

INVOSA SYSTEMS ONLINE TEST

version 2017

PETUNJUK

Buatlah program yang bisa menyelesaikan soal – soal di bawah.

Petunjuk:

- Batas waktu maksimal 1 minggu setelah download soal ini
- Kerjakan soal-soal tsb menggunakan bahasa pemrograman yang anda kuasai
Misalnya: PHP, Python, C/C++, Java, VB, Delphi, dll
- Submit source code ke onlinetest@invosa.com
Dengan Subject: <Nama anda>, Soal no: <x, y, z, ...>
Contoh Subject: Agus Budi, Soal no: 1,2,3
- Source code bisa disatukan dan dikirim dalam satu email, ataupun
Setiap program jawaban dikirim dalam email yang berbeda.
- Soal wajib dan soal optional:
 - ✓ Basic Level : Seluruh soal mandatory untuk dijawab
 - ✓ Intermediate Level : Minimal 2 soal mandatory untuk dijawab
1 soal optional untuk dijawab
 - ✓ Advanced Level : Optional untuk dijawab (merupakan nilai plus)

BASIC LEVEL

1. BUNGA BANK

1.1. Memakai Loop / Recursion

Pak Adi menabung di bank sebanyak Rp 1 juta dan mendapatkan bunga 1% perbulan. Setelah b bulan berapakah saldo tabungannya.

Input : b bulan

Output : s saldo tabungan

Contoh:

- Input : 0
Output : 1,000,000.00
- Input : 1
Output : 1,010,000.00
- Input : 12
Output : 1,126,825.03

- d. Input : 60
Output : 1,816,696.70
- e. Input : 360
Output : 35,949,641.33

1.2. Tanpa Loop

Buatlah program seperti no 1.1. di atas tetapi **TANPA** menggunakan loop, recursion atau sejenisnya.

2. DISKON BERTINGKAT

Buatlah program yang bisa menghitung harga setelah diskon, yang mana diskon bisa bertingkat.

Contoh diskon bertingkat:

50% + 20% Diskon 50%, kemudian diskon lagi 20%

30% + 20% + 10% Diskon 30%, setelah itu 20%, kemudian 10%

Harga barang ditetapkan Rp 100,000 (seratus ribu rupiah).

Input : 1. Jumlah tingkat diskon (integer positif)
2. Diskon-diskon (0 - 100)

Output : Harga setelah diskon.

Contoh:

a. Input 1 : 1 (Diskon 1 tingkat)
Input 2 : 30 (Tingkat 1, diskon 30%)
Output : Rp 70,000

b. Input 1 : 2 (Diskon 2 tingkat)
Input 2 : a. 50 (diskon 50%)
b. 20 (diskon 20%)
Output : Rp 40,000

c. Input 1 : 3 (Diskon 3 tingkat)
Input 2 : a. 40 (diskon 40%)
b. 20 (diskon 20%)
c. 50 (diskon 50%)
Output : Rp 24,000

3. ENCODING / DECODING

A. Tabel di bawah ini adalah kode ASCII nomor 64 – 95.

| | | | | | | | | | | | | | | | | |
|----------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ASCII 64 – 79: | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| ASCII 80 – 95: | P | Q | R | S | T | U | V | W | X | Y | Z | [| \ |] | ^ | _ |

Buatlah program yang bisa meng-encode (mengkodekan) 32 kode ASCII tsb, dengan mengikuti aturan seperti tabel di bawah:

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| P | Q | R | S | T | U | V | W | X | Y | Z | [| \ |] | ^ | _ |

Aturan:

- Jika 'A' menjadi 'Q', jika 'B' menjadi 'R', jika 'C', menjadi 'S', dst
- Demikian juga sebaliknya, jika 'Q' menjadi 'A', jika 'R' menjadi 'B', dst
- Jika input merupakan huruf kecil, maka dianggap huruf besar (case insensitive)
- Dilarang menggunakan syntax “if”, “switch”, atau sejenisnya.

Contoh Input:

Saya_mau_makan

Output:

CQIQO]QEO]Q[Q^

INTERMEDIATE LEVEL

4. KONVERSI ANGKA ROMAWI KE ANGKA ARAB

Buatlah program yang bisa mengkonversi dari angka Romawi (I, V, X, L, C, D, M) ke angka Arab (1, 2, 3, ...). Batasan maksimal yang bisa dikonversikan adalah 3999.

