

Pembahasan Olimpiade Sains 2016 Tingkat Provinsi Bidang Informatika/Komputer



Dibahas oleh:

- Chalvin Polpan
- Degoldie Sonny
- Ensof Barhami
- Fata Nugraha
- Firman Hadi Prayoga
- Hansvin Tandi Sugata
- Kevin Luvian
- Muhammad Ashlah Shinfain
- Norman Bintang
- Ricky Jeremiah Djajadi
- Turfa Auliarachman
- Yonas Adiel Wiguna

Diedit oleh:

- Dr. Inggriani Liem
- Muhammad Ayaz Dzulfikar
- William Gozali

Bagian A: Pertanyaan Analitika/Logika (30 soal)

- Perhatikan bahwa digit terkanan $(4 + x)^{20}$ hanya dipengaruhi oleh digit terkanan $4 + x$. Kita bisa memeriksa untuk setiap kemungkinan digit terkanan:

Digit terkanan $(4 + x)$	Digit terkanan $(4 + x)^{20}$
0	0
1	1
2	6
3	1
4	6
5	5
6	6
7	1
8	6
9	1

Terlihat bahwa digit terkanan $(4 + x)^{20}$ bernilai 1 apabila $(4 + x)$ memiliki digit terkanan 1, 3, 7, dan

9. Nilai $(4 + x)$ yang minimum dan memenuhi x positif adalah 7, sehingga x bernilai 3.

Jawaban: 3

-
- Diperlukan pemahaman soal konsep *graph* untuk menyelesaikan soal ini.

Permasalahan ini dapat dimodelkan sebagai berikut:

“Ada berapa *edge* yang perlu ditambah agar terdapat *eulerian path* dari *graph* tersebut?”

Suatu *graph* dapat memiliki suatu *eulerian path* jika dan hanya jika banyak *node* yang memiliki *degree* ganjil tidak lebih dari 2 (perhatikan bahwa tidak mungkin terdapat *graph* yang memiliki banyaknya *node* dengan *degree* ganjil sebanyak 1). Perlu ditambahkan

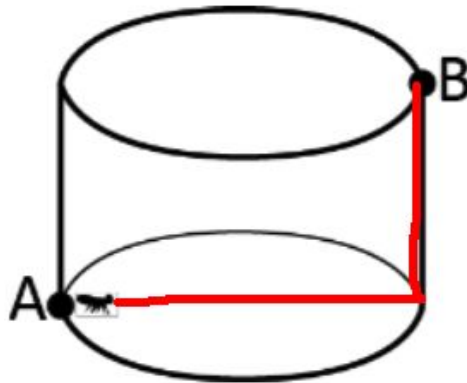
sedikit mungkin *edge* agar syarat tersebut terpenuhi.

Dari gambar didapatkan ada 6 *node* yang memiliki *degree* ganjil. Hubungkan 4 *node* diantaranya dengan 2 *edge* agar jumlah *node* yang memiliki *degree* ganjil tinggal 2. Dengan demikian jawaban dari soal ini adalah 2.

Materi tentang *eulerian path* dapat diakses di <http://bit.ly/toki-eulerian-path>.

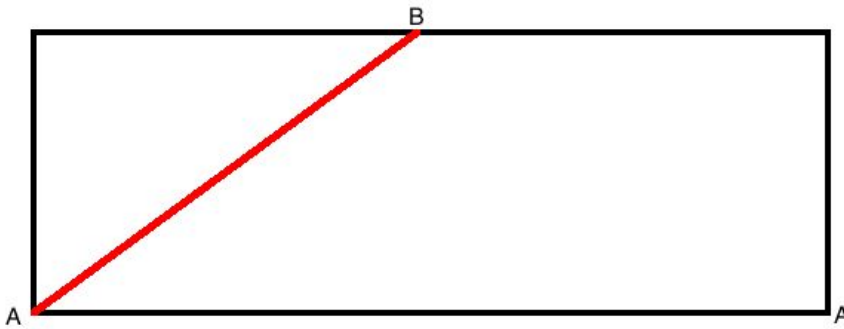
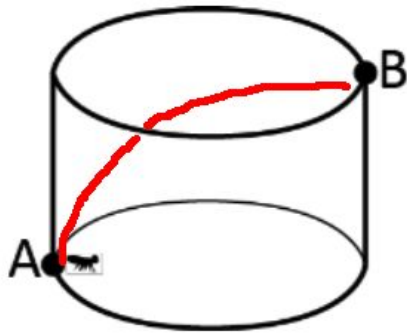
Jawaban: 2

-
3. Soal ini menjebak, kebanyakan orang akan berpikir sebaiknya semut berjalan seperti ini



Jarak semut dari A ke B dengan rute tersebut adalah $14 + 15 = 29$.

Ternyata, ada rute lain yang lebih pendek, yaitu menelusuri selimut tabung. Solusi ini memberikan jarak yang lebih pendek daripada rute sebelumnya.



Jarak semut dari A ke B dengan rute tersebut: $\sqrt{(\pi * 7)^2 + 15^2} = 26.62$

Jawaban: 26.62

4. Banyak kelompok arisan adalah banyaknya bilangan yang bukan merupakan bilangan kuadrat atau kubik sempurna. Sebab, orang yang beridentitas berupa bilangan kuadrat atau kubik sempurna pasti dalam kelompoknya memiliki **seorang** teman yang identitasnya bukan bilangan kuadrat maupun kubik sempurna.

Banyaknya bilangan bukan kuadrat atau kubik yang tidak lebih dari 2016:
 $2016 - \lfloor \sqrt[3]{2016} \rfloor - \lfloor \sqrt{2016} \rfloor + \lfloor \sqrt[6]{2016} \rfloor = 1963$.

Perhatikan bahwa banyaknya kelompok harus ditambah 1, karena orang dengan identitas 1 tidak memiliki teman kelompok yang identitasnya bukan bilangan kuadrat atau kubik sempurna. Jadi, terdapat 1964 kelompok.

Jawaban: 1964

5. Definisikan $f(n, xy)$ adalah banyak cara mengisi n pot, dengan dua pot terakhir diisi tanaman dengan karakter x dan y .

Untuk kasus paling sederhana, diketahui:

$$f(2, aa) = f(2, ab) = f(2, ac) = f(2, ba) = f(2, bb) = f(2, bc) = f(2, ca) = f(2, cb) = f(2, cc) = 1$$

2 pot terakhir bisa berisi aa bila pot terakhir a, dan kita isi pot yang baru dengan a.

