PEMBAHASAN OLIMPIADE SAINS 2013
TINGKAT KABUPATEN/KOTA
BIDANG INFORMATIKA/KOMPUTER

Bagian A: Aritmetika dan Logika (30 Soal)

1. Lampu yang tidak terhubung dengan sakelar adalah lampu-lampu yang bernomor bukan merupakan kelipatan 2, 3, 5, 7, maupun 11. Nomor-nomor yang memenuhi adalah 1, dan yang memiliki faktor prima di atas 11, yakni ada 6 buah:

- 1 = (*tidak memiliki faktor prima*)
- 13 = 13
- 17 = 17
- 19 = 19
- 23 = 23
- 29 = 29

Jawaban: D

---

2. Banyaknya lampu yang menyala akan maksimum apabila Sugeng menekan sakelar B, C, D, dan E. Setelah sakelar-sakelar tersebut ditekan, akan terdapat 16 lampu yang menyala. Hal ini diperoleh dengan cara coba-coba. Sebagai contoh, akan menguntungkan apabila Sugeng menekan sakelar C, D, dan E karena tidak ada lampu yang bentrok, dan seterusnya.

Jawaban: D

---

3. Dengan cara *backtracking*, yakni mencoba-coba semua cara yang mungkin, ditemukan 11 cara, yakni sebagai berikut. Huruf yang sama terdapat pada domino yang sama.

```
aabb      aabb      aabb      aabb      aabb      aabc
ccdd      ccde      cdde      cdee      cdef      ddbc
eeff      ffde      cffe      cdff      cdef      eeff

aabc      abbc      abcc      abcc      abcd
debc      addc      abdd      abde      abcd
deff      eeff      eeff      ffde      eeff
```

Jawaban: B

---

4. Mari kita nomori objek-objek tersebut dari kiri ke kanan. K = katak, B = batu, D = daun, S = seberang sungai.

0	1	2	3	4	5	6	7	8	9	10	11	12
K	B	B	D	B	B	B	B	D	B	D	B	S

Misalkan f(x) = banyaknya cara untuk mencapai seberang sungai dari petak nomor x. Mari kita hitung nilai f(x) dari x besar ke kecil. Tentu saja kita tidak perlu menghitung f() untuk petak yang berisi daun.

- f(12) = 1 (diam di tempat)
- f(11) = f(12) = 1 (lompat ke petak 12)
- f(9) = f(11) = 1 (lompat ke petak 11)
- f(7) = f(9) = 1 (lompat ke petak 9)
- f(6) = f(7) = 1 (lompat ke petak 7)
- f(5) = f(6) + f(7) = 1 + 1 = 2 (lompat ke petak 6 atau 7)
- f(4) = f(5) + f(6) = 2 + 1 = 3 (lompat ke petak 5 atau 6)
- f(2) = f(4) = 3 (lompat ke petak 4)
- f(1) = f(2) = 3 (lompat ke petak 2)
- f(0) = f(1) + f(2) = 3 + 3 = 6 (lompat ke petak 1 atau 2)

Jawaban yang diinginkan adalah f(0) = 6.
Jawaban: D

---

5. Terdapat beberapa kasus:

- Cibi mendapat 7 permen. Sisa permen = 5, cara membagikan ke Abi dan Bibi = 6.
- Cibi mendapat 8 permen. Sisa permen = 4, cara membagikan ke Abi dan Bibi = 5.
- Cibi mendapat 9 permen. Sisa permen = 3, cara membagikan ke Abi dan Bibi = 4.
- Cibi mendapat 10 permen. Sisa permen = 2, cara membagikan ke Abi dan Bibi = 3.
- Cibi mendapat 11 permen. Sisa permen = 1, cara membagikan ke Abi dan Bibi = 2.
- Cibi mendapat 12 permen. Sisa permen = 0, cara membagikan ke Abi dan Bibi = 1.

Maka, total banyaknya cara adalah 6 + 5 + 4 + 3 + 2 + 1 = 21 cara.
Jawaban: E

---

6. Soal ini adalah permasalahan kombinasi dengan perulangan. Ilustrasinya adalah sebagai berikut. Anggap Pak Dengklek memiliki 4 buah keranjang:

- Keranjang nomor 1 untuk menyimpan donat rasa stroberi
- Keranjang nomor 2 untuk menyimpan donat rasa coklat
- Keranjang nomor 3 untuk menyimpan donat rasa vanilla
- Keranjang nomor 4 untuk menyimpan donat rasa cappuccino

Anggap sekarang Pak Dengkek sedang memegang keranjang nomor 1. Setiap pembelian donat Pak Dengklek selalu dapat dinyatakan sebagai serangkaian operasi, dengan setiap operasi adalah salah satu dari:

1. Membeli donat dengan rasa yang sesuai dengan keranjang sekarang.
2. Menyudahi keranjang sekarang dan memegang keranjang nomor selanjutnya.

dan sebaliknya, setiap serangkaian operasi selalu menyatakan sebuah pembelian donat.