Input : Angka Romawi, bisa huruf besar atau kecil (case insensitive)

Output : Angka Arab

Contoh:

- a. Input : VI atau vi
Output : 6
- b. Input : XCIX atau xcix
Output : 99
- c. Input : DCCCLXXXVIII atau dcccclxxxviii
Output : 888

5. SHORTEST PATH PROBLEM

Perhatikan segitiga di bawah ini. Masing-masing kotak dihubungkan oleh tanda panah, yang melambangkan rute yang boleh dilalui dari satu kotak ke kotak lain. Start dari rute tersebut adalah baris teratas yaitu kotak n11, dan finishnya adalah baris terbawah (nx1 – nxx).

Carilah rute terpendek dari segitiga di bawah ini. Panjang rute ditentukan oleh jumlah dari semua angka yang ada di dalam kotak yang dilewati.

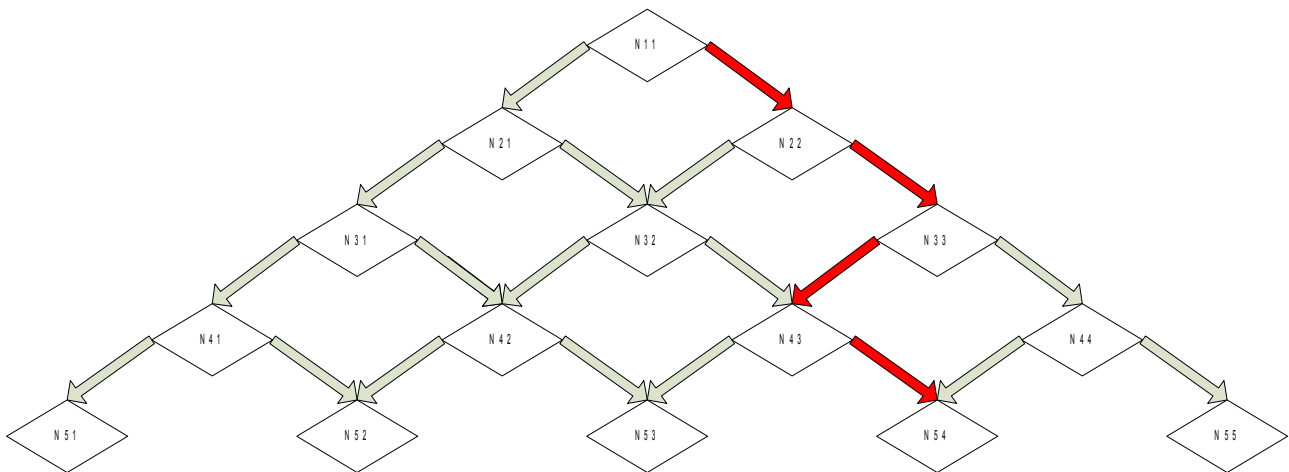
Perhatian:

- 1. Jumlah baris fleksibel, diinput oleh user (tidak selalu 5)
- 2. Angka-angka di dalam kotak (n11 – nxx) diinput oleh user secara bebas.

Input:

- 1. Jumlah baris
- 2. n11 – nxx (x = jumlah baris)

| | | | | | |
|---------|-----|-----|-----|-----|-----|
| Baris 1 | n11 | | | | |
| Baris 2 | n21 | n22 | | | |
| Baris 3 | n31 | n32 | n33 | | |
| Baris 4 | n41 | n42 | n43 | n44 | |
| Baris 5 | n51 | n52 | n53 | n54 | n55 |



Output:

