

Hari 1 / Soal 1: Bukit dan Lembah

Kode Soal: bukit
Batas Run-time: 1 detik / test-case
Batas Memori: 1 MB
Masukan: Standard input
Keluaran: Standard output

Deskripsi

Diberikan data ketinggian yang di catat dalam perjalanan dari suatu posisi awal ke posisi akhir. Data ketinggian adalah bilangan-bilangan integer (bulat) positif. Jalan kadang menaik, kadang menurun, kadang datar saja. Posisi dimana terjadi perubahan menaik kemudian menurun (boleh diselingi jalan datar) didefinisikan sebagai puncak dari suatu bukit. Sebaliknya, posisi terjadi perubahan dari menurun terus menaik (boleh diselingi bagian jalan yang datar) didefinisikan sebagai titik terbawah suatu lembah. Walaupun perubahan tersebut kecil saja, definisi itu tetap berlaku. Carilah beda ketinggian terbesar antara puncak bukit dengan titik terbawah lembah berikutnya atau sebaliknya antara titik terbawah lembah dengan puncak bukit berikutnya pada data perjalanan tersebut.

Masukan

Masukan berisi data yang bisa sangat banyak sekali. Setiap elemen data dalam baris tersendiri. Anda membacanya dari yang pertama hingga end of file; minimal ada dua data dalam masukan..

Keluaran

Program hanya menghasilkan satu bilangan yang menyatakan beda ketinggian terbesar yang diperoleh. Perbedaan tinggi paling besar dijamin tidak akan melebihi harga long integer dalam Pascal.

Contoh

Masukan

```
10
26
26
35
35
27
30
30
45
10
8
9
```

Keluaran

```
37
```

Penjelasan

Ada 12 data. Beda ketinggian pertama (10 ke 35) adalah 25, beda kedua (35 ke 27) adalah 8, beda ketiga (27 ke 45) adalah 18, beda ketinggian keempat (45 ke 8) adalah 37, dan beda ketinggian kelima (8-9) adalah 1. Jadi beda ketinggian tertinggi adalah 37.

Petunjuk Standard I/O

Banyak peserta yang tidak mengikuti PJJ karena berbagai alasan. Dalam deskripsi soal-soal PJJ diberikan contoh-contoh bagaimana membaca dari standard input/output. Tapi, ok sekarang diberikan contoh membaca semua bilangan dari standard input dengan format seperti yang digunakan pada sal Bukit-Lembah.

```
var bil: integer;
begin
  while not eof(input) do begin
    readln(bil);
    write(bil, ' ');
  end;
end.
```

Misalnya, disimpan dengan nama coba.pas. Untuk mengujinya, buatlah file coba.txt berisi bilangan-bilangan dengan format seperti untuk soal Bukit-Lembah sbb.

```
34
2
44
2
4
5
245
3
32
```

Setelah dicompile, jalankan dengan perintah

```
coba < coba.txt
```

Soal-soal lainnya pun menggunakan standard input/output sehingga anda tidak perlu assign(...), dst.

Hari 1 / Soal 2: Kata Spiral

Kode Soal: spiral
Batas Run-time: 1 detik / test-case
Batas Memori: 1 MB
Masukan: Standard input
Keluaran: Standard output

Deskripsi

Suatu sistem sandi menyandikan kalimat yang diberikan dalam bentuk spiral. Penyusunan tersebut dilakukan membentuk matriks spiral yang dimulai pusat matriks 1 karakter pertama, lalu 1 karakter berikutnya ke kanan, lalu 1 karakter berikutnya ke bawah, lalu 2 karakter berikutnya ke kiri, lalu 2 karakter berikutnya ke atas, 3 karakter berikutnya ke kanan, 3 karakter berikutnya ke bawah, dan seterusnya hingga semua karakter dalam kalimat termasuk dalam spiral. Khususnya, karakter spasi di ganti dengan “_” (underscore), dan jika ada baris/kolom tersisa setelah karakter terakhir maka elemen-elemen matriks diisi juga dengan “_” (underscore) tsb. Misalnya kalimat “Seluruh peserta OSN bidang komputer harus mengerjakan soal-soal sebaik-baiknya untuk mendapatkan peringkat terbaik.” Dikodekan kedalam matriks sebagai berikut.

