

SOAL BABAK PENYISIHAN TAHAP II
Informatic Logical Programming Competition
2010



29 Januari 2010

Jurusan Informatika – Fakultas Teknik

Universitas Surabaya

Peraturan Babak Penyisihan ILPC 2010 Hari Kedua

1. Babak penyisihan ILPC hari kedua bersifat **ONLINE**.
2. Soal terdiri atas 10 soal programming dengan total waktu pengerjaan soal 6 jam (09.00 - 15.00 WIB).
3. Soal akan diberikan dalam bentuk Portable Document Format (PDF).
4. Bahasa yang digunakan di dalam soal ini adalah Bahasa Indonesia.
5. Peserta dapat memilih bahasa pemrograman berikut ini untuk membuat program berdasarkan soal yang telah diberikan.
 - MS Visual C++
 - Dev-Cpp
 - Free Pascal
 - Turbo Pascal 7
 - MS VB 6.0 / .NET
 - Java Netbeans 6.0
6. Setiap jawaban harus dikirimkan melalui email (ilpc@if.ubaya.ac.id). Jawaban yang dikirimkan adalah berupa file penyelesaian soal yang sudah dikerjakan dengan salah satu bahasa pemrograman yang dipilih. Format penamaan file adalah sebagai berikut :
 - Apabila menggunakan bahasa pemrograman MS Visual C++ dan Dev-Cpp maka penamaan file adalah "[nama_tim]-[kode_soal].cpp"(tanpa tanda kutip). Misalkan nama tim anda adalah tim1, dan soal yang dikerjakan adalah Permasalahan A, maka penamaan filenya adalah tim1-A.cpp.
 - Apabila menggunakan bahasa pemrograman Free Pascal dan Turbo Pascal maka penamaan file adalah "[nama_tim]-[kode_soal].pas"(tanpa tanda kutip). Misalkan nama tim anda adalah tim1, dan soal yang dikerjakan adalah Permasalahan A, maka penamaan filenya adalah tim1-A.pas.
 - Apabila menggunakan bahasa pemrograman MS VB 6.0/.NET dan Java Netbeans 6.0 maka folder project dikirimkan dalam bentuk zip file dengan penamaan "[nama_tim]-[kode_soal].zip"(tanpa tanda kutip). Misalkan nama tim anda adalah tim1, dan soal yang dikerjakan adalah Permasalahan A, maka penamaan filenya adalah tim1-A.zip.
7. Sistem penilaian adalah sebagai berikut :
 - Penilaian hanya dilakukan dengan sistem benar-salah dan real-time.
 - Jika jawaban benar, maka waktu pengumpulan jawaban benar itu akan dicatat dan diakumulasikan dengan waktu pengumpulan jawaban benar untuk soal lainnya.
 - Jika jawaban salah, waktu pengumpulan tidak akan dicatat, namun akan ditambahkan 20 menit (untuk setiap pengumpulan jawaban salah) pada saat jawaban yang benar untuk soal tersebut dikumpulkan.
 - Peserta diperbolehkan mengirimkan jawaban lagi, jika jawaban yang dikirimkan sebelumnya salah.
 - Peringkat ditentukan dengan banyaknya soal yang telah dijawab dengan benar dan dengan berdasarkan total waktu tercepat.
 - Papan skor dapat dilihat pada web ilpc.ubaya.ac.id.
 - Papan skor akan di-update secara real-time mulai dari saat penyisihan dimulai hingga 1 jam terakhir.
 - Pada 1 jam terakhir, papan skor tidak akan di-update lagi (freeze), dan akan kembali di-update setelah babak penyisihan selesai untuk menampilkan hasil akhir.

8. Sebagai catatan, untuk pencetakan output, boleh dilakukan per test case. Jadi, misalkan pada program terdapat 5 test case, maka setelah test case pertama diinputkan, output boleh langsung ditampilkan. Demikian seterusnya hingga test case ke lima.
9. Penilaian pada babak ini akan dinilai berdasarkan peringkat pada papan skor. Peringkat pertama akan mendapatkan poin 160 dan pada peringkat berikutnya, nilai yang didapatkan akan berkurang 2,5 untuk setiap peringkatnya.`

Contoh :