Sehingga dapat dirumuskan:

$$f(n, aa) = f(n-1, aa) + f(n-1, ba) + f(n-1, ca)$$

Kita dapat mendefinisikan rumus yang mirip untuk akhiran bb dan cc. Dapat dipastikan nilai $f(n, aa)$

bernilai sama dengan $f(n, bb)$ dan $f(n, cc)$.

Kemudian, apabila 2 pot terakhir berisi ab, dapat dirumuskan:

$$f(n, ab) = f(n-1, aa) + f(n-1, ba)$$

Perhatikan bahwa apabila berakhiran ab, tidak boleh didapat dari yang 2 sebelumnya ca, karena akan menghasilkan 3 pot bersebelahan yang semuanya berbeda. Nilai $f(n, ab)$ akan sama dengan $f(n, ac)$, $f(n, ba)$, $f(n, bc)$, $f(n, ca)$, $f(n, cb)$.

Selanjutnya, kita hitung masing-masing nilai $f(n, aa)$ dan $f(n, ab)$:

$$f(2, aa) = 1$$

$$f(2, ab) = 1$$

$$f(3, aa) = 3$$

$$f(3, ab) = 2$$

$$f(4, aa) = 7$$

$$f(4, ab) = 5$$

$$f(5, aa) = 17$$

$$f(5, ab) = 12$$

$$f(6, aa) = 41$$

$$f(6, ab) = 29$$

Maka, banyak cara mengisi 6 pot adalah $3 * f(6, aa) + 6 * f(6, ab) = 3 * 41 + 6 * 29 = 297$.

Jawaban: 297

6. Alternatif 1:

Gunakan teknik *dynamic programming* (DP). Definisikan $f(x, y)$ sebagai fungsi untuk mencari kekuatan awal minimum di titik (x, y) agar dapat sampai ke titik akhir, yaitu $(6, 6)$.

Selain itu, definisikan petak $[x][y]$ sebagai perubahan kekuatan ketika ada di posisi (x, y) . Oleh karena itu, basis dari DP kita adalah:

$$f(6, 6) = \max(1, 1 - \text{petak}[x][y])$$

Dari suatu petak, gerakan yang mungkin adalah ke bawah atau ke kanan, yaitu ke petak $(x+1, y)$ dan $(x, y+1)$. Kekuatan di titik (x, y) , setelah mendapat vitamin atau racun, tentu harus dapat mencapai kekuatan minimum di kedua petak tersebut. Jadi, rekurens dari DP kita adalah:

$$f(x, y) = \max(1, \min(f(x + 1, y), f(x, y + 1)) - \text{petak}[x][y])$$

Hasil dari DP kita adalah:

83	60	71	54	62	127
64	86	61	63	52	117
81	61	3	31	7	52
70	79	32	1	11	31
60	49	1	18	1	56
57	71	1	1	1	42

Jawaban yang kita inginkan terdapat di $f(1, 1)$. Nilai tersebut adalah 83.

Materi tentang *dynamic programming* dapat diakses di <http://bit.ly/toki-dp>

Alternatif 2:

Soal ini dapat diselesaikan dengan mencoba semua kemungkinan jawaban, lalu mencari kekuatan awal terkecil yang mungkin mencapai ujung kanan bawah. Namun terlalu banyak kemungkinan jawaban. Oleh karena itu, dapat digunakan teknik *Binary Search the Answer*. Anggap batas bawah adalah 0 dan batas atas adalah 400. Kita ambil nilai tengah 200, lalu kita telusuri dengan prioritas langkah kanan lalu bawah:

177	188	171	179	169	159
					94
					73
					98
					84
					73

Rupanya ksatria mampu mencapai titik akhir bila kekuatan awalnya 200. Sekarang ubah batas atas menjadi 200, sehingga nilai tengah menjadi $\lfloor \frac{200+0}{2} \rfloor = 100$

72	88	71	79	69	59
				24	

				28	
				18	
				20	
				61	20

Rupanya ksatria masih mampu mencapai titik akhir. Sekarang ubah batas atas menjadi 100, sehingga nilai tengah menjadi $\lfloor \frac{100+0}{2} \rfloor = 50$.

27	38	21	29	19	9
44	19				
24					
14					
3					
17					

Ternyata ksatria tak mampu mencapai titik akhir, sehingga batas bawah dinaikkan menjadi 51. Nilai tengah akan menjadi $\lfloor \frac{100+51}{2} \rfloor = 75$

52	63	46	54	44	34
69	44		22		
49					
39					
28					
42					

Lakukan terus hingga menemukan hasil 83, yang mana dengan 83 mampu mencapai titik akhir namun 82 tidak cukup. Kemungkinan terburuk adalah $(\log_2 400)$ kali mencoba, yakni 9 kali.

Contoh lain tentang persoalan *Binary Search the Answer* dapat diakses di <http://bit.ly/toki-binser>

Jawaban: 83

7. Faktorisasi prima dari 337500 adalah: $2^2 * 3^3 * 5^5$

Untuk menghitung banyaknya angka yang membagi 337500, dapat dipilih sebuah angka

dengan faktorisasi $2^a * 3^b * 5^c$, yang mana nilai a, b, dan c masing-masing berada dalam rentang [0, 2], [0, 3], dan [0, 5].

Oleh karena itu banyaknya bilangan bulat positif yang habis membagi 337500 adalah $(2 + 1) * (3 + 1) * (5 + 1) = 72$

Jawaban: 72

8. Banyak caranya adalah banyak permutasi dari AABBCCDDEE. Karena makanan harus didahulukan sebelum minuman, maka suatu makanan dan minuman "X" harus dilambangkan dengan huruf yang sama. Banyak permutasi adalah $\frac{10!}{2!*2!*2!*2!*2!}$

Jawaban: $\frac{10!}{2!*2!*2!*2!*2!}$

9. Pak Dengklek tidak perlu menambahkan apapun karena sudah terdapat kumpulan bilangan yang menghasilkan nilai 0, yaitu {0}. Maka, jawabannya adalah 0.

Jawaban: 0

10. Pertama, enumerasikan teman-teman Pak Dengklek = {Pak Ganesh, Bu Ganesh, Bu Blangkon, Bu Sarah, Bu Sura, Bu Dengklek, Pak James}

Kemudian enumerasikan teman-teman Bu Dengklek = {Pak Wawan, Pak Ronald, Pak James, Bu Blangkon, Pak Dengklek, Bu Ganesh, Bu Sarah}

Agar Bu Dengklek tidak melihat foto Pak Dengklek, Pak Dengklek tidak boleh mengirim fotonya ke teman Bu Dengklek. Tentu saja, Pak Dengklek juga tidak boleh mengirim foto ke Bu Dengklek. Sehingga, Pak Dengklek dapat mengirim fotonya ke Pak Ganesh dan Bu Sura.