Karena Pak Dengklek ingin membeli 4 buah donat, banyaknya operasi (1) adalah 4. Karena Pak Dengklek memiliki 4 buah keranjang, banyaknya operasi (2) adalah 3. Maka, Pak Dengklek harus melakukan 4+3 = 7 buah operasi. Dari 7 operasi, 4 di antaranya haruslah operasi (1). Maka, banyaknya cara adalah C(7, 4) = 35.

Secara umum, jika kita memiliki n jenis barang, dan kita ingin memilih k buah barang, dengan urutan barang tidak dipentingkan dan setiap jenis boleh dipilih lebih dari sekali, maka banyaknya cara adalah C(n+k-1, n) = C(n+k-1, k-1).
Jawaban: E

---

7. Terdapat 200 orang yang saling bersalaman. Jika aturan “tidak boleh bersalaman dengan pasangan sendiri” ditiadakan, jumlah salaman yang terjadi adalah 199+198+...+1+0 salaman, yaitu 19900 salaman (hitung dengan rumus deret aritmetika). Dengan aturan “tidak boleh bersalaman dengan pasangan sendiri”, berarti ada 100 salaman yang tidak sah karena ada 100 pasangan. Maka, terdapat 19800 salaman yang terjadi.
Jawaban: C

---

8. Jika C pencuri, maka, dari pernyataan C, dapat disimpulkan bahwa A adalah prajurit. Dan karena ada setidaknya satu prajurit (A), maka pernyataan B benar, sehingga B adalah prajurit. Namun, ini menyebabkan pernyataan A salah, karena ada lebih dari satu orang prajurit. Kontradiksi. Jadi, C adalah prajurit. Karena C prajurit, maka dapat disimpulkan A pencuri dari pernyataan C. Karena ada setidaknya satu orang prajurit (C), maka B adalah prajurit. Sehingga, hanya B dan C yang merupakan prajurit.
Jawaban: D

---

9. Terdapat 6 permutasi:
- ABC : 5 + 3 = 8
- ACB : 4 + 3 = 7
- BAC : 5 + 4 = 9
- BCA : 3 + 4 = 7
- CAB : 4 + 5 = 9
- CBA : 3 + 5 = 8

Rata-rata = (8 + 7 + 9 + 7 + 9 + 8) / 6 = 8.
Jawaban: A

---

10. Mari kita tinjau bilangan-bilangan asli menurut banyaknya digit.

- Terdapat 9 bilangan satuan (1 sampai dengan 9). Masing-masing memiliki 1 digit. Maka, untuk menuliskan seluruh bilangan satuan dibutuhkan 1*9 = 9 digit.
- Terdapat 90 bilangan puluhan (10 sampai dengan 99). Masing-masing memiliki 2 digit. Maka, untuk menuliskan seluruh bilangan satuan dibutuhkan 2*90 = 180 digit.
- Terdapat 900 bilangan ratusan (100 sampai dengan 999). Masing-masing memiliki 3 digit. Maka, untuk menuliskan seluruh bilangan satuan dibutuhkan 3*900 = 2700 digit.

Karena 9 + 180 masih kurang dari 2013, dan 9 + 180 + 2700 sudah lebih dari 2013, maka kita tahu bahwa N adalah bilangan ratusan. Untuk menuliskan seluruh bilangan satuan dan puluhan, dibutuhkan 9 + 180 = 189 digit. Maka, tersisa 2013 - 189 = 1824 digit untuk menuliskan bilangan-bilangan dari 100 sampai dengan N. Karena setiap bilangan ratusan membutuhkan 3 digit, ini berarti ada 1824 / 3 = 608 bilangan yang dituliskan. Dengan demikian, N = 100 + 608 - 1 = 707.

Jawaban: B

---

11. Dalam papan catur berukuran n*n, terdapat n+1 buah garis vertikal yang membujur dari atas ke bawah, dan n+1 buah garis horizontal yang melintang dari kiri ke kanan. Untuk membentuk sebuah persegi panjang, kita harus memilih tepat 2 buah garis vertikal dan tepat 2 buah garis horizontal. Maka, banyaknya cara adalah:

C(n+1, 2) * C(n+1, 2) = (C(n+1))^2
= (n(n+1)/2)^2

Jawaban: B

---

12. Kita akan menghitung berapa banyak angka dari 1..10000 (inklusif) yang **tidak** mengandung digit 1. Karena angka 10000 jelas-jelas memiliki digit 1, kita bisa mengabaikannya. Jadi, kita akan menghitung berapa banyak angka dari 1..9999 (inklusif) yang tidak mengandung digit 1.