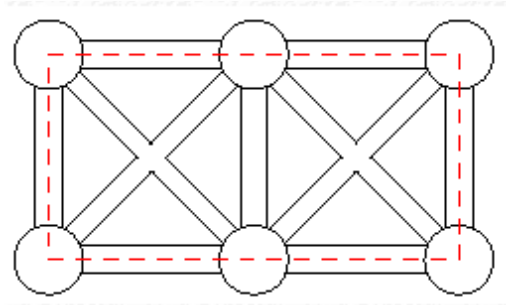
Peringkat pertama mendapat 160 poin, peringkat kedua akan mendapatkan 157,5 dan seterusnya

10. Jawaban **tidak akan diterima (ditolak)** apabila dikirim setelah pukul 15.00 WIB.

Permasalahan A

PENGANTAR KORAN YANG TERSESAT

Seorang pengantar Koran sedang bingung bagaimana cara mengantarkan korannya dengan efisien. Pengantar Koran ini tinggal di sebuah kota dengan jalan dalam kortanya seperti matriks berikut.



Gambar di atas merupakan gambaran kota dengan matriks 2×3 . Jarak antar kota secara horisontal maupun secara vertikal adalah 1 satuan panjang. Tugas anda adalah menghitung jarak terpendek yang harus ditempuh oleh sang pengantar Koran untuk mengunjungi semua kota 1 kali saja dan kembali ke kota awal dia berangkat.

Input:

Input akan dimulai dengan sebuah integer x , di mana x adalah jumlah case yang terjadi. Untuk setiap x , terdapat 2 integer, $1 < n < 100$ dan $1 < m < 100$, di mana n adalah baris dan m adalah kolom.

Output:

Untuk setiap case, cetak jarak yang terpendek yang harus ditempuh dengan ketelitian 2 desimal.

Contoh input:

```
3
2 3
2 2
3 2
```

Contoh output:

```
6.00
4.00
6.00
```

Permasalahan B

SUBSTRING

Substring adalah sebuah string yang merupakan bagian dari string lainnya. Sebagai contoh, 'aku' merupakan substring dari 'daku'. Buatlah program untuk mengenali substring dari sebuah kata.

Input:

Baris pertama dalam input adalah integer t, yang merupakan banyaknya test case. nilai t tidak lebih dari 100. Untuk t baris selanjutnya, diinputkan 2 buah string s1 dan s2, yang dipisahkan oleh sebuah spasi. Masing-masing string hanya berisi huruf 'a'-'z', dan tidak ada karakter lain. Panjang masing-masing string tidak lebih dari 50, dan panjang s2 pasti kurang dari atau sama dengan panjang s1.

Output:

Tentukan apakah string s2 merupakan substring dari s1, dengan ketentuan prioritas sebagai berikut:

- Outputkan 'sama' jika s1 sama dengan s2.
- Outputkan 'pencerminan' jika s2 merupakan kebalikan dari s1.
- Outputkan 'substring' jika s2 merupakan substring dari s1 secara urut
- Outputkan 'substring pencerminan' jika s2 merupakan substring dari pencerminan s1 secara urut
- Outputkan 'bukan substring' jika s2 tidak memenuhi semua persyaratan di atas.

Jika ada 2 syarat yang terpenuhi, maka diambil syarat dengan prioritas lebih tinggi.

Contoh input:

```
5
substring bring
substring bus
substring bit
abc abc
abc cba
```

Contoh output:

```
substring
substring pencerminan
bukan substring
sama
pencerminan
```

Permasalahan C

MINESWEEPER

Tentunya Anda sudah mengetahui permainan minesweeper.

Input:

Baris pertama setiap test case akan berisi 2 buah integer n dan m ($0 < n, m \leq 50$) yang merupakan ukuran baris dan kolom dari field minesweeper. N baris berikutnya akan berisi konfigurasi '.' dan '*' yang menunjukkan sel kosong dan sel yang berisi bom untuk setiap barisnya. Input akan diakhiri dengan n = m = 0.

Output:

Untuk setiap testcase, cetak field yang berisi nilai-nilai bom yang ada di sekitar setiap sel. Cetak sebuah baris kosong untuk mengakhiri output setiap test case.