Jawaban: Pak Ganesh, Bu Sura

11. Banyaknya pecahan mata uang yang diperlukan Pak Dengklek adalah sebanyak angka 1 pada representasi biner 714, yaitu $(10\ 1100\ 1010)_2$. Terdapat 5 buah angka 1, sehingga minimum Pak Dengklek membutuhkan 5 koin.

Jawaban: 5

12. Untuk menghitung minimal bola yang menjamin Pak Dengklek mengambil 2 bola untuk setiap warna, kita harus mencari kasus terburuk dari kejadian ini. Asumsikan Pak Dengklek sedang sangat tidak beruntung saat itu. Berapa bola terbanyak yang dibutuhkan Pak Dengklek untuk mendapatkan 2 bola untuk tiap warna?

Untuk kasus tersebut, bola yang diambil adalah: PPPMMMMHH
Sehingga, jawabannya adalah 8 bola.

Jawaban: 8

-
13. Cara menghitung Pak Dengklek berulang lagi ketika mencapai jari manis untuk yang kedua kalinya dalam satu putaran (yang mana hitungannya mencapai hitungan ke-8). Hitungan terakhir pada putaran selanjutnya adalah 16, kemudian 24, 32, dan seterusnya. Sehingga, dapat dicari tahu jari mana yang merepresentasikan hitungan kesekian dengan menggunakan metode modulo (modulo 8).

Apabila hasil modulo adalah 1, maka jari yang merepresentasikan adalah jari kelingking. Apabila hasil modulo 2, maka jari yang merepresentasikan adalah jari manis, dan seterusnya. Contohnya dapat dilihat pada tabel berikut:

Putaran ke	Hitungan ke	Representasi jari
1	1	Kelingking
	2	Manis
	3	Tengah
	4	Telunjuk
	5	Jempol
	6	Telunjuk
	7	Tengah
	8	Manis
2	9	Kelingking

	10	Manis
	11	Tengah

Untuk hitungan ke-2016, hasil dari $2016 \bmod 8$ adalah 0, sehingga jari yang mempresentasikan adalah jari manis.

Jawaban: jari manis

14. Berikut ini adalah masing-masing kemungkinan untuk tiap banyak batang di suatu ruas:

Banyak Batang	Kemungkinan	Nilai
2	1	1
3	7	7
4	4	4
5	2, 3, 5	2, 3, 5
6	0, 1+1, 6, 9	0, 2, 6, 9
7	1+7, 7+1, 8	8, 8, 8
8	1+4, 4+1, 7+7	5, 5, 14

9	1+2, 2+1, 1+3, 3+1, 1+5, 5+1, 4+7, 7+4	3, 3, 4, 4, 6, 6, 11, 11
10	0+1, 1+0, 1+1+1, 1+6, 6+1, 4+4, 2+7, 7+2, 1+9, 3+7, 7+3, 9+1, 5+7, 7+5	1, 1, 3, 7, 7, 8, 9, 9, 10, 10, 10, 10, 12, 12
11	2+4, 4+2, 0+7, 3+4, 4+3, 7+0, 1+1+7, 1+7+1, 1+8, 4+5, 5+4, 7+1+1, 8+1, 6+7, 7+6, 7+9, 9+7	6, 6, 7, 7, 7, 7, 9, 9, 9, 9, 9, 9, 9, 13, 13, 16, 16

Terdapat 2 observasi yang dapat kita buat:

- Kita harus meletakkan sebuah tanda sama dengan, sehingga bila kita ingin membuat suatu persamaan dengan x batang, maka sisanya adalah $x-2$ batang.
- Cari jumlah batang yang mungkin untuk menghasilkan suatu nilai di ruas kiri agar sama dengan ruas kanan. Kita hanya menghitung yang total batangnya tidak lebih dari 15.

Tabel di bawah merupakan kemungkinan yang valid, dibantu dengan tabel sebelumnya.

Persamaan	Ruas Kiri	Ruas Kanan	Total Batang	Total Cara
1=1	2	10	14	2
	10	2	14	2
	2	2	6	1
2=2	5	5	12	1
	5	6	13	1
	6	5	13	1

	6	6	14	1
3=3	5	5	12	1
4=4	4	4	10	1
	4	9	15	2
	9	4	15	2
5=5	5	5	12	1
	5	8	15	2
	8	5	15	2
6=6	6	6	14	1
7=7	3	3	8	1
	3	10	15	2
	10	3	15	2
8=8	(tidak ada yang memenuhi untuk total batang <= 15)			0
9=9	6	6	14	1
0=0	6	6	14	1
Total				28

Sehingga, terdapat 28 persamaan yang bisa Pak Dengklek buat.

Jawaban: 28

15. Berdasarkan tabel di nomor sebelumnya, kita hanya perlu menambahkan yang total batangnya tepat 15, sehingga terdapat 12 cara.

Jawaban: 12

16. **Untuk benteng hitam:**

Benteng pertama (yang diletakkan di baris pertama) memiliki 6 kemungkinan.

Benteng kedua (yang diletakkan di baris kedua) memiliki 5 kemungkinan, disebabkan salah satu kolom telah terpakai oleh benteng pertama.

Begitu seterusnya hingga peletakan benteng hitam keenam. Sehingga, banyak cara peletakan 6 benteng hitam adalah $6 * 5 * 4 * 3 * 2 * 1 = 720$

Untuk benteng putih:

Untuk peletakan benteng ini, menggunakan teori “*Derangement*”, yaitu permutasi dari n elemen, yang mana posisi semua elemen itu tidak ada yang berada di posisi aslinya.

Derangement disimbolkan dengan notasi $!n$ (*subfactorials* n), yang mana $!0 = 1$, $!1 = 0$, dan memiliki persamaan $!n = (n - 1) * (!n - 1) + !(n - 2)$ untuk $n > 1$.