Perhatikan bahwa seluruh angka pada interval ini dapat dinyatakan dengan 4 digit angka. Angka 1 dinyatakan oleh 0001, angka 2 dinyatakan oleh 0002, dan seterusnya. Normalnya, tiap digit memiliki 10 kemungkinan (0..9). Karena itu ada 10*10*10*10 - 1 angka dari 1..9999 (dikurangi 1 karena ada 0000). Karena kita tidak ingin ada digit 1, maka tiap digit hanya memiliki 9 kemungkinan (0, 2..9). Ini berarti ada 9*9*9*9 - 1 angka dari 1..9999 (inklusif) yang tidak mengandung digit 1.

Banyaknya angka dari 1..10000 (inklusif) yang mengandung setidaknya 1 digit 1 adalah banyaknya angka dari 1..1000 (inklusif) dikurangi banyak angka dari 1..10000 (inklusif) yang tidak mengandung digit 1, yaitu 10000 - 6560 = 3440.

Jawaban: C

---

13. Total kebahagiaan maksimum akan diperoleh secara *greedy* apabila kita mengurutkan nilai-nilai dari Dewa Warisan dan nilai-nilai dari Dewa Sembilanbelas, lalu menikahkan anak-anak pada posisi yang sama.

Dewa Warisan:	-6, -3, 2, 5, 9
Dewa Sembilanbelas:	-4, -1, 0, 7, 8

Kebahagiaan = (-6)*(-4) + (-3)*(-1) + 2*0 + 5*7 +9*8 = 24 + 3 + 0 + 35 + 72 = 134.

Jawaban: A

---

14. Angka 0 di belakang n! dihasilkan dari faktor 10 pada n!. Karena 10 = 2 * 5, maka banyaknya angka 0 di belakang n! = min(banyaknya faktor 2 pada n!, banyaknya faktor 5 pada n!). Karena 2 < 5, maka di dalam n!, faktor 2 pasti muncul lebih banyak daripada faktor 5. Oleh karena itu, banyaknya angka 0 di belakang n! = banyaknya faktor 5 pada n!.

Banyaknya faktor 5 dari n! dapat dihitung dengan rumus berikut. Buktinya diserahkan kepada pembaca.

floor(n/5) + floor(n/(5^2)) + floor(n/(5^3)) … *(sampai suku bernilai 0)*

Mari kita hitung rumus tersebut untuk setiap pilihan jawaban:

- 8065/5 + 8065/25 + 8065/125 + 8065/625 + 8065/3125 = 2013
- 8064/5 + 8064/25 + 8064/125 + 8064/625 + 8064/3125 = 2012
- 8060/5 + 8060/25 + 8060/125 + 8060/625 + 8060/3125 = 2012
- 8051/5 + 8051/25 + 8051/125 + 8051/625 + 8051/3125 = 2010
- 8050/5 + 8050/25 + 8050/125 + 8050/625 + 8050/3125 = 2010

Maka, yang memenuhi adalah 8065.
Jawaban: A

---

15. Soal ini akan lebih mudah dikerjakan apabila kita menganggap bahwa kartu-kartu dinomori dari 0 hingga 30, dan disusun secara melingkar (bukan tumpukan), sedemikian sehingga untuk kartu bernomor X, kartu di sebelah kanannya adalah kartu bernomor (X + 1) mod 31, dan kartu di sebelah kirinya adalah kartu bernomor (X - 1) mod 31. Pertama-tama, kita “tandai” bahwa kartu teratas adalah kartu bernomor 0.

Perhatikan bahwa melakukan 1 langkah sama saja dengan memindahkan tanda tersebut sebanyak 5 kartu ke sebelah kiri. Oleh karena itu, dalam 45 langkah, tanda akan berpindah sebanyak 5*45 = 225 langkah ke sebelah kiri. Maka, tanda tersebut akan berada pada kartu nomor (0 - 225) mod 31 = -225 mod 31 = -8 mod 31 = 23. Maka, kartu teratas adalah kartu nomor 24.
Jawaban: A

---

16. Dengan cara yang sama seperti nomor sebelumya, tanda akan berada pada kartu nomor (0 - 7*50) mod 31 = -350 mod 31 = -9 mod 31 = 22. Maka, kartu teratas adalah kartu nomor 23.
Jawaban: B

---

17. Selesaikan persamaan berikut:

$$1 \equiv 0 - 3x \pmod{31}$$
$$1 \equiv -3x \pmod{31}$$
$$3x \equiv -1 \pmod{31}$$
$$3x \equiv 30 \pmod{31}$$

Maka, x = 10.
Jawaban: D

---

18. Banyaknya langkah minimum akan tercapai apabila kita melakukan *selection sort*.
Pada mulanya: 3, 2, 1, 5, 7, 4, 8, 6, 10, 9