b	a	i	k	.	_	_	_	_	_	_
r	a	i	k	n	y	a	_	u	n	t
e	b	m	e	n	g	e	r	j	a	u
t	-	_	b	i	d	a	n	g	k	k
_	k	s	_	h	_	p	e	_	a	_
t	i	u	N	u	S	e	s	k	n	m
a	a	r	S	r	u	l	e	o	_	e
k	b	a	O	_	a	t	r	m	s	n
g	e	h	_	r	e	t	u	p	o	d
n	s	_	l	a	o	s	-	l	a	a
i	r	e	p	_	n	a	k	t	a	p

Masukan

Program membaca satu baris teks paling panjang 250 karakter.

Keluaran

Program harus menghasilkan sejumlah baris sesuai dengan matriks yang dibentuk. Setiap baris keluaran berisikan karakter-karakter dari baris yang sama berturut-turut dari kolom paling kiri ke paling kanan tanpa pemisahan (karakter-karakter dituliskan bersambungan menjadi satu string serta jangan lupa setiap spasi menjadi underscore).

Contoh 1

Masukan (tertulis dalam satu baris)

Seluruh peserta OSN bidang komputer harus mengerjakan soal-soal sebaik-baiknya untuk mendapatkan peringkat terbaik.

Keluaran

```
baik._____  
raiknya_unt  
ebmengerjau  
t-_bidangkk  
_ks_h_pe_a_  
tiuNuSesknm  
aarSruleo_e  
kbaO_atrmsn  
geh_retupod  
ns_laos-laa  
irep_naktap
```

Contoh 2

Masukan (tertulis dalam satu baris)

```
TOKI
```

Keluaran (ada ralat pada ralat ini)

```
TO  
IK
```

Contoh 3

Masukan (tertulis dalam satu baris)

```
OSN
```

Keluaran

```
OS  
_N
```

Contoh 4

Masukan (tertulis dalam satu baris)

```
Bisakah Kamu?????
```

Keluaran

_h_Ka
_aBim
_kasu
?????

Hari 1 / Soal 3: Faktorial

Kode Soal: faktorial
Batas Run-time: 1 detik / test-case
Batas Memori: 1 MB
Masukan: Standard input
Keluaran: Standard output

Deskripsi

Pasti anda sudah pernah belajar apakah itu bilangan faktorial. Sesuai dengan definisinya bilangan faktorial $n! = n.(n-1).(n-2).....1$. Untuk n yang kecil bilangan tersebut masih mudah dihitung dengan manual tetapi untuk n yang cukup besar bisa menghasilkan bilangan dengan jumlah digit yang amat banyak sehingga tentu melelahkan jika dihitung dengan manual bahkan dengan komputerpun kita perlu cara khusus untuk menanganinya akibat adanya batasan representasi bilangan integer. Oleh sebab itu dalam rumus-rumus bilangan faktorial tetap dinyatakan dalam bentuk $n!$, misalnya $n!/(n-k)!k!$. Untuk perhitungan $(50! * 100! * 75!)/(73! * 99! * 52!)$ tentu tidak memungkinkan kalau masing-masing bilangan faktorialnya dihitung terlebih dahulu lalu diperkalikan/diperbagikan kemudian. Sebaliknya, akan lebih mudah jika dilakukan penyederhanaan dengan menghilangkan faktor-faktor yang sama antara pembilang (di atas tanda bagi) dan penyebut (di bawah tanda bagi) sampai tinggal sejumlah bilangan nonfaktorial yang tersisa. Contohnya.

$$(50! * 100! * 75!)/(73! * 99! * 52!) \\ = (74 * 75 * 100) / (51 * 52)$$

yang berikutnya, dengan menguraikan ke faktor-faktor bilangan prima dapat disederhanakan lebih lanjut menjadi (catatan: notasi $^$ adalah tanda pangkat).

$$= (2 * 37 * 3 * 5^2 * 2^2 * 5^2) / (3 * 17 * 2^2 * 13) \\ = (2 * 5^4 * 37) / (13 * 17)$$

Dengan ekspresi tersebut, perhitungan kemudian menjadi lebih memungkinkan untuk dilakukan dibandingkan saat masih dalam bentuk faktorial.