Karena terdapat 6 benteng maka *derangement* dari peletakan benteng-benteng ini adalah $!6$. Selanjutnya, kita cukup menghitung nilai tersebut:

- $!0 = 1$
- $!1 = 0$
- $!2 = (2 - 1) * !(2 - 1) + !(2 - 2) = (1) * (0 + 1) = 1$
- $!3 = (3 - 1) * !(3 - 1) + !(3 - 2) = (2) * (1 + 0) = 2$
- $!4 = (4 - 1) * !(4 - 1) + !(4 - 2) = (3) * (2 + 1) = 9$
- $!5 = (5 - 1) * !(5 - 1) + !(5 - 2) = (4) * (9 + 2) = 44$
- $!6 = (6 - 1) * !(6 - 1) + !(6 - 2) = (5) * (44 + 9) = 265$

Sehingga total semua kemungkinan yang bisa dilakukan adalah $6! * !6 = 720 * 265 = 190800$

Materi tentang *derangement* dapat diakses di <http://bit.ly/toki-derangement>.

Jawaban: 190800

-
17. Definisikan $f(n, x)$ sebagai banyaknya *binary string* dengan panjang n dengan akhiran x . Perhatikan bahwa fungsi ini bersifat rekursif.

Base case dari $f(n, x)$:

- $f(0, 0) = f(0, 1) = 0$
- $f(1, 0) = f(1, 1) = 1$

Rekurens dari $f(n, x)$:

- $f(n, 0) = f(n - 1, 0) + f(n - 1, 1)$
- $f(n, 1) = f(n - 1, 0) + f(n - 1, 1) - f(n - 2, 0)$

Selanjutnya, lakukan perhitungan nilai $f(n, x)$:

$$f(0, 0) = 0$$

$$f(0, 1) = 0$$

$$f(1, 0) = 1$$

$$f(1, 1) = 1$$

$$f(2, 0) = 2$$

$$f(2, 1) = 2$$

$$f(3, 0) = 4$$

$$f(3, 1) = 3$$

.

.

.

$$f(15, 0) = 1596$$

$$f(15, 1) = 987$$

Maka, banyaknya string biner 15 bit yang tidak mengandung '001' = $1596 + 987 = 2583$

Jawaban: 2583

18. Representasikan angka 1234 dalam basis 2.

$$\text{Didapatkan } 1234_{10} = 10011010010_2 = 2^1 + 2^4 + 2^6 + 2^7 + 2^{10}$$

$$\text{Sehingga, nilai } a_1 + a_2 + a_3 + a_4 + a_5 = 1 + 4 + 6 + 7 + 10 = 28$$

Jawaban: 28

19. Dari percakapan di soal, sedikit demi sedikit informasi mengenai bilangan x dan y dapat digali.

- Pak Dengklek: "Saya tidak tahu nilai kedua bilangan tersebut"
Karena Pak Dengklek hanya mengetahui hasil dari penjumlahan x dan y, maka tidak mungkin hasil penjumlahannya adalah 2 ataupun 3. Hal ini karena hasil penjumlahan 2 hanya dapat dibentuk dari 1 + 1 dan hasil penjumlahan 3 hanya dapat dibentuk dari 1 + 2. Ingat kembali bahwa x dan y adalah bilangan bulat positif.
- Bu Dengklek: "Saya juga tidak tahu"
Ini berarti angka yang diberikan ke Bu Dengklek bukan bilangan prima. Sebab, jika angka yang diberikan ke Bu Dengklek bilangan prima, maka Bu Dengklek pasti dapat menebak angka tersebut.
- Pak Dengklek: "Saya mengetahui bilangannya sekarang"
Ini berarti dari perkataan Bu Dengklek sebelumnya, Pak Dengklek mendapatkan informasi yang dapat mengeliminasi kemungkinan angka x dan y yang dimilikinya. Angka yang tereliminasi adalah pasangan bilangan yang apabila dikalikan membentuk bilangan prima. Kemungkinan yang mungkin hanyalah pasangan bilangan yang nilai x nya adalah 1 (berdasarkan definisi bilangan prima).
Karena Pak Dengklek langsung dapat menyimpulkan pasangan angka yang benar, berarti dari eliminasi yang telah disebutkan di atas hanya menyisakan satu kemungkinan pasangan angka x dan y. Dari sini dapat disimpulkan bahwa angka yang didapat Pak Dengklek hanya dapat dibentuk dari 2 pasang kemungkinan x tambah y.
Angka yang mungkin hanyalah 4 (1 + 3 dan 2 + 2) serta 5 (1 + 4 dan 2 + 3).
Namun, untuk angka 5 tidak ada pasangan kemungkinan x dan y yang dapat

dieliminasi ($\{1, 4\}$ tidak dapat dieliminasi karena $1 * 4$ bukanlah angka prima).
Dan berarti kita telah mendapatkan jawabannya yaitu $x = 2$ dan $y = 2$.

- Bu Dengklek: "Saya juga"
Bu Dengklek dapat mengobservasi percakapan tersebut sebagaimana observasi kita.

Sehingga, $5x + y = 5 * 2 + 2 = 12$.

Jawaban: 12

20. Kita dapat mencoba membuat tiap angka secara berturut-turut:

- $1 = 1$
- $2 = 1 + 1$
- $3 = 3$
- $4 = 3 + 1$
- $5 = 3 + 2 = 3 + 1 + 1$
- $6 = 6$
- $7 = 6 + 1$
- $8 = 6 + 1 + 1$
- $9 = 6 + 3$
- $10 = 6 + 4 = 6 + 3 + 1$
- $11 = 6 + 5 = 6 + 3 + 2 = 6 + 3 + 1 + 1$
- $12 = 6 + 6$
- $13 = 6 + 6 + 1$
- $14 = 6 + 6 + 2 = 6 + 6 + 1 + 1$
- $15 = 6 + 6 + 3$
- $16 = 6 + 6 + 4 = 6 + 6 + 3 + 1$
- $17 = 6 + 6 + 5 = 6 + 6 + 3 + 2 = 6 + 6 + 3 + 1 + 1$
- $18 = 6 + 6 + 6$ (tidak dapat dibuat)

Sehingga, N yang memenuhi adalah 17

Jawaban: 17

21. Untuk membentuk angka 3, tambahkan angka 1 dan 2 ke dalam sekuens. Oleh karena itu, sekuens menjadi (1, 2, 3, 4, 6, 6) dan sekuens ini dapat membentuk angka hingga 22. Apabila ditambahkan angka 23 ke dalam sekuens, dapat dibuat angka 23 sampai dengan 45. Sebab, kita dapat membuat angka 1 sampai dengan 22 dengan menggunakan sekuens (1, 2, 3, 4, 6, 6), lalu tinggal ditambahkan dengan 23. Selanjutnya, tambahkan angka 46 dengan konsep yang sama, dan dapat dibuat angka dari 46 sampai dengan 91. Karena diminta N adalah 92, cukup tambahkan 1 lagi ke sekuens. Sekuens akhir menjadi (1, 1, 2, 3, 4, 6, 6, 23, 46). Sehingga, perlu ditambahkan 5 buah angka ke dalam sekuens.