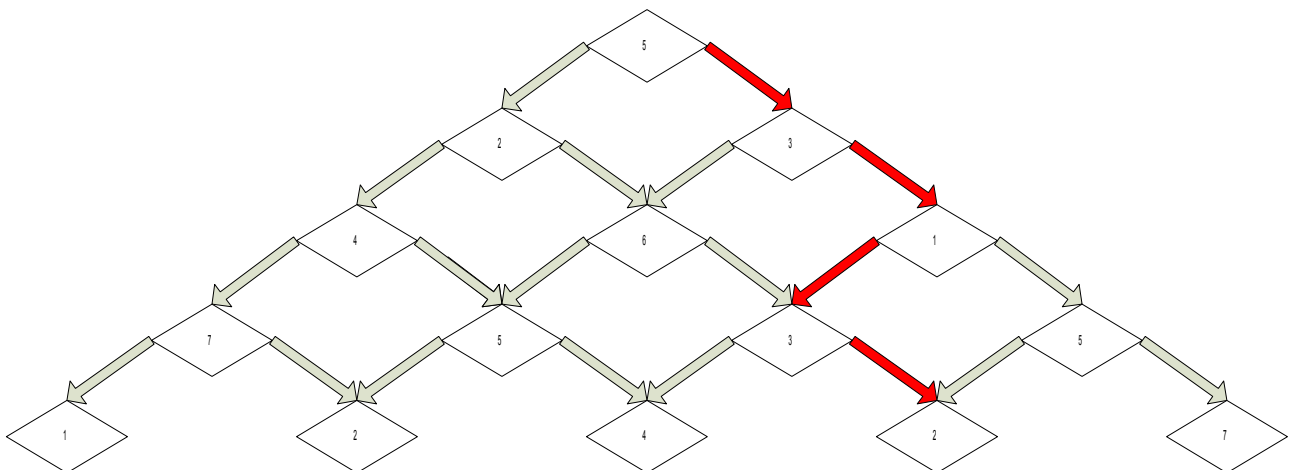
Path/rute dari baris paling atas sampai baris terbawah dengan jumlah terkecil.

Contoh:

Input 1 : 5

Input 2 : Tabel di bawah

| | | | | | |
|---------|---|---|---|---|---|
| Baris 1 | 5 | | | | |
| Baris 2 | 2 | 3 | | | |
| Baris 3 | 4 | 6 | 1 | | |
| Baris 4 | 7 | 5 | 3 | 5 | |
| Baris 5 | 1 | 2 | 4 | 2 | 7 |



Output :

$$\begin{aligned}
 \text{Path terpendek} &= n11 + n22 + n33 + n43 + n54 \\
 &= 5 + 3 + 1 + 3 + 2 \\
 &= 14
 \end{aligned}$$

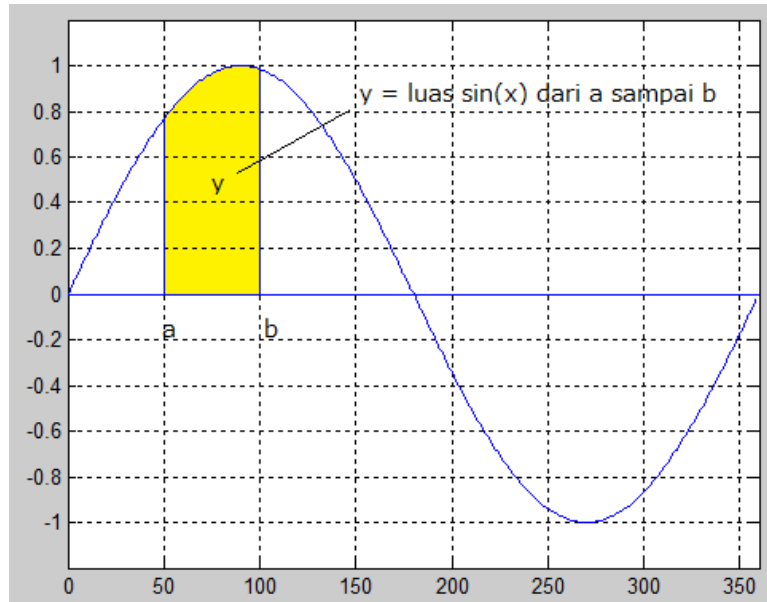
Jika bisa, buatlah menggunakan Class (Object Oriented).

6. LUAS KURVA SIN

Buatlah program yang bisa menghitung luas kurva fungsi sin (sinus) seperti pada gambar di bawah.