Buatlah program yang dapat melakukan penyederhanaan ekspresi perkalian/pembagian bilangan-bilangan faktorial hingga faktor-faktor bilangan prima seperti di atas.

Masukan

Masukan terdiri atas dua baris. Baris pertama menyatakan harga-harga n dari bilangan-bilangan faktorial yang berada pada bagian pembilang (yang dibagi). Baris kedua menyatakan harga-harga n dari bilangan-bilangan faktorial yang berada pada bagian penyebut. Format masing-masing bilangan sama sbb. Bilangan pertama menyatakan jumlah bilangan faktorial pada baris ybs. Jumlah bilangan tersebut paling sedikit 1 dan paling banyak 20. Misalkan jumlah bilangan itu m , berikutnya ada m bilangan bulat positif a, b, c, \dots , dst. yang masing-masing dapat berharga 2 sampai dengan 1000. Masing-masing bilangan tersebut menyatakan bilangan-bilangan faktorial a, b, c, \dots dst.

Keluaran

Keluaranya terdiri dari dua baris, baris pertama untuk pembilang dan baris kedua untuk penyebut. Bilangan-bilangan adalah bilangan prima yang tersisa dan untuk pangkat yang lebih dari 1 dituliskan di dalam tanda kurung langsung setelah bilangan prima ybs (tanpa spasi). Bilangan-bilangan prima dituliskan dari yang terkecil ke yang terbesar. Sekali lagi, penulisan pangkat hanya untuk bilangan prima dengan pangkat lebih dari 1. Jika salah satu dari pembilang atau penyebut berharga 1 maka baris untuk ybs dikosongkan (sebagai konsekuensi tidak adanya faktor bilangan prima padanya).

Contoh 1

Masukan

3 50 100 75
3 73 99 52

Keluaran

2 5(4) 37
13 17

Contoh 2

Masukan

1 100
1 99

Keluaran

2(2) 5(2)

Contoh 3

Masukan

1 48
1 49

Keluaran

7(2)





















Penjelasan: pada contoh kedua, baris kedua kosong dan pada contoh ketiga, baris pertama yang kosong.

Hari 2 / Soal 1: Sandi Ayam

Kode Soal: ayam
Batas Run-time: 0,5 detik / test-case
Batas Memori: 16 MB
Masukan: Standard input
Keluaran: Standard output

Pak Dengklek punya tetangga baru yang berprofesi sebagai peternak ayam. Tentunya, tetangga baru ini memiliki beberapa ayam. Bebek-bebek Pak Dengklek tidak suka dengan kehadiran ayam-ayam ini, antara lain karena mereka curiga ayam-ayam tersebut akan melakukan hal-hal yang tidak baik. Dalam beberapa hari semenjak kedatangannya saja, bebek-bebek sudah menemukan goresan-goresan aneh di tanah. Setelah diteliti, para bebek menyimpulkan bahwa ayam-ayam tersebut sedang menuliskan sandi angka. Penelitian lebih lanjut memberikan hasil mengenai arti dari setiap sandi, seperti yang dijelaskan di bawah ini.

Ayam-ayam menggunakan 20 macam simbol. Setiap simbol adalah hasil dari patokan paruh ayam (bulatan kecil) dan/atau goresan cakar ayam (garis). Bukti-bukti yang sejauh ini ada menunjukkan bahwa setiap simbol memiliki padanan seperti yang tertera pada gambar 1.

0	1	2	3	4
				
5	6	7	8	9
				
10	11	12	13	14
				
15	16	17	18	19
				

Gambar 1. Terjemahan Sandi Angka Ayam

Untuk membentuk angka yang bernilai lebih dari 19, simbol-simbol di atas ditulis secara vertikal dan dibaca seperti layaknya bilangan dengan basis 20. Simbol dengan bobot lebih besar digores paling atas. Tentunya, dengan sistem ini para bebek berharap bahwa nilai satuan dari setiap simbol, mulai dari simbol yang mewakili satuan terkecil, adalah 1, 20, 400, 8000, dst. Akan tetapi, ayam-ayam tersebut lebih cerdas. Ternyata, harga satuan simbol ketiga dari bawah (bila ada) hanya 18 kali lebih

besar dari harga satuan simbol kedua dari bawah. Akan tetapi untuk simbol-simbol berikutnya, nilai satuannya tetap 20 kali lebih besar daripada satuan sebelumnya. Untuk lebih jelasnya, lihat contoh berikut ini.