Contoh input:

```
5 3
...
.*.
.**
*.*
***
4 7
..*...*.
...*...
.**...*.
.....*
0 0
```

Contoh output:

```
111
2*3
3**
*7*
***

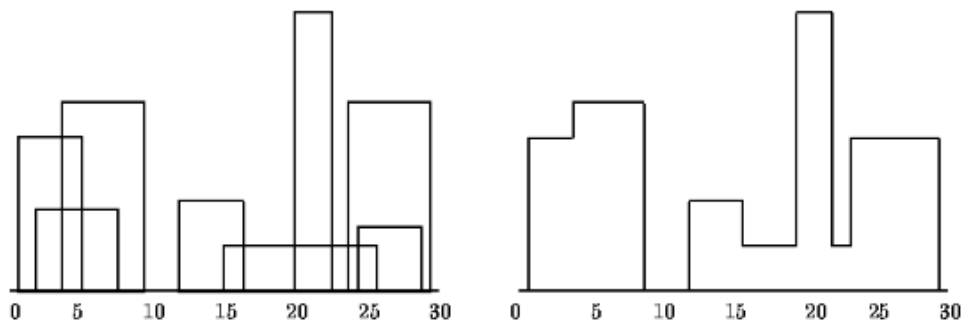
01*21*1
134*121
1**222*
12211*1
```

Permasalahan D

PEMETAAN BANGUNAN

Anda diminta untuk merancang sebuah program yang digunakan untuk memetakan ketinggian bangunan-bangunan yang ada pada sebuah kota. Sebagai asumsi nya, bangunan yang didirikan selalu berbentuk persegi atau persegi panjang dan kota itu dibangun pada lahan yang sangat datar. Peta ketinggiannya hanya 2 dimensi dan dilihat dari samping. Sebuah bangunan dinyatakan dalam (L_i, H_i, R_i) di mana L_i dan R_i adalah koordinat kiri dan kanan bangunan. Sedangkan H_i , adalah ketinggian bangunan tersebut.

Pada gambar di bawah ini, ditunjukkan peta bangunan-bangunan yang memiliki data sebagai berikut : $(1,11,5)$, $(2,6,7)$, $(3,13,9)$, $(12,7,16)$, $(14,3,25)$, $(19,18,22)$, $(23,13,29)$, $(24,4,28)$. Pada gambar kanan, adalah hasil pemetaan gabungan dari data-data bangunan di atas $(1, 11, 3, 13, 9, 0, 12, 7, 16, 3, 19, 18, 22, 3, 23, 13, 29, 0)$



Input:

Baris pertama input adalah n , $5 < n < 5.000$, di mana n adalah banyaknya bangunan. N baris selanjutnya adalah pasangan 3 data untuk setiap bangunan (L_i, H_i, R_i) . $0 \leq L_i < R_i \leq 10000$, dan $0 < H_i \leq 1000$. Bangunan dengan koordinat kiri L_i yang lebih kecil akan di inputkan terlebih dahulu.

Output:

Outputkan data gabungan dari data-data yang sudah diinputkan dalam bentuk seperti yang sudah dijelaskan di atas $(v_1, v_2, v_3, v_4, \dots, v_m)$. Untuk setiap v_i mendeskripsikan nilai-nilai koordinat bangunan dan tinggi bangunan secara bergantian. Hasil data gabungan ini diurutkan dari nilai koordinat kiri yang terkecil, dan diakhiri dengan 0. Setiap koordinat harus dipisahkan dengan 1 spasi.

Contoh Input dan Output:

INPUT	OUTPUT
8	1 11 3 13 9 0 12 7 16 3 19 18 22 3 23 13 29 0
1 11 5	
2 6 7	
3 13 9	
12 7 16	
14 3 25	
19 18 22	
23 13 29	
24 4 28	

Permasalahan E

WAKTU FAVORIT

Anda baru saja berhasil membuat sebuah warnet. Sekarang anda sedang ingin menyelidiki waktu favorit para pengunjung. Jika ingin menghitungnya secara manual, tentu saja merepotkan, maka anda memutuskan untuk membuat sebuah program penghitung waktu favorit.

Input:

Input diawali dengan sebuah integer T ($1 \leq T \leq 100$) yang mendefinisikan jumlah test case. Setiap test case diawali dengan sebuah baris yang berisi integer N ($1 \leq N \leq 100$) yang mendefinisikan banyaknya pengunjung warnet anda. N baris berikutnya akan berisi pasangan jam dan menit awal dan akhir pemakaian komputer dalam format $HH:MM$ $HH_2:MM_2$ di mana $00 \leq HH$, $HH_2 \leq 23$ dan $00 \leq MM$, $MM_2 \leq 59$. HH , HH_2 , MM dan MM_2 akan selalu 2 digit.