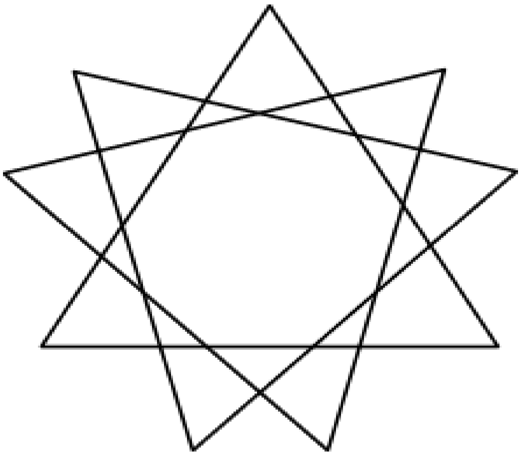
- Langkah 1: 1, 2, 3, 5, 7, 4, 8, 6, 10, 9
- Langkah 2: 1, 2, 3, 4, 7, 5, 8, 6, 10, 9
- Langkah 3: 1, 2, 3, 4, 5, 7, 8, 6, 10, 9
- Langkah 4: 1, 2, 3, 4, 5, 6, 8, 7, 10, 9
- Langkah 5: 1, 2, 3, 4, 5, 6, 7, 8, 10, 9
- Langkah 6: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Maka, dibutuhkan minimum 6 langkah.

Jawaban: C

---

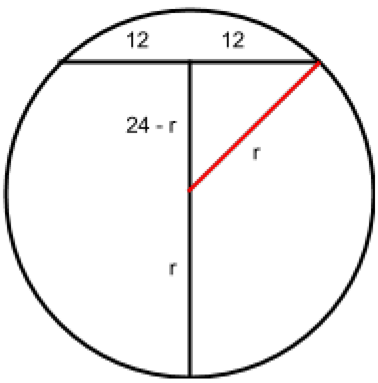
19. Jika kita menindihnya seperti ini, akan terdapat 30 segitiga (termasuk 3 segitiga awalnya).



Jawaban: D

---

20. Misalkan jari-jari pipa tersebut adalah r, maka:



Dengan rumus Pythagoras,

$r^2 = (24 - r)^2 + 12^2$

$r^2 = 576 - 48r + r^2 + 144$

$48r = 720$

$r = 15$

Maka, diameternya adalah 30 cm.

Jawaban: B

---

21. Misalkan f(X) = biaya minimum untuk mengalirkan air dari keran X menuju ke kanan. Kita hitung nilai-nilai dari f(X) dari kanan ke kiri sebagai berikut.

- f(A3) = 16 + 21 = 37
- f(B3) = 11 + 14 = 25
- f(A2) = 23 + min(f(A3), 4 + f(B3)) = 23 + min(37, 29) = 23 + 29 = 52
- f(B2) = 25 + min(f(B3), 7 + f(A3)) = 26 + min(25, 44) = 26 + 25 = 51
- f(A1) = 6 + min(f(A2), 5 + f(B2)) = 6 + min(52, 56) = 6 + 52 = 58
- f(B1) = 17 + min(f(B2), 3 + f(A2)) = 17 + min(51, 65) = 17 + 51 = 68

Maka, biaya minimum keseluruhan adalah min(13 + f(A1), 9 + f(B1)) = min(71, 77) = 71.
Jawaban: B

---

22. Banyaknya cara mendapatkan 4 buah kartu dengan nilai yang sama adalah 13 cara, karena terdapat 13 macam nilai dan masing-masing nilai memiliki tepat 4 buah kartu. Kartu kelima dapat dipilih dalam 52 - 4 = 48 cara. Maka, banyaknya cara terjadinya *Four of a Kind* adalah 13 * 48 = 624.
Jawaban: D

---

23. Banyaknya cara mendapatkan nilai X adalah 13, dan banyaknya cara mendapatkan nilai Y adalah 12. Banyaknya cara mendapatkan 3 dari 4 kartu bernilai X adalah C(4, 3) = 4. Banyaknya cara mendapatkan 2 dari 4 kartu bernilai Y adalah C(4, 2) = 6. Maka, banyaknya cara terjadi *Full House* adalah 13 * 12 * 4 * 6 = 3744.
Jawaban: E

---

24. Mari kita hitung total waktu yang diperlukan setiap peserta.
- Andy: 1000 / 100 + floor(1000 / 80) * 45/60 = 19 jam
- Budi: 1000 / 80 + floor(1000 / 75) * 30/60 = 19 jam
- Cory: 1000 / 100 + floor(1000 / 70) * 15/60 = 13,5 jam
- Didy: 1000 / 80 + floor(1000 / 65) * 12/60 = 15,5 jam
- Erna: 1000 / 100 + floor(1000 / 60) * 6/60 = 11,6 jam

