanır.
- C) Noktalar önce y değerine göre artan, eşit y'lerde x değerine göre artan sırayla sıralanır.
- D) cmetod fonksiyonu sadece x değerini dikkate aldığı için y sıralaması yapılmaz.
- E) signum fonksiyonu, qsort içinde doğrudan karşılaştırma yapmak için kullanılır ve x değerlerinin sıralamasında etkilidir.

**ÇÖZÜM**

$\text{fabs}(p1 \rightarrow y - p2 \rightarrow y) < 1e-8$  ifadesi, iki noktanın y koordinatlarının eşit olup olmadığını kontrol eder (tamsayı olmadığı için  $==$  operatörü kullanılması tercih edilmez). Eğer y değerleri eşitse, sıralama x değerlerine göre yapılır. Bu durumda, signum fonksiyonu kullanılarak, ilk noktanın x değeri ikinci noktadan küçükse -1, eşitse 0, büyükse 1 döndürülür. Eğer y değerleri eşit değilse, sıralama y değerlerine göre yapılır. Bu durumda, ilk noktanın y değeri daha küçükse -1, değilse (yani daha büyükse) 1 döndürülür. y değerleri eşit olmadığı sürece 0 döndürülmesine gerek yoktur. Bu nedenle, sıralama işlemi önce y değerine göre artan şekilde, y değerleri eşit olan durumlarda ise x değerine göre artan şekilde gerçekleştirilir. Doğru cevap C şıkkıdır.

**CEVAP: C**