Jawaban: 5

22. Dari Tim I sampai Tim VI, hanya ada 2 pasangan tim yang dapat mengadakan rapat bersama, yaitu (Tim II - Tim III) dan (Tim III - Tim VI). Karena Tim II dan Tim VI tidak dapat mengadakan rapat bersama (Gogo berada pada kedua tim tersebut), kita hanya dapat mengambil 1 pasangan tim untuk rapat pada hari bersama. Tanpa menghilangkan keumuman, kita ambil pasangan (Tim II - Tim III). Sisa tim harus rapat pada hari yang berbeda, sehingga perlu minimal 5 hari agar sang pimpinan proyek dapat memastikan bahwa seluruh tim sudah melakukan rapat bulanan.

Jawaban: 5

23. Perhatikan bahwa Tim VII dapat melakukan rapat pada hari yang sama dengan Tim V, sehingga hari yang diperlukan tetap sama, yaitu 5 hari.

Jawaban: 5

24. Ada 3 kemungkinan pengambilan air:

- 2 liter gelas 200ml
- 2 liter gelas 500ml
- 1 liter gelas 200ml dan 1 liter gelas 500ml

Untuk kemungkinan 1 dan 2, masing-masing hanya ada 1 konfigurasi urutan gelas.

Untuk kemungkinan 3, banyaknya konfigurasi sama dengan banyak permutasi

200-200-200-200-200-500-500, yaitu $\frac{7!}{5! \cdot 2!} = 21$. Sehingga, banyaknya urutan minum adalah

$$1 + 1 + 21 = 23.$$

Jawaban: 23

25. Definisikan $f(x)$ sebagai banyak cara meminum tepat x liter.

Base case:

- $f(0) = 1$
- $f(x) = 0$ untuk setiap $x < 0$

Rekurens:

- $f(x) = f(x - 500) + f(x - 300) + f(x - 200)$

Sehingga, didapat:

- $f(200) = 1$
- $f(300) = 1$
- $f(400) = 1$
- $f(500) = 3$

- $f(600) = 2$
- $f(700) = 5$
- $f(800) = 6$
- $f(900) = 8$
- $f(1000) = 14$
- $f(1100) = 16$
- ...
- $f(2000) = 448$

Jawaban: 448

Bagian A: Pertanyaan Algoritmika (20 Pertanyaan)

26. Jawaban untuk soal ini dapat diperoleh dengan mengobservasi kegunaan dari tiap fungsi yang ada di soal.

- Fungsi klik yang melakukan pengurangan 1 terhadap nilai i .
- Fungsi klek yang mengalikan nilai dari bilangan ke- x dari array A dengan bilangan selanjutnya (apabila x adalah indeks terakhir, bilangan yang dikalikan adalah bilangan terakhir dan pertama).
- Fungsi klok yang merupakan fungsi rekursif. Pada tiap perulangannya, fungsi ini memanggil fungsi klek dengan argumen i yang dikurangkan 1 menggunakan fungsi klik. Hasil dari pemanggilan fungsi klek ini dijadikan *return value* dengan ditambahkan dengan hasil fungsi klok yang selanjutnya. Rekursi berhenti ketika i telah mencapai nilai 0.

Karena nilai i pada awalnya adalah n , atau lebih tepatnya 6, program ini memproses fungsi klek dari $x=6$ hingga $x=1$. Didapatkan nilai $8*7 + 5*6 + 4*5 + 3*4 + 2*3 + 1*2 = 118$

Jawaban: 118

$$\begin{aligned} 27. \text{proses}(11) &= \text{proses}(1) + \text{proses}(6) = 4 \\ &= 1 + \text{proses}(0) + \text{proses}(3) = 3 \\ &= 1 + 0 + \text{proses}(1) + \text{proses}(2) = 2 \\ &= 1 + 0 + 1 + \text{proses}(0) + \text{proses}(1) = 1 \\ &= 1 + 0 + 1 + 0 + 1 = 3 \end{aligned}$$

Jawaban: 3

28. Selesaikan soal ini dengan simulasi. Simulasi dapat dikerjakan dengan mudah menggunakan cara *bottom-up*.

n	0	1	2	3	4	5	6	7	8
noan(n)	0	1	2	3	5	9	16	28	49

Jawaban: 49

29. Fungsi mantaps merupakan fungsi yang mengurutkan array dari indeks pertama hingga indeks ke- n secara rekursif. Sehingga setelah pemanggilan mantaps(5) array ar bernilai {6, 4, 3, 2, 1, 7, 2, 4, 2, 1}.

Jawaban: ar = {6, 4, 3, 2, 1, 7, 2, 4, 2, 1}

30. Sebelum loop while, $x = 1$, dan $y = 0$.
Setelah loop pertama, $x = 2$, dan $y = 0 + 1 = 1$
Setelah loop kedua, $x = 4$, dan $y = 1 + 2 = 3$
Setelah loop ketiga, $x = 8$, dan $y = 3 + 4 = 7$
Setelah loop keempat, $x = 12$, dan $y = 7 + 8 = 15$
Kemudian loop berhenti karena $x > 10$. Sehingga, $y = 15$.

Jawaban: 15

31. Program ini terbagi atas 3 bagian :

```
a := -1;
for i := 1 to 10 do begin
  if a = -1 then a := i
  else if ar[i] > ar[a] then a := i;
end;
```

Bagian ini mencari indeks nilai terbesar pada array ar

```
b := -1;
for i := 1 to 10 do begin
  if i <> a then begin
    if b = -1 then b := i
    else if ar[i] > ar[b] then b := i;
  end;
end;
```

Bagian ini mencari indeks nilai terbesar kedua pada array ar. Perhatikan saat $i=a$, isi loop tidak dijalankan dan perhatikan juga pada bagian pertama, $ar[a]$ merupakan nilai terbesar dari array ar.

```
c := -1;
for i := 1 to 10 do begin
  if (i <> a) and (i <> b) then begin
    if c = -1 then c := i
    else if ar[i] > ar[c] then c := i;
  end;
end;
```

Bagian ini mencari indeks nilai terbesar ketiga pada array ar.
Sehingga, jawabannya adalah 8 5 7