Output:

Tampilkan selang waktu yang merupakan waktu favorit pengunjung dalam 1 baris untuk setiap test case dalam format $HH:MM$ $HH_2:MM_2$, di mana $HH:MM$ adalah jam dan menit awal waktu favorit, dan $HH_2:MM_2$ adalah jam dan menit akhir waktu favorit.

Contoh Input dan Output:

INPUT	OUTPUT
2	09:00 10:25
3	19:15 19:45
00:00 10:25	
13:07 15:09	
09:00 14:00	
3	
19:00 19:30	
19:31 20:00	
19:15 19:45	

Permasalahan F

MESIN HITUNG

Pada umumnya, kalkulator dioperasikan untuk operasi matematika pada basis 10. Karena bosan dengan kebiasaan itu, Andi ingin membuat kalkulator yang bisa melakukan penjumlahan 2 bilangan integer basis 9. Andi mengalami kebingungan, jadi bantulah dia untuk membuat kalkulator itu.

Input:

Setiap baris input terdiri dari 2 bilangan integer, a dan b yang memenuhi $0 < a, b < 10000000000$. Kedua bilangan tersebut merupakan integer yang berbasis 9. Input akan diakhiri dengan sebuah baris dengan nilai $a=b=0$. Anda tidak perlu mencetak hasil baris terakhir ini.

Output:

Untuk setiap baris input, cetak dalam 1 baris hasil penjumlahan dalam basis 9.

Contoh input:

```
148 765
111 888
8734 8345
0 0
```

Contoh output:

```
1024
1110
18180
```

Permasalahan G

BATU GUNTING KERTAS

Batu, gunting, kertas adalah sebuah permainan 2 orang. Masing-masing pemain, mengeluarkan tanda dari salah satu dari batu, gunting, atau kertas. Tindakan tersebut dilakukan secara berturut-turut sebanyak n kali. Pemenang tiap putaran ditentukan dengan aturan sebagai berikut:

- Batu selalu mengalahkan gunting.
- Gunting selalu mengalahkan kertas.
- Kertas selalu mengalahkan batu.

Input:

Input diawali dengan sebuah integer t ($0 < t < 1000$) yang menandakan jumlah test case dari input. Untuk setiap test case, diawali dengan integer n ($0 < n < 100$) yang merupakan jumlah putaran permainan ini. n baris selanjutnya adalah baris yang menandakan apa yang dikeluarkan oleh pemain pertama dan pemain kedua. Baris ini ditandai dengan huruf B, G, atau K yang diikuti spasi dan huruf B, G, atau K lagi. B mewakili batu, G mewakili gunting, dan K mewakili kertas.

Output:

Untuk setiap test case, output-kan dalam 1 baris, pemenang dari permainan itu. Apabila terjadi seri (draw), maka output-kan SERI.

Contoh Input:

```
3
2
B K
G B
3
K K
B G
G B
1
K B
```

Contoh Output:

```
Pemain 2
SERI
Pemain 1
```

Permasalahan H

BERAPA LUAS TERBESAR?

Diketahui sebuah area seluas 10 x 10 satuan panjang. Dari area tersebut, sebagian sudah dibangun rumah yang besarnya beraneka ragam. Lilo ingin membangun rumah seluas mungkin. Bantulah Lilo untuk menghitung, berapa luas rumah maksimal yang dapat ia bangun.

Input:

Baris pertama input adalah integer t , $0 < t < 20$, yang menunjukkan banyaknya test case. Untuk setiap test case terdiri dari 10 baris. Dan masing masing baris terdiri dari 10 karakter. Sepuluh baris tersebut merupakan area yang akan dicari luas maksimalnya. Karakter yang digunakan hanya '.' (area kosong) dan '#' (area yang sudah dibangun rumah).

Output:

Carilah luas terbesar area kosong dengan ketentuan area tersebut harus saling terhubung 1 sama lain. Kemudian cetak hasilnya dalam 1 baris dengan format : "Skenario #i: L", dengan i adalah nomer test case, dan L adalah luas area kosong terbesarnya. Cetak 1 baris kosong untuk mengakhiri output.

