**A. Membuat Kerajaan**

|  |  |
| --- | --- |
| Time limit | 1 s |
| Memory limit | 64 MB |

**Deskripsi**

Terdapat sebuah dunia yang berbentuk segi empat dengan ukuran R\*C. Dunia tersebut terbagi menjadi petak-petak berukuran 1\*1, sehingga terdapat R\*C petak. Jika suatu petak para baris i dan kolom j dikuasai, maka yang menguasai akan mendapatkan keuntungan sebesar Aij. Terkadang, menguasai sebuah petak justru mendatangkan kerugian, hal ini ditandai Aij yang bernilai negatif. Seorang penghuni dunia tersebut ingin membuat satu kerajaan yang berbentuk persegi panjang, serta mendapatkan keuntungan sebesar-besarnya. Tentukan keuntungan yang bisa ia dapat!

**Format Masukan**

Baris pertama berisi dua buah bilangan bulat R dan C, banyak baris dan kolom pada dunia.

R baris selanjutnya berisi C buah bilangan bulat Aij, keuntungan yang didapat dari suatu petak.

**Format Keluaran**

Satu baris berisi sebuah bilangan bulat, keuntungan maksimum yang bisa didapat.

**Contoh Masukan**

3 3

3 -2 4

3 -2 4

4 -2 2

**Contoh Keluaran**

14

**Batasan**

* 1 ≤ R, C ≤ 100
* -100.000 ≤ Aij ≤ 100.000

**B. Travelling Salesman Problem**

|  |  |
| --- | --- |
| Time limit | 2 s |
| Memory limit | 64 MB |

**Deskripsi**

Diberikan N buah kota yang dinomori dari 1 sampai N. Dari setiap kemungkinan pasangan kota, terdapat tepat dua buah jalan dua arah dengan panjang tertentu yang menghubungkan kedua kota tersebut. Kecuali dari suatu kota ke dirinya sendiri, hanya terdapat tepat satu jalan.

Pada awalnya, Anda berada di kota bernomor 1. Hitunglah jarak terpendek yang harus ditempuh sehingga Anda melewati setiap kota tepat sekali dan kembali pada kota nomor 1!

Perhatikan bahwa pada saat Anda mendatangi kota bernomor 1 untuk terakhir kalinya, Anda tidak dihitung melewati kota tersebut.

**Format Masukan**

Baris pertama berisi sebuah bilangan N, banyaknya kota.

Selanjutnya diberikan sebuah matriks berukuran NxN yang mana setiap elemen pada setiap baris dipisahkan oleh spasi.

Bilangan pada baris ke-i dan kolom ke-j adalah Wij yang menandakan terdapat suatu jalan yang menghubungkan kota ke-i dengan kota ke-j dengan panjang Wij.

**Format Keluaran**

Keluarkan sebuah baris yaitu jarak terpendek yang harus ditempuh.

**Contoh Masukan 1**

4

0 1 5 4

6 0 2 6

8 7 0 3

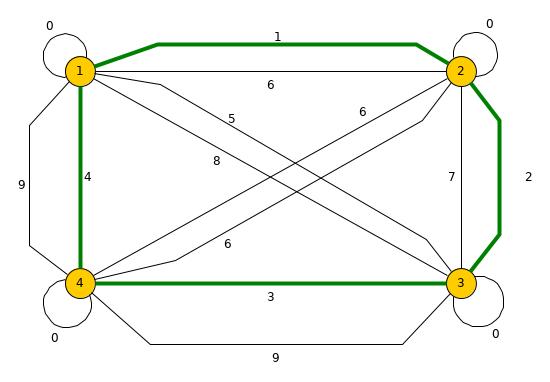
9 6 9 0

**Contoh Keluaran 1**

10

**Penjelasan Contoh**

Berikut adalah peta kota yang sesuai dengan masukan:



Jalan yang digunakan adalah jalan yang ditandai dengan garis hijau tebal.