Maka, pemenangnya adalah Erna.
Jawaban: E

---

25. Terdapat 5 buah kemungkinan, yakni:

D		Supir
G	F	C
E	B	A

D		Supir
G	F	C
E	A	B

D		Supir
G	F	C
A	B	E

D		Supir
G	F	C
B	A	E

G		Supir
D	F	C
E	B	A

Jawaban: B

---

26. Dari 5 buah kemungkinan pada nomor sebelumnya, hanya terdapat 2 buah kemungkinan orang pada posisi 1: G atau D. Jika G bukan yang paling depan, maka pastilah D.
Jawaban: D

---

27. Berdasarkan prinsip inklusi-eksklusi, banyaknya untaian bit dengan panjang 10 yang diawali 000 atau diakhiri 1111 sama dengan banyak untaian bit dengan panjang 10 yang diawali 000 + banyak untaian bit dengan panjang 10 yang diakhiri 1111 - banyak untaian bit dengan panjang 10 yang diawali 000 dan diakhiri 1111.
Banyak untaian bit dengan panjang 10 yang diawali 000 ada 2^7.

Banyak untaian bit dengan panjang 10 yang diakhiri 1111 ada 2^6.
Banyak untaian bit dengan panjang 10 yang diawali 000 dan diakhiri 1111 ada 2^3.
Maka, banyak untaian bit dengan panjang 10 yang diawali 000 atau diakhiri 1111 ada 128+64-8 = 184.
Jawaban: B

---

28. Tinjau pernyataan-pernyataan tersebut.
- Basith adalah anggota asosiasi alumni TOKI. (benar)
- Cakra adalah alumni TOKI (belum tentu karena Cakra belum tentu ikut pelatnas)
- Nathan pernah mengikuti pelatnas (benar)
- Cakra pernah mengikuti IOI (belum tentu karena Cakra belum tentu ikut pelatnas)

Jawaban: D

---

29. Berdasarkan prinsip inklusi-eksklusi, banyaknya nomor telepon yang mudah diingat adalah (banyaknya nomor dengan abc = def) + (banyaknya nomor dengan abc = efg) - (banyaknya nomor dengan abc = def dan abc = efg).

Banyaknya nomor dengan abc = def ada 10*10*10*10 kemungkinan. (a, b, c, g masing-masing memiliki 10 kemungkinan).

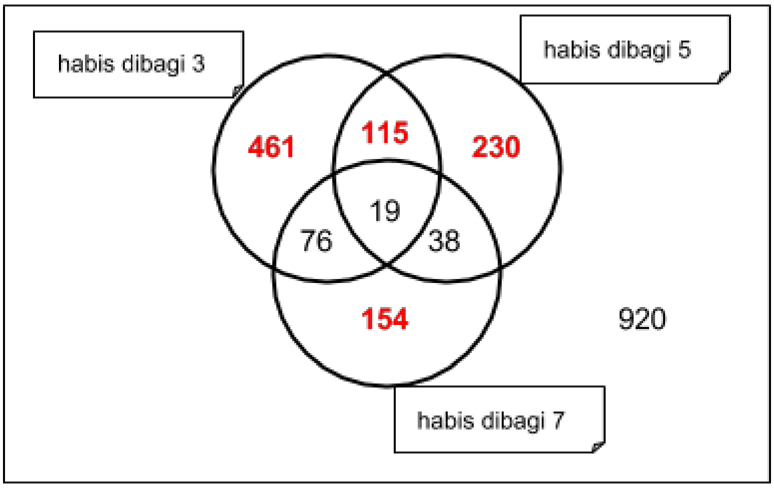
Banyaknya nomor dengan abc = efg ada 10*10*10*10 kemungkinan. (a, b, c, d masing-masing memiliki 10 kemungkinan).

Ketika abc = def dan abc = efg, ini berarti a = d = e, b = e = f,c = f = g. Jika kita telusuri lebih lanjut, ini berarti a = b = c = d = e = f = g, yang berarti semua digit sama. Ada 10 kemungkinan.

Maka, banyaknya nomor telepon yang mudah diingat ada sebanyak 19990.
Jawaban: E

---

30. Kita gambarkan Diagram Venn untuk masalah ini:



Untuk membuatnya, pertama hitung banyaknya bilangan yang habis dibagi 3, 5, dan 7, yaitu floor(2013 / (3*5*7)) = 19. Lalu, hitung banyaknya bilangan yang habis dibagi 3 dan 5, yaitu floor(2013 / (3*5)) = 134. Karena sudah terisi 19, maka sisanya 115. Dan seterusnya.