Jawaban: 8 5 7

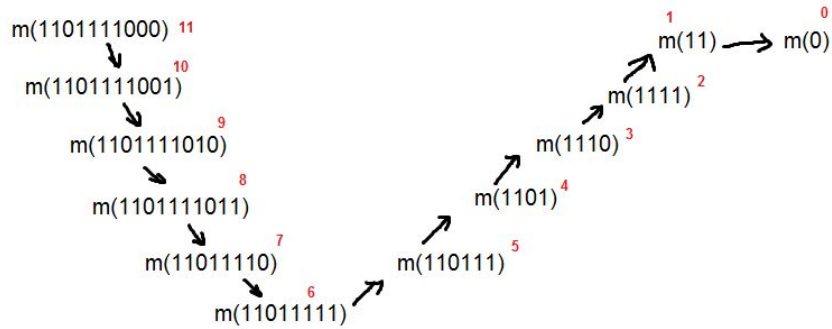
32. Soal ini dapat diselesaikan dengan cara simulasi. Proses simulasi dapat lebih mudah dilakukan apabila menggunakan representasi biner.

Perhatikan bahwa “if (x mod 2 = 1 and (x div 2) mod 2 = 1)” sama saja dengan memeriksa apakah 2 digit terakhir dari x dalam representasi biner adalah 11.

Kemudian, (x div 2) div 2 adalah menghilangkan 2 digit terakhir dari x dalam representasi biner.

Representasi biner dari 888 adalah 1101111000.

Alur rekursi adalah sebagai berikut:



Sehingga, meong(888) akan menghasilkan 11.

Jawaban: 11

-
33. Penyelesaian soal ini dapat menggunakan simulasi. Pada setiap iterasi pada while, nilai dari array 'arr' indeks pertama dibandingkan dengan 'arr' indeks ke-m yang mana m adalah indeks tengah dari rentang yang di tinjau. Rentang yang ditinjau dibatasi dengan variabel left dan right. Apabila nilai bilangan di indeks pertama lebih kecil atau sama dengan bilangan di indeks ke-m, maka rentang akan diperkecil dengan menggeser nilai left ke indeks m + 1. Sebaliknya, rentang diperkecil dengan menggeser nilai right ke indeks m apabila indeks pertama lebih besar. Hal ini dilakukan hingga left dan right bernilai sama. Pada akhirnya, didapatkan nilai keluaran fungsi get adalah 6.

Jawaban: 6

-
34. Pada awalnya, isi setiap elemen pada array a bernilai false. kemudian pada iterasi berikutnya:
- Pada saat $i = 1$, semua nilai elemen pada array a dinegasikan, yaitu dari nilai false menjadi true.
 - Pada saat $i = 2$, semua nilai elemen pada array a yang mempunyai indeks kelipatan 2 dinegasikan, yaitu dari nilai true menjadi nilai false.
 - Pada saat $i = 3$, semua nilai elemen pada array a yang mempunyai indeks kelipatan 3 dinegasikan, yaitu apabila true menjadi false, dan jika false menjadi true.
 - ...dan seterusnya sampai pada saat $i = 1000$.

Observasi:

- Apabila suatu nilai boolean dinegasikan sebanyak 2 kali, maka nilai boolean tersebut akan kembali ke nilai awal sebelum ia dinegasikan. Maka, apabila suatu nilai boolean dinegasikan sebanyak X kali yang mana X adalah bilangan genap, nilai boolean tersebut akan sama dengan nilai awal sebelum diberikan operasi negasi.
- Nilai akhir dari suatu elemen pada array a bergantung dengan banyak faktor dari indeks elemen tersebut.
- Apabila banyak faktor dari nilai indeks suatu elemen adalah genap, berarti nilai akhir dari elemen tersebut adalah false. Jika tidak, nilai akhir dari elemen tersebut adalah true.
- Bilangan kuadrat sempurna adalah satu-satunya bilangan yang banyak faktornya ganjil. Maka, elemen pada indeks yang bernilai kuadrat sempurna mempunyai nilai akhir true.

Banyak elemen yang mempunyai nilai akhir true adalah $\lfloor \sqrt[3]{1000} \rfloor = 31$.

Jawaban: 31

35. Kita bisa mensimulasikan pemanggilan fungsi X(2, 6, 4):

1. X(2, 6, 4):
 - temp = (2 + 6) div 2 = 4
 - X = X(2, 4, 4)
2. X(2, 4, 4):
 - temp = (2 + 4) div 2 = 3
 - X = X(2, 3, 4)
3. X(2, 3, 4):
 - temp = (2 + 3) div 2 = 2
 - X = X(3, 3, 4)
4. X(3, 3, 4):
 - X = isi[3] = 7

Sehingga, X(2, 6, 4) akan menghasilkan 7.

Jawaban: 7

36. Seperti nomor sebelumnya, kita bisa mensimulasikan pemanggilan X(1, 10, 0):

1. X(1, 10, 0):
 - temp = (1 + 10) div 2 = 5
 - X = X(6, 10, 0)
2. X(6, 10, 0)
 - temp = (6 + 10) div 2 = 8

- $X = X(6, 8, 0)$
- 3. $X(6, 8, 0)$
 - $\text{temp} = (6 + 8) \text{ div } 2 = 7$
 - $X = X(6, 7, 0)$
- 4. $X(6, 7, 0)$
 - $\text{temp} = (6+7) \text{ div } 2 = 6$
 - $X = X(6, 6, 0)$
- 5. $X(6, 6, 0)$
 - $X = \text{isi}[6] = 1$

Jadi, total pemanggilan fungsi X adalah 5 kali.

Catatan: potongan program ini sebenarnya adalah implementasi dari *binary search*, untuk mencari nilai pertama yang lebih dari v (jika array-nya terurut menaik)

Jawaban: 5

37. Observasi:

- Procedure oppa mengisi array dengan karakter 'T', 'O', 'K', dan 'I' secara berturut-turut.
- Procedure anyeong akan mengambil arr_sz karakter pertama array daebak dan membagi menjadi beberapa bagian dengan ukuran part_sz untuk diproses saranghae
- Procedure saranghae akan menggeser dari bagian yang diberikan procedure anyeong ke kanan 1 indeks, dan array terkanan akan dipindah ke kiri.
- Di program utama, procedure anyeong akan dijalankan sebanyak x kali.