Jarak tempuh dari rute 1 -> 2 -> 3 -> 4 -> 1 adalah 1 + 2 + 3 + 4 = 10. Tidak ada rute yang menghasilkan jarak tempuh yang lebih pendek dari 10.

**Batasan**

* 2 ≤ N ≤ 15
* 0 ≤ Wij ≤ 100.000

**C. Koleksi Matriks Pak Chanek**

|  |  |
| --- | --- |
| Time limit | 2 s |
| Memory limit | 64 MB |

**Deskripsi**

Pak Chanek sangat senang dengan matriks. Saat ini, ia telah mengoleksi N buah matriks yang dinomori dari 1 sampai N secara berturut-turut. Untuk tiap dua buah matriks dengan nomor yang berurutan, banyaknya kolom pada matriks dengan nomor yang lebih kecil akan sama dengan banyaknya baris pada matriks dengan nomor yang lebih besar.

Kali ini, ia akan mengalikan N matriks tersebut sehingga akan menjadi satu matriks saja. Urutan perkalian matriks tersebut boleh diacak. Biaya mengalikan matriks yang berukuran a x b dengan matriks yang berukuran b x c adalah a x b x c. Pak Chanek tidak tahu berapa biaya minimum yang diperlukan. Oleh karena itu, ia akan menanyakan 2 buah pertanyaan pada Anda:

1. Berapa biaya minimum yang ia perlukan untuk mengalikan N matriks tersebut?  
2. Berapa banyak cara berbeda mengalikan N matriks tersebut untuk menghasilkan biaya minimum? (setelah di modulo 1.000.000.007)

Bisakah Anda menjawab pertanyaan tersebut?

**Format Masukan**

Baris pertama berisi sebuah bilangan bulat N, banyaknya matriks Pak Chanek.  
Baris kedua berisi N+1 bilangan bulat a[0], a[1], ..., a[n]. Untuk tiap matriks i dengan 1 ≤ i ≤ N, jumlah baris matriks i dinyatakan dalam a[i-1] dan jumlah kolom matriks i dinyatakan dalam a[i].

**Format Keluaran**

Keluarkan dua buah bilangan bulat, jawaban untuk masing-masing pertanyaan secara berturut-turut.

**Contoh Masukan 1**

4

1 2 2 1 3

**Contoh Keluaran 1**

9 2

**Contoh Masukan 2**

3

1 2 3 4

**Contoh Keluaran 2**

18 1

**Penjelasan Masukan 1**

Cara untuk mengalikan matriks-matriks pada masukan agar menghasilkan biaya minimum adalah sebagai berikut:  
(((A x B) x C) x D)  
((A x (B x C)) x D)

**Batasan**

* 1 ≤ N ≤ 100
* 1 ≤ a[i] ≤ 1000, untuk tiap 0 ≤ i ≤ N

**D. Grid Escape**

|  |  |
| --- | --- |
| Time limit | 1 s |
| Memory limit | 64 MB |

**Deskripsi**

Anda baru saja terperangkap di sebuah *maze* yang berbentuk segiempat dan terdiri dari R\*C ruang. Pembuat *maze* ini orang yang cukup baik, di ruangan anda sekarang terdapat sebuah peta yang berisi banyak informasi. Saat ini, Anda berada di pojok kiri atas *maze*, atau ruang di (1,1). Menurut peta itu, pintu keluar terdapat di pojok kanan bawah *maze*, atau ruang di (R,C). Selain itu, di *maze* tersebut terdapat N ruang jebakan, yang tentunya tidak boleh anda masuki jika ingin keluar hidup-hidup. *Maze* ini juga aneh, karena Anda cuma bisa bergerak ke kanan dan ke bawah. Dengan kata lain, jika Anda berada di ruang (a,b), anda hanya bisa bergerak ke ruang (a,b+1) atau (a+1,b). Sebagai matematikawan, Anda malah penasaran, sebenarnya, ada berapa cara Anda bisa mencapai pintu keluar?