Total bilangan yang dicetak merah merupakan jawaban yang diinginkan, yakni 461 + 115 + 230 + 154 = 960.
Jawaban: B

Bagian B: Algoritmika (20 Soal)

31. Fungsi panggil1() bersifat rekursif. Mari kita selesaikan dengan cara *bottom-up*. Perhatikan bahwa nilai dari panggil1(x) bergantung pada nilai panggil1(x-1) dan panggil1(x-2). Oleh karena itu, kita hitung nilai fungsi ini dari nilai x yang kecil ke besar.
- panggil1(1) = panggil1(2) = 0
- panggil1(3) = panggil1(2) + 2*panggil1(1) + 3 = 0 + 2*0 + 3 = 3
- panggil1(4) = panggil1(3) + 2*panggil1(2) + 3 = 3 + 2*0 + 3 = 6
- panggil1(5) = panggil1(4) + 2*panggil1(3) + 3 = 6 + 2*3 + 3 = 15
- panggil1(6) = panggil1(5) + 2*panggil1(4) + 3 = 15 + 2*6 + 3 = 30
- panggil1(7) = panggil1(6) + 2*panggil1(5) + 3 = 30 + 2*15 + 3 = 63

Jawaban: D

---

32. Fungsi sikat() bersifat rekursif. Mari kita selesaikan dengan cara *bottom-up*. Perhatikan bahwa nilai dari sikat(x, y) bergantung pada nilai sikat(x+1, y) dan sikat(x, y-1). Oleh karena itu, kita hitung nilai fungsi ini dari nilai x yang besar ke kecil, dan dari nilai y yang kecil ke besar.
- sikat(1, 1) = 1
- sikat(2, 2) = 2
- sikat(1, 2) = 3*sikat(2, 2) + 2*sikat(1, 1) = 3*2 + 2*1 = 8
- sikat(3, 3) = 3
- sikat(2, 3) = 3*sikat(3, 3) + 2*sikat(2, 2) = 3*3 + 2*2 = 13
- sikat(1, 3) = 3*sikat(2, 3) + 2*sikat(1, 2) = 3*13 + 2*8 = 55

Jawaban: B

---

33. Fungsi hitung() bersifat rekursif. Cara sebelumnya tidak praktis digunakan karena terlalu banyak jika kita harus menghitung nilai hitung(0), hitung(1), …, hitung(2903). Perhatikan bahwa nilai hitung(n) bergantung pada nilai hitung(n div 5). Dengan kata lain, pada setiap pemanggilan fungsi hitung, nilai n selalu berkurang menjadi seperlimanya. Oleh karena itu, lebih cepat jika kita menghitung dengan cara *top-down*.

```
hitung(2903)  = hitung(580) + 3
              = hitung(116) + 0 + 3
              = hitung(23) + 1 + 0 + 3
              = hitung(4) + 3 + 1 + 0 + 3
              = hitung(0) + 4 + 3 + 1 + 0 + 3
              = 0 + 4 + 3 + 1 + 0 + 3
              = 11
```

Jawaban: C

34. Pertama-tama, kita analisis terlebih dahulu, apa yang sebenarnya fungsi wow(n) hitung? Jika kita baca dan pelajari kodenya baik-baik, rupanya jika faktorisasi prima dari n adalah p_1^a_1 * p_2^a_2 * … * p_k^a_k, maka wow(n) bernilai (a_1+1) * (a_2+1) * … * (a_k+1).

Oleh karena itu, kita dapat saja mencoba memfaktorisasikan setiap pilihan jawaban:

- 32 = 2^5 -> wow(32) = 6
- 512 = 2^9 -> wow(512) = 10
- 1000 = 2^3 * 5^3 -> wow(1000) = 4 * 4 = 16
- 48 = 2^4 * 3 -> wow(48) = 5 * 2 = 10
- 38 = 2 * 19 -> wow(38) = 2 * 2 = 4

Dari semua pilihan di atas, nilai n terkecil yang menghasilkan wow(n) = 10 adalah 48.
Jawaban: D

---

35. Kita memiliki sebuah struktur data *stack*. Pada awalnya, *stack* tersebut berisi nilai-nilai 1, 3, 5, 7, 9 dari atas (*head*) ke bawah (*tail*). Fungsi pop() akan membuang nilai teratas dari *stack* dan mengembalikannya. Jika *stack* kosong, maka pop() akan mengembalikan nilai -1. Nilai tersebut tidak benar-benar dibuang, namun *head*-nya saja yang bergerak 1 elemen mendekati *tail*.