Jika input adalah "16 2 12 4", kita akan mengambil 12 karakter pertama, membaginya menjadi 3 bagian dengan panjang 4:
(TOKI TOKI TOKI) TOKI

Bila kita jalankan prosedur saranghae, array akan menjadi:
(OKIT OKIT OKIT) TOKI

Kemudian, karena $x = 2$, array akan diputar sekali lagi, menghasilkan:
(KITO KITO KITO) TOKI

Sehingga, output sesuai program utama adalah KITO KITO KITO TOKI.

Jawaban: KITO KITO KITO TOKI

38. Sesuai penjelasan nomor 37, jika input adalah "16 4 10 5", kita akan mengambil 10 karakter pertama, membaginya menjadi 2 bagian dengan panjang 5:
(TOKIT OKITO) KITOKI

Kemudian prosedur saranghae dijalankan 4 kali sehingga menjadi:
(OKITT KITOO) KITOKI

Sehingga, output sesuai program utama adalah OKIT TKIT OOKI TOKI.

Jawaban: OKIT TKIT OOKI TOKI

39. Pada dasarnya, program ini akan mengiterasi semua karakter dalam string s. Namun, karakter yang diambil untuk dimasukkan variabel temp dimulai dari karakter r (paling akhir), 1 (paling awal), r-1 (kedua paling akhir), dst hingga bertemu di tengah.

Setiap karakter akan dikurangi ord('A') lalu ditambahkan ke x, sehingga karakter A akan menambahkan 0 pada x, B menambah 1, hingga Z menambah 25. Lalu, x akan dimodulo 26.

Terakhir, string s akan disambung dengan karakter x (yang mana 0 adalah A, 1 adalah B, dst)

Iterasi yang terjadi adalah sebagai berikut:

temp	l	r	k	x	string t
	1	9	0	0	
E		8	1	$(0 + 4) \bmod 26 = 4$	E
I	2		2	$(4 + 8) \bmod 26 = 12$	EM
C		7	3	$(12 + 2) \bmod 26 = 14$	EMO
X	3		4	$(14 + 23) \bmod 26 = 11$	EMOL
V		6	5	$(11 + 21) \bmod 26 = 6$	EMOLG
Y	4		6	$(6 + 24) \bmod 26 = 4$	EMOLGE
J		5	7	$(4 + 9) \bmod 26 = 13$	EMOLGEN
B	5		8	$(13 + 1) \bmod 26 = 14$	EMOLGENO
E		4	9	$(14 + 4) \bmod 26 = 18$	EMOLGENOS

Jawaban: EMOLGENOS

40. Untuk mencari input yang menghasilkan "RJJJVDPPII", iterasi string output dari kiri. Setiap huruf memiliki nilai x yang menghasilkan huruf tersebut, yakni 0 untuk A, 1 untuk B, dst. Nilai dari $\text{ord}(\text{temp}) - \text{ord}('A')$ akan menambahkan x, sehingga bernilai selisih dari

nilai x dengan nilai x sebelumnya, kemudian dimodulo 26 menjadi bilangan non-negatif. Kita cukup mencari karakter tiap temp, yang mana 0 merupakan A, 1 merupakan B, dst.

Huruf baru di string s	Nilai x yang menyebabkan huruf tsb	ord(temp) yang menyebabkan nilai x tsb	Karakter temp
R	17	17	R
J	9	$(9+26-17) \bmod 26 = 18$	S
J	9	$(9+26-9) \bmod 26 = 0$	A
J	9	$(9+26-9) \bmod 26 = 0$	A
V	21	$(21+26-9) \bmod 26 = 12$	M
D	3	$(3+26-21) \bmod 26 = 8$	I
P	15	$(15+26-3) \bmod 26 = 12$	M
I	8	$(8+26-15) \bmod 26 = 19$	T
I	8	$(8+26-8) \bmod 26 = 0$	A

Agar urutan pengambilan karakter temp sesuai algoritma soal, taruh karakter dari paling kanan, lalu kiri, kedua dari kanan, dan seterusnya hingga selesai di tengah, yakni "SAITAMMAR".

Jawaban: SAITAMMAR

41. Pada for loop yang pertama, untuk setiap $1 \leq i \leq 15$ program menyimpan jumlah dari $Q[1]$ hingga $Q[i]$ pada array P dengan indeks i ($P[i]$). Hal ini biasa disebut sebagai *prefix sum*. Kemudian, pada bagian kedua, program menyimpan total penjumlahan dari rentang a sampai b (inklusif). Jadi, apabila nilai $a = 6$ dan $b = 10$ maka akan menghasilkan jumlah elemen array Q dari indeks 6 sampai 10, yaitu $6 + 2 + 7 + 3 + 4 = 22$

Materi tentang *prefix sum* dapat diakses di <http://bit.ly/toki-prefix-sum>.

Jawaban: 22

42. Karena kita telah memiliki array yang berfungsi sebagai *prefix sum* untuk array Q, maka kita dapat mengetahui jumlah suatu *subsequence* konsekutif dengan menggunakan *prefix sum* dari batas akhir dikurangi *prefix sum* dari indeks sebelum batas awal.

Maka, baris 1-4 dapat diganti dengan 1 baris yang berisi:

$$\text{total} := P[b] - P[a - 1]$$

Jawaban: total := P[b] - P[a - 1]

Berikut adalah pembahasan untuk nomor 43-44

Terdapat 2 fungsi yang terdapat di kedua nomor tersebut:

- Whatsup
 - Fungsi ini akan menukar elemen di indeks ke-l dengan elemen di indeks ke-r
- naooon
 - Perhatikan bahwa setelah iterasi di dalam fungsi, semua elemen pada rentang [l, r] yang bernilai > 'ini' akan diletakkan di l, l + 1, sampai kanan, sedangkan yang bernilai < 'ini' akan diletakkan di kanan + 1, kanan + 2, sampai r.
 - Setelah itu dilakukan pemanggilan rekursif naooon
 - Dapat disimpulkan bahwa naooon akan mengurutkan elemen pada rentang [l, r] secara menurun (dari terbesar ke terkecil)

43. naooooon(1, 10) akan mengurutkan menurun pada rentang [1, 10]. Maka, setelah pemanggilan tersebut, isi menjadi [100, 29, 18, 10, 9, 8, 7, 5, 3, 1]. Maka, outputnya adalah 100 29 18 10 9 8 7 5 3 1

Jawaban: 100 29 18 10 9 8 7 5 3 1

44. naooooon(3, 6) akan mengurutkan menurun pada rentang [3, 6]. Maka, setelah pemanggilan tersebut, isi menjadi [5, 10, 18, 9, 7, 1, 3, 8, 100, 29]. Maka, outputnya adalah 5 10 18 9 7 1 3 8 100 29

Jawaban: 5 10 18 9 7 1 3 8 100 29

45. Perhatikan bahwa procedure S akan menukar elemen ar[a] dan ar[b]. Untuk menyelesaikan soal ini, kita bisa awalnya membuat suatu array A, yang mana A[i] menyatakan indeks pada array asli yang pada array akhir menempati indeks ke-i. Awalnya, A = [1, 2, 3, 4, 5].