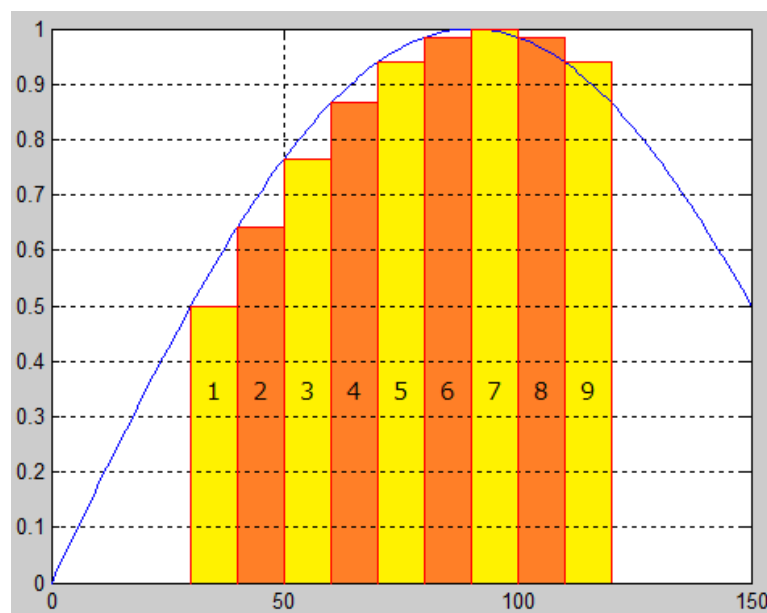
Input : a dan b (dalam derajat, 0 s/d 360)

Output : y adalah luas dari a sampai b



Hint: Cara Menghitung Luas

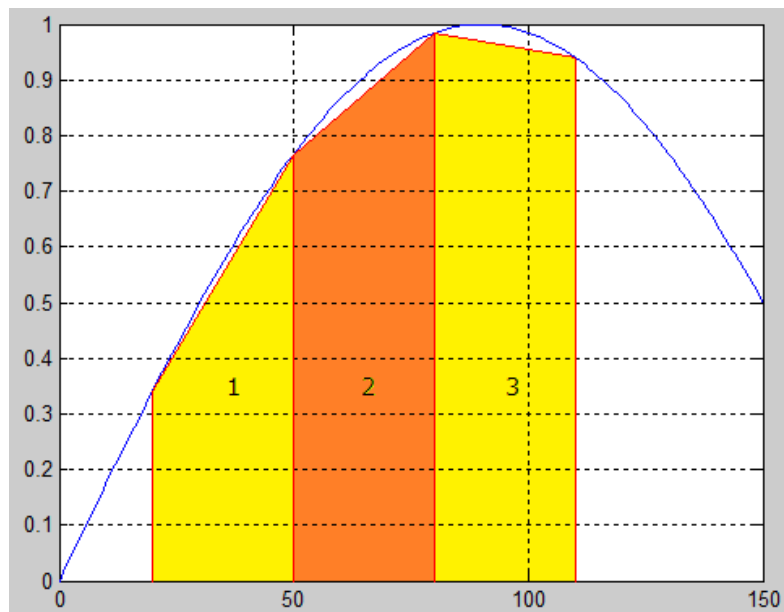
Bagilah area tersebut menjadi beberapa area persegi panjang, kemudian hitung luas masing2 persegi panjang dan jumlahkan semuanya. Lihat gambar di bawah.



Luas = luas 1 + luas 2 + luas 3 + ... + luas 9

Note: Semakin kecil pembagian area persegi panjang, akan menghasilkan hasil yang lebih presisi.

Atau jika ingin lebih presisi, bagilah area luas menjadi beberapa trapezium, kemudian hitung luas masing2 trapezium dan jumlahkan semuanya. Lihat gambar di bawah.



Luas = trapezium 1 + trapezium 2 + trapezium 3

Contoh:

Untuk referensi, dibawah ini adalah tabel a, b, dan luas y:

| a | b | y |
|-----|-----|-------|
| 0 | 90 | 57.3 |
| 0 | 180 | 114.6 |
| 0 | 360 | 0 |
| 150 | 210 | 0 |
| 45 | 90 | 40.5 |
| 90 | 180 | 57.3 |
| 90 | 360 | -57.3 |

Note:

- π = simbol pi (3.14159)
- Di internal komputer semua perhitungan menggunakan satuan radian (bukan derajat)

ADVANCED LEVEL

7. PREDIKSI JUMLAH KELERENG

Di dalam sebuah kotak ada kelereng sejumlah **Z** butir.

- Kelereng berwarna **kuning** berjumlah **X** butir
- Kelereng berwarna **biru** berjumlah **Y** butir
- Total kelereng **Z** butir

Ali membuka kotak tersebut dan mengeluarkan kelereng satu-persatu sambil menghitung jumlah kelereng kuning dan kelereng biru.

Buatlah program yang bisa memprediksi jumlah akhir kelereng kuning dan biru, sebelum semua kelereng dikeluarkan.