Kondisi awal *stack* adalah sebagai berikut:

head					**tail**
1	3	5	7	9	

Setelah dilakukan pop() satu kali, kondisinya berubah menjadi:

head				**tail**
3	5	7	9	

Kemudian, 3 nilai teratas akan di-pop() sekaligus dicetak. Maka, yang tercetak adalah:
3
5
7
Jawaban: C

---

36. Kita coba simulasikan kode tersebut.
x = 1, pop() = 1, kondisi berubah menjadi:

head				**tail**
3	5	7	9	

Karena 1 <> -1, maka pop() dipanggil lagi (menghasilkan dan membuang 3), lalu dicetak.
Kondisi berubah menjadi:

head			**tail**
5	7	9	

x = 2, pop() = 5, kondisi berubah menjadi:

head		tail
7	9	

Karena 5 <> -1, maka pop() dipanggil lagi (menghasilkan dan membuang 7), lalu dicetak. Kondisi berubah menjadi:

head	tail
9	

x = 3, pop() = 9, kondisi berubah menjadi:

head, tail

Karena 9 <> -1, maka pop() dipanggil lagi (menghasilkan -1), lalu dicetak.

Maka, yang tercetak adalah:
3
7
-1
Jawaban: D

---

37. Fungsi Bunga() bersifat rekursif. Mari kita selesaikan dengan cara *bottom-up*. Perhatikan bahwa nilai dari Bunga(x) bergantung pada nilai dari Bunga(y), untuk 0 <= y <= x. Oleh karena itu, kita hitung nilai fungsi ini dari nilai x yang kecil ke besar.

- Bunga(0) = 1
- Bunga(1) = Bunga(0) = 1
- Bunga(2) = Bunga(0) + Bunga(1) = 1 + 1 = 2
- Bunga(3) = Bunga(0) + Bunga(1) + Bunga(2) = 1 + 1 + 2 = 4

Jawaban: D

---

38. Jika kita teruskan menghitung nilai-nilai Bunga pada nomor sebelumnya, maka akan terlihat pola bahwa Bunga(x) = 2^(x-1) untuk x > 0. Oleh karena itu, Bunga(21) = 2^20 = 1048576. Hal ini tentu saja dapat dibuktikan dengan metode induksi, yang buktinya diserahkan kepada para pembaca.
Jawaban: B

---

39. Mari kita simulasikan *loop while* pada kode tersebut. Perhatikan bahwa potongan kode:

```
if(num1 mod 2 + num2 mod 2 = 1) then
        dare := dare + temp
else
        dare := dare + ( (num1 mod 2 + num2 mod 2) * temp );
```

sebenarnya sama saja dengan

```
dare := dare + ( (num1 mod 2 + num2 mod 2) * temp );
```

sehingga dapat mempermudah penghitungan.

Hasil simulasi:

Iterasi	num1	num2	num1 mod 2 + num2 mod 2	dare	temp
0	16	8	0	0	1
1	8	4	0	0	2
2	4	2	0	0	4
3	2	1	1	0	8
4	1	0	1	8	16
5	0	0		**24**	32

Jawaban: B

---

40. Hasil simulasi:

Iterasi	num1	num2	num1 mod 2 + num2 mod 2	dare	temp
0	120	30	0	0	1
1	60	15	1	0	2
2	30	7	1	2	4
3	15	3	2	6	8
4	7	1	2	22	16
5	3	0	1	54	32
6	1	0	1	86	64
7	0	0		**150**	128

Jawaban: B

---

41. Terdapat setidaknya dua cara untuk menjawab soal ini.
Cara 1: Pilihan jawaban yang mana yang cocok dengan kedua contoh berikut?

- Masukan: 16 8 Keluaran: 24
- Masukan: 120 30 Keluaran: 150

Tentu saja, yang cocok adalah A. Menjumlahkan 2 buah bilangan bulat positif.

Cara 2: Menganalisis program tersebut lebih mendalam. Perhatikan bahwa yang dilakukan oleh program tersebut adalah memproses kedua bilangan masukan bit-per-bit, dari nilai satuan, duaan, empatan, delapanan, enam belasan, dan seterusnya (yakni, dalam basis dua). Nilai temp adalah 2^i, dengan i adalah posisi bit saat ini. Nilai num1 mod 2 menunjukkan apakah bit ke-i dari num1 menyala atau tidak. Nilai num2 mod 2 menunjukkan apakah bit ke-i dari num2 menyala atau tidak. Apabila dijumlahkan, ini menunjukkan banyaknya bit yang menyala pada posisi ke-i. Kemudian, nilai dare ditambahkan dengan (temp * banyaknya bit yang menyala). Pada akhirnya, semua bit akan diproses dan ini menunjukkan bahwa sebenarnya yang dilakukan adalah menjumlahkan kedua bilangan tersebut.
Jawaban: A

42. Karena M = 100, tidak praktis kalau kita melakukan simulasi program. Kita harus melakukan analisis mengenai apa yang sebenarnya dilakukan program. Ternyata, program ini melakukan algoritma Sieve of Eratosthenes. Setelah dijalankan, A[x] akan bernilai true jika x prima (atau 1) dan false jika selainnya. B[x] akan bernilai bilangan prima terkecil ke-x. Sehingga, nilai B[10] adalah bilangan prima ke-10 yakni 29.
Jawaban: D