- S(3, 4) menyebabkan A menjadi [1, 2, 4, 3, 5]
- S(4, 1) menyebabkan A menjadi [3, 2, 4, 1, 5]
- S(5, 2) menyebabkan A menjadi [3, 5, 4, 1, 2]
- S(5, 1) menyebabkan A menjadi [2, 5, 4, 1, 3]

Maka, agar array akhir ar menjadi [1, 2, 3, 4, 5]:

- ar[2] = 1
- ar[5] = 2
- ar[4] = 3
- ar[1] = 4

- $ar[3] = 5$

Sehingga, awalnya ar berisi [4, 1, 5, 3, 2]. Maka, inputnya 4 1 5 3 2

Jawaban: 4 1 5 3 2

Bagian B: Membuat Program/Algoritmika Sederhana (3 Pertanyaan)

1. Observasi:

- (1) Satu-satunya bilangan prima yang bukan ganjil adalah 2.
- (2) Penjumlahan dua bilangan ganjil menghasilkan bilangan genap.
- (3) Penjumlahan bilangan ganjil dengan bilangan genap menghasilkan bilangan ganjil.
- (4) Kita tidak dapat menjumlahkan 2 bilangan prima yang ganjil untuk mendapatkan bilangan prima (lihat observasi (2))
- (5) Salah satu cara untuk menjumlahkan 2 bilangan prima untuk mendapatkan bilangan prima adalah menjumlahkan bilangan prima genap (2) dengan bilangan prima lainnya.

Kita dapat lakukan hal berikut:

- (1) Cek apakah terdapat angka 2 pada input bilangan prima.
- (2) Jika tidak, langsung tampilkan "TIDAK ADA".
- (3) Jika ya, cek apakah ada bilangan prima lain yang apabila dijumlahkan dengan 2, akan menghasilkan bilangan prima juga. Pengecekan bilangan prima bisa dilakukan dengan kompleksitas $O(\sqrt{N})$.
- (4) Jika ada, tampilkan "ADA".
- (5) Jika tidak, tampilkan "TIDAK ADA".

Kompleksitas total adalah $O(N \sqrt{N})$.

Contoh implementasi terdapat di halaman selanjutnya.

```

var
    N:longint;
    P:array[1..100000] of longint;
    i:longint;
    adaDua:boolean;
    adaPrimaValid:boolean;

function cekPrima(x:longint): boolean;
var
    i:longint;
    hasil:boolean;
begin
    i:=2; hasil:=true;
    while (i*i <= x) do begin
        if (x mod i = 0) then begin
            hasil:=false;
            break;
        end;
        i:=i+1;
    end;
    cekPrima:=hasil;
end;

begin
    readln(N);
    for i:=1 to N do begin
        read(P[i]);
        if (P[i] = 2) then adaDua:=true;
    end;
    if (not adaDua) then writeln('TIDAK ADA')
    else begin
        for i:=1 to N do begin
            if cekPrima(P[i]+2) then begin
                adaPrimaValid:=true;
                break;
            end;
        end;
        if (adaPrimaValid) then writeln('ADA')
        else writeln('TIDAK ADA')
    end;
end.

```

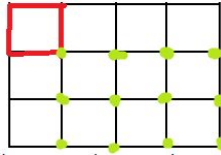
2. Sebelumnya, untuk persegi panjang berukuran 3 x 4:

- Terdapat (3 x 4) persegi berukuran 1 x 1.

Catatan: Terdapat (3x4) cara menempatkan persegi berukuran 1 x 1, salah satu cara

menghitungnya adalah mencari ada berapa titik yang bisa ditempatkan sebagai titik pojok

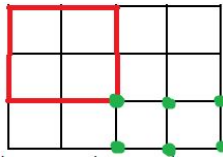
kanan bawah persegi tersebut.



- Terdapat $((3 - 1) \times (4 - 1)) = 2 \times 3$ persegi berukuran 2 x 2.

Catatan: Sama seperti persegi ukuran 1 x 1, kita dapat mencari ada berapa titik yang bisa

ditempatkan sebagai titik pojok kanan bawah persegi.



- Terdapat $((3 - 2) \times (4 - 2)) = 1 \times 2$ persegi berukuran 3 x 3
- Persegi terbesar yang dapat kita bentuk adalah 3 x 3.
- Terdapat $(3 \times 4) + (2 \times 3) + (1 \times 2) = 20$ persegi dalam persegi panjang

berukuran 3 x 4.

Untuk persegi panjang berukuran NxM, persegi terbesar yang dapat kita bentuk adalah persegi

berukuran A x A dengan $A = \min(N, M)$. Pada contoh 3 x 4 terlihat pola bahwa banyak persegi

adalah $N \times M + (N - 1) \times (M - 1) + (N - 2) \times (M - 2) + \dots + (N - A + 1) \times (M - A + 1)$.

Contoh implementasi:

```
var
    n,m,banyakPersegi: Int64;
    i,minimum: longint;

function min(a,b: Int64): longint;
begin
    if (a<b) then min:=a
    else min:=b;
end;

begin
    readln(n,m);
    minimum:=min(n,m);
    banyakPersegi:=0;
```

```
for i:=1 to minimum do banyakPersegi:=banyakPersegi+((n-i+1)*(m-i+1));
writeln(banyakPersegi);
end.
```

3. Untuk memeriksa suatu string adalah palindrom atau tidak, kita cukup mengecek apakah karakter pertama sama dengan karakter ke-N, karakter kedua sama dengan karakter ke-(N - 1), karakter ketiga sama dengan karakter ke-(N - 2), dan seterusnya sampai mengecek karakter ke-(N / 2) sama dengan karakter ke-(N - N / 2).

Contoh implementasi:

```
var
  n,i:integer;
  s:string;
  palindrom:boolean;
begin
  readln(n);
  readln(s);
  palindrom:=true;
  for i:=1 to (n div 2) do begin
    if (s[i] <> s[n - i + 1]) then begin
      palindrom:=false;
      break;
    end;
  end;
  if (not palindrom) then writeln('BUKAN PALINDROM')
  else writeln('PALINDROM');
end.
```