Gambar 2. Contoh Sandi Ayam

Seperti yang tertera pada gambar di atas, sandi di atas terdiri dari 4 simbol. Simbol paling atas melambangkan angka 2, simbol di bawahnya melambangkan angka 0, berikutnya angka 17 dan yang paling bawah melambangkan angka 1. Bila kita mengikuti sistem bilangan yang digunakan oleh para bebek (yang untungnya adalah sama seperti kita, desimal), maka angka yang dimaksud oleh sandi di atas adalah $14741 (= 2 \times 7200 + 0 \times 360 + 17 \times 20 + 1)$.

Para bebek memiliki kesulitan untuk menerjemahkan sandi-sandi yang panjang. Untuk itu, mereka meminta bantuan kalian untuk menerjemahkannya.

FORMAT MASUKAN

Masukan akan berisi sebuah angka yang tertulis dalam sandi ayam. Untuk mempermudah tugas kalian, patokan ayam akan dimasukkan sebagai titik (.), dan cakaran ayam akan dimasukkan sebagai tanda hubung (-). Simbol 0 (lingkaran besar) akan dimasukkan sebagai angka 0. Setiap dua simbol akan dipisahkan oleh sebuah baris kosong. Masukan akan diakhiri oleh penanda akhir berkas (*end-of-file*).

Masukan berisi setidaknya sebuah simbol, tapi tidak lebih dari 30 simbol. Untuk setidaknya setengah dari total bobot testcase yang diujikan, masukan terdiri tidak lebih dari 14 simbol.

CONTOH MASUKAN

```
..  
0  
..  
-  
-  
-  
.  
.
```

FORMAT KELUARAN

Keluaran hanya terdiri dari sebuah baris berisi sebuah bilangan bulat yang merupakan hasil penerjemahan sandi ayam pada masukan.

CONTOH KELUARAN

```
14741
```

Peringatan

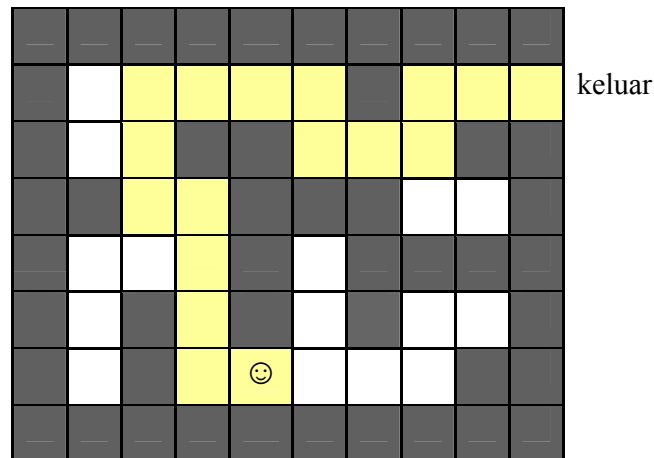
Keluaran untuk masukan dengan banyak simbol lebih dari 14 buah *tidak* dijamin dapat dimuat oleh sebuah variabel integer 64-bit bertanda (int64 di FreePascal).

Hari 2 / Soal 2: Maze

Kode Soal: maze
Batas Run-time: 1 detik / test-case
Batas Memori: 1 MB
Masukan: Standard input
Keluaran: Standard output

Deskripsi

Anda tahu permainan maze? Kalau melihat gambar berikut ini pasti tahu. Dalam permainan maze ini makhluk yang digambarkan dengan bulatan wajah Mr Groovy harus mencari jalan ke luar dari grid maze yang diberikan.



Jalan keluar yang harus dilaluinya adalah kota-kotak yang berwarna kuning tsb. Masalahnya karena bisa terdapat beberapa cara untuk mencapai jalan keluar maka disini anda harus menemukan jumlah kotak yang paling sedikit dalam lintasan (menyatakan juga jumlah langkah terpendek untuk mencapai bagian luar). Dalam hal contoh maze di atas yang paling sedikit adalah 17 kotak/langkah yaitu yang berwarna kuning tsb.