---

43. Cukup lakukan simulasi terhadap program tersebut. Potongan kode ini:

```
t := A[i] xor A[j];
A[i] := t xor A[i];
A[j] := t xor A[i]
```

berarti menukarkan nilai A[i] dan A[j].

i	A[1]	A[2]	A[3]	A[4]	A[5]
	1	2	3	4	5
1	1	2	3	4	5
2	2	1	3	4	5
3	3	2	1	4	5
4	4	3	2	1	5
5	5	4	3	2	1

Jawaban: B

---

44. Lakukan simulasinya secara *top-down* (pemanggilan fungsi kambing(x) untuk x > 10 tidak ditulis untuk menghemat tempat):

kambing(8) **data[8]**
kambing(4) **data[4]**
kambing(9) **data[9]**
kambing(2) **data[2]**
kambing(10) **data[10]**
kambing(5) **data[5]**

kambing(1) **data[1]**

kambing(6) **data[6]**

kambing(3) **data[3]**

kambing(7) **data[7]**

Dengan demikian, urutan indeks pencetakan adalah 8, 4, 9, 2, 10, 5, 1, 6, 3, 7. Sehingga, yang tercetak adalah 8 6 5 9 10 1 3 4 2 7.
Jawaban: D

45. Agar hasil yang tercetak terurut menaik, menurut hasil pada soal sebelumnya, maka hal ini harus berlaku: data[8] = 1, data[4] = 2, data[9] = 3, …, data[7] = 10. Maka, *array* yang memenuhi adalah (7, 4, 9, 2, 6, 8, 10, 1, 3, 5).
Jawaban: A

---

46. Cukup lakukan simulasi terhadap program tersebut.

x	13	40	20	10	5	16	8	4	2	1
i	0	1	2	3	4	5	6	7	8	**9**

Jawaban: E

---

47. Perhatikan fungsi rekursif ghi(). Untuk memudahkan simulasi, kita harus mencari bentuk tertutup (non-rekursif) dari fungsi ini. Jika dilakukan analisis mendalam, maka ghi(x, b) bernilai:

- x, jika b = 1
- x^b mod 100, jika b > 1.

Setelah mengetahui fakta ini, kita lakukan simulasi terhadap program tersebut:

```
def(7, 100)   = def(7, abc(100))
              = def(7, 298)
              = def(7, abc(298))
              = def(7, 890)
              = def(7, abc(890))
              = def(7, 2669)
              = ghi(7, 2669) mod 100
              = (7^2669 mod 100) mod 100
              = 7^2669 mod 100
              = 7
```

Hasil penghitungan terakhir di atas dapat dilakukan menggunakan teknik *exponentiation by squaring* atau menggunakan teorema Euler.
Jawaban: B

---

48. Kembali lakukan simulasi terhadap program yang telah dimodifikasi tersebut:

```
def(7, 151)   = def(7, abc(151))
              = def(7, 301)
              = def(7, abc(301))
              = def(7, 601)
              = ghi(7, 601) mod 100
              = (7^601 mod 100) mod 100
              = 7^601 mod 100
              = 7
```

Hasil penghitungan terakhir di atas dapat dilakukan menggunakan teknik *exponentiation by squaring* atau menggunakan teorema Euler.
Jawaban: B

---

49. Kembali lakukan simulasi terhadap program yang telah dimodifikasi tersebut:

```
def(3, 30)  = def(3, abc(30))
            = def(3, 88)
            = def(3, abc(88))
            = def(3, 260)
            = def(3, abc(260))
            = def(3, 779)
            = ghi(3, 779) mod 10
            = (3^779 mod 100) mod 10
            = 67 mod 10
            = 7
```

Hasil penghitungan kedua dari terakhir di atas dapat dilakukan menggunakan teknik *exponentiation by squaring* atau menggunakan Teorema Euler.

Jawaban: C

---

50. Cukup lakukan simulasi terhadap program tersebut.

```
tebak_angka(18, 30) = tebak_angka(18, 26)
                    = tebak_angka(18, 22)
                    = tebak_angka(18, 18)
                    = tebak_angka(18, 14)
                    = tebak_angka(14, 18)
                    = tebak_angka(14, 13)
                    = tebak_angka(13, 14)
                    = tebak_angka(13, 10)
                    = tebak_angka(10, 13)
                    = tebak_angka(10, 12)
                    = tebak_angka(10, 11)
                    = tebak_angka(10, 10)
                    = tebak_angka(10, 9)
                    = tebak_angka(9, 10)
                    = tebak_angka(9, 5)
                    = tebak_angka(5, 9)
                    = tebak_angka(5, 8)
                    = tebak_angka(5, 7)
                    = tebak_angka(5, 6)
                    = tebak_angka(5, 5)
                    = tebak_angka(5, 4)
                    = tebak_angka(4, 5)
                    = 5
```

Jawaban: E