Contoh input:

```
2
...x..x..
..x.x...x
...x...xx.
...x.x...
..x.xxx...
..x.x..x..
xx..x..x.x
...x...x.
..x.x..x..
...x.x...

...x...x..
...x..x.x
..xx...x.
.x..x...x.
x.x..x...
x.x...x...
..x..x.x..
..xx..xx.x.
x..xxxx..x
...xxx...
```

Contoh output:

Skenario #1: 34

Skenario #2: 24

Permasalahan I

MENANG KALAH

Bejo, joko, Tukiman, Supratman senang bermain kartu “Aneh”. “Aneh” adalah sebuah permainan dari 2 tim dengan masing-masing tim anggotanya 2 orang. Setiap 1 game berakhir, pasangan dari tiap tim akan diacak lagi. Setiap pemain akan mencatat hasil menang kalah mereka. Sayangnya, Bejo kehilangan kertas perhitungannya. Tetapi dia bias melihat kertas perhitungan ketiga temannya. Buatlah sebuah program untuk membantu Bejo menghitung jumlah menang kalahnya.

Input:

Setiap test case dari input akan terdiri dari 6 integer. Dua integer pertama adalah jumlah menang kalah Joko, 2 integer selanjutnya adalah jumlah menang kalah Tukiman, dan 2 integer yang terakhir adalah jumlah menang kalah Supratman. Enam angka nol akan mengakhiri input.

Output:

Untuk setiap test case, tampilkan jumlah menang kalah Bejo sesuai dengan format di bawah.

Contoh input:

```
10 3 6 7 8 5
1874 2945 2030 2789 1025 3794
0 0 0 0 0 0
```

Contoh output:

```
Jumlah menang kalah Bejo adalah 2-11.
Jumlah menang kalah Bejo adalah 4709-110.
```

Permasalahan J

SPIRAL TAPI LURUS

Robo ingin menyandikan pesan yang ingin dikirimnya kepada Robi agar tidak diketahui oleh pihak lainnya. Cara penyandian yang digunakannya adalah dengan membuat sebuah matriks dan mengisinya dengan pesan yang ingin dikirim mulai dari pusat matriks, secara spiral. Penyusunan dimulai dari 1 karakter pertama pada pusat matriks, lalu 1 karakter berikutnya ke kanan, lalu 1 karakter berikutnya ke bawah, lalu 2 karakter berikutnya ke kiri, lalu 2 karakter berikutnya ke atas, lalu 3 karakter berikutnya ke kanan, 3 karakter berikutnya ke bawah, dan seterusnya hingga semua karakter dalam kalimat telah selesai dimasukkan ke dalam matriks. Dalam proses penyandian, karakter spasi diganti dengan '_' (underscore), dan jika ada baris atau kolom tersisa setelah karakter terakhir, maka baris /kolom itu diisi dengan karakter '_'(underscore). Lalu, hasil penyandian adalah berupa 1 baris kalimat yang dibaca secara horizontal dari kiri ke kanan tiap baris pada matriks.

Misalkan kalimat yang akan disandikan adalah "Informatic Logical Programming Competition 2010 bertema pemanfaatan teknologi informasi untuk meningkatkan daya saing bangsa." Maka susunan matriksnya adalah sebagai berikut:

```
a _ s a i n g _ b a n g
y o l o g i _ i n f o s
a n _ 2 0 1 0 _ b e r a
d k n r o g r a m r m .
_ e o P a t i c m t a _
n t i _ m I n _ i e s _
a _ t l r o f L n m i _
k n i a c i g o g a _ _
t a t e p m o C _ _ u _
a t a a f n a m e p n _
k g n i n e m _ k u t _
```

Sedangkan hasil penyandian dari kalimat di atas adalah :

```
a_saing_bangyologi_infosan_2010_beradknrogramm._eoPaticmta_nti_mIn_
ies_a_tlrofLnmi_kniacigoga__tatepmoC__u_ataafnamepn_kgninem_kut_
```

Input:

Input berupa kalimat dengan panjang maksimal 250 karakter.

Output:

Cetak hasil penyandian yang sudah dilakukan.

Contoh 1:

- **Input: (dalam 1 baris)**
Informatic Logical Programming Competition 2010 bertema pemanfaatan teknologi informasi untuk meningkatkan daya saing bangsa.
- **Output: (dalam 1 baris)**
a_saing_bangyologi_infosan_2010_beradknrogramr._eoPaticmta_nt
i_mIn_ies_a_tlrofLnmi_kniacigoga__tatepmoC__u_ataafnamepn_kgni
nem_kut_

Contoh 2:

- **Input:**
Aku pasti juara!
- **Output:**
sti_aAkjp_uu!ara

Contoh 3:

- **Input:**
Teknik Informatika
- **Output:**
__Inf_kTeoainkrkitam