Masukan

Baris pertama berisikan b dan k yang menyatakan jumlah baris dan jumlah kolom matriks grid tersebut. Kedua bilangan dipisahkan satu spasi. Jumlah baris/kolom terkecil adalah 3 dan jumlah baris/kolom terbesar adalah 100. Pada b baris berikutnya maze didefinisikan sbb. Harga -1 menyatakan dinding yang tidak dapat ditembus, harga 0 menyatakan ruang yang dapat dilalui. Dipastikan sekurangnya ada satu jalan keluar. Setiap baris grid pada baris masukan tersendiri. Pada baris yang sama harga-harga tersebut dituliskan terpisah satu spasi dengan harga berikutnya. Pada baris terakhir terdapat dua bilangan a dan b yang menyakan posisi awal Mr Groovy berada, a nomor baris dan b nomor kolom dengan penomoran baris: 1, 2, 3, ..., dimulai dari atas ke bawah dan penomoran kolom: 1, 2, 3, ..., dimulai dari kiri ke kanan.

Dipastikan posisi awal ini selalu berada pada ruangan (bukan tembok!) dan di dalam matriks.

Keluaran

Program hanya mengeuarkan jumlah langkah paling sedikit untuk mencapai luar (sama juga dengan jumlah kotak paling sedikit yang dilalui).

Contoh

Masukan

```
8 10
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
-1 0 0 0 0 0 -1 0 0 0
-1 0 0 -1 -1 0 0 0 -1 -1
-1 -1 0 0 -1 -1 -1 0 0 -1
-1 0 0 0 -1 0 -1 -1 -1 -1
-1 0 -1 0 -1 0 -1 0 0 -1
-1 0 -1 0 0 0 0 0 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1
7 5
```

Keluaran

```
17
```

Hari 2 / Soal 3: Segitiga Bebek

Kode Soal: bebek3
Batas Run-time: 0,5 detik / test-case
Batas Memori: 16 MB
Masukan: Standard input
Keluaran: Standard output

Suatu hari Pak Dengklek mendapat penghasilan lebih dari hasil berjualan telur bebek asin. Oleh karena itu ia ingin membagi keuntungannya dengan bebek-bebeknya, dengan cara memberi makan lebih untuk mereka. Sayangnya uangnya hanya cukup untuk memberi makan lebih kepada tiga ekor bebek. Untuk itu ia membuat suatu cara untuk memilih tiga ekor bebek mana yang akan diberinya makan lebih hari itu.

Saat itu, bebek-bebeknya sedang berkeliaran di ladangnya yang luas. Pak Dengklek memutuskan untuk memberi makan lebih kepada tiga ekor bebek yang di posisinya sekarang membentuk segitiga dengan luas minimum. Yang dimaksud segitiga dengan luas minimum di sini adalah, semua segitiga lain yang dibentuk oleh bebek-bebek memiliki luas yang lebih besar dari luas segitiga tersebut.

Karena banyaknya jumlah bebek-bebek, Pak Dengklek meminta bantuanmu untuk masalah ini. Bantulah Pak Dengklek dengan mencari luas segitiga minimum tersebut akurat hingga dua tempat desimal.

FORMAT MASUKAN

Baris pertama dari masukan berisi sebuah bilangan N , yang menyatakan jumlah bebek-bebek yang tersebar di ladang Pak Dengklek saat ini. $1 \leq N \leq 300$. Baris kedua hingga baris ke $N+1$ masing masing akan berisi dua buah bilangan bulat x_i dan y_i , yang merupakan koordinat bebek ke- i di sistem koordinat kartesian. $-10000 \leq x_i, y_i \leq 10000$. Tidak akan ada dua bebek yang berbeda yang berada di posisi (x, y) yang sama.

CONTOH MASUKAN

```
4
0 0
-5 10
3 0
0 9
```

FORMAT KELUARAN

Keluaran hanya terdiri dari sebuah baris berisi bilangan yang merupakan luas segitiga minimum menurut syarat yang sudah dijelaskan dengan dua tempat desimal. Jika tidak ada segitiga yang memenuhi syarat tersebut, keluarkan angka -1.00 .

CONTOH KELUARAN

13.50