Input : 1. Jumlah total kelereng **Z**
2. Warna kelereng yg keluar (looping **Z** kali)

Output : Prediksi jumlah akhir kelereng kuning dan biru (looping **Z** kali)
Output tampil setiap satu kelereng keluar

Contoh :

Masukkan jumlah kelereng: 100

(*k: kuning, b: biru*)

| No | Input (Warna kelereng yg keluar) | Output (Prediksi akhir jumlah kelereng) |
|---------|----------------------------------|--|
| 1 | Masukkan warna kelereng 1: k | Prediksi jumlah kelereng k: 100 Prediksi jumlah kelereng b: 0 |
| 2 | Masukkan warna kelereng 2: k | Prediksi jumlah kelereng k: 100 Prediksi jumlah kelereng b: 0 |
| 3 | Masukkan warna kelereng 3: b | Prediksi jumlah kelereng k: 70 Prediksi jumlah kelereng b: 30 |
| 4 | Masukkan warna kelereng 4: b | Prediksi jumlah kelereng k: 50 Prediksi jumlah kelereng b: 50 |
| 5 | Masukkan warna kelereng 5: k | Prediksi jumlah kelereng k: 58 Prediksi jumlah kelereng b: 42 |
| 6 | Masukkan warna kelereng 6: k | Prediksi jumlah kelereng k: 63 Prediksi jumlah kelereng b: 37 |
| ... dst | ... dst | ... dst |
| 100 | Masukkan warna kelereng 100: k | Prediksi jumlah kelereng k: 67 Prediksi jumlah kelereng b: 33 |

Note:

1. Output di atas hanya merupakan contoh, bukan jawaban
2. Output yang dihasilkan adalah pendekatan (prediksi). Semakin tepat prediksi dengan sample (loop) yg sesedikit mungkin menandakan algoritmanya semakin bagus.
3. Asumsinya warna kelereng yg keluar adalah random, tidak diatur dengan sengaja misalnya keluar diawal 50 butir kuning, kemudian setelahnya 50 butir biru.

8. PENGATURAN LIFT

Ada 3 buah lift yang bekerja dalam sebuah gedung 8 lantai. Lift-lift tersebut dikontrol oleh sebuah komputer. Buatlah program yang dapat menjemput pengguna lift sedemikian sehingga akumulasi waktu tunggu minimum.

Untuk memudahkan algoritma, dibuat alur dan aturan sbb:

- Semua lift dalam keadaan standby masing-masing di suatu lantai (bisa di lantai yg sama maupun lantai yg berbeda).
- Ada 4 orang pengguna lift yang masing-masing sedang berada di suatu lantai (bisa di lantai yg sama maupun lantai yg berbeda).
- Empat orang tsb memencet tombol lift naik (↑) atau turun (↓) secara bersamaan.
- Semua orang yang memencet tombol naik (↑) bertujuan ke lantai 8, dan yg memencet tombol turun (↓) bertujuan ke lantai 1.
- Dua orang atau lebih dengan arah yg berlawanan tidak dapat berada dalam lift yg sama. Artinya, satu lift tidak dapat menjemput 2 orang yg masing-masing memencet tombol naik (↑) dan turun (↓).
- Waktu tempuh lift dari satu lantai ke lantai berikutnya adalah 1 detik.
- Waktu yg diperlukan untuk menaikkan/menurunkan orang adalah 1 detik.

Contoh:

Lift dari lantai 1 ke lantai 8 dengan terlebih dahulu menjemput orang di lantai 2 dan 7 membutuhkan waktu 9 detik.

Contoh Input & Output:

Input:

Posisi Lift

| | |
|------------------------|-----|
| Masukkan posisi lift 1 | : 7 |
| Masukkan posisi lift 2 | : 5 |
| Masukkan posisi lift 3 | : 3 |

Posisi dan Tujuan Orang (n=naik, t=turun)

| | |
|---------------------------------|--------|
| Masukkan posisi dan tujuan si A | : 2, n |
| Masukkan posisi dan tujuan si B | : 2, t |
| Masukkan posisi dan tujuan si C | : 6, t |
| Masukkan posisi dan tujuan si D | : 5, n |

Output:

Waktu tunggu

si A : 5 detik

si B : 1 detik

si C : 1 detik

si D : 0 detik

Akumulasi waktu tunggu minimum: 7 detik

Happy programming