**Format Masukan**

Baris pertama berisi 3 buah bilangan bulat R, C, dan N, masing-masing menyatakan banyak baris dan banyak kolom pada *maze*, serta banyak ruang jebakan.

N baris selanjutnya berisi dua buah bilangan bulat bi dan ki, baris dan kolom ruang jebakan ke-i.

**Format Keluaran**

Satu baris berisi sebuah bilangan bulat, banyaknya cara Anda bisa keluar dari *maze*. Berhubung angka ini bisa besar sekali, keluarkan jawabannya modulo 1.000.000.007.

**Contoh Masukan**

4 4 2

2 2

3 3

**Contoh Keluaran**

4

**Batasan**

* 1 ≤ R, C, N ≤ 1.000
* 1 ≤ bi≤ R
* 1 ≤ ki ≤ C
* Tiap ruang jebakan merupakan ruang yang berbeda.
* Ruang awal dan pintu keluar bukan merupakan ruang jebakan.

**E. LIS**

|  |  |
| --- | --- |
| Time limit | 1 s |
| Memory limit | 64 MB |

**Deskripsi**

Salah satu persoalan *dynamic programming* yang cukup klasik adalah *longest increasing subsequence* atau biasa disingkat LIS. LIS dari suatu sekuens merupakan subsekuens terpanjang, sehingga elemen-elemen pada subsekuens itu terurut menaik, dan tidak ada elemen yang kembar. Sekarang, diberikan sekuens, anda diminta mencari panjang LIS-nya, serta mencari banyaknya LIS dari sekuens tersebut. Karena banyaknya LIS bisa cukup besar, cukup keluarkan hasilnya dalam modulo 1.000.000.007.

**Format Masukan**

Baris pertama berisi sebuah bilangan bulat N, banyak bilangan pada sekuens.

Baris kedua berisi N buah bilangan bulat Ai, elemen ke-i dari sekuens.

**Format Keluaran**

Satu baris berisi dua buah bilangan bulat, panjang LIS, dan banyaknya LIS dalam modulo 1.000.000.007.

**Contoh Masukan**

5

2 1 4 4 5

**Contoh Keluaran**

3 4

**Penjelasan**

Tidak ada subsekuens dengan panjang melebihi 3 yang memiliki properti semuanya terurut menaik, dan tidak ada elemen yang kembar. LIS pada sekuens yang terdapat di contoh masukan adalah:

1. 1, 4, 5 (4 pada indeks ke-3)
2. 1, 4, 5 (4 pada indeks ke-4)
3. 2, 4, 5 (4 pada indeks ke-3)
4. 2, 4, 5 (4 pada indeks ke-4)

Sehingga ada 4 LIS.

**Batasan**

* 1 ≤ N ≤ 5.000
* 1 ≤ Ai ≤ 5.000

**F. Mail Error**

|  |  |
| --- | --- |
| Time limit | 1 s |
| Memory limit | 64 MB |

**Deskripsi**

Anda baru saja menerima pesan yang katanya penting dari bos anda. Namun, entah karena berbagai gangguan, dari bit yang bergeser sampai konspirasi organisasi tertentu, pesan yang anda terima mengalami gangguan! Pesan tersebut malah muncul sebagai dua buah string, yang anda sebut S dan T. Anda lalu memiliki ide aneh, bahwa pesan yang dikirim sebenarnya adalah *longest common subsequence* dari S dan T. Sekarang, ketimbang menghubungi bos anda, sekarang anda justru sibuk mencari panjang pesan hasil ide aneh anda. Tentukanlah panjang pesan hasil ide aneh anda!

Subsekuens dari suatu string adalah hasil menghapus sebagian atau seluruh karakter dari string tersebut. Sebagai contoh, "adc" adalah subsekuens dari "abdac", namun "db" bukan.

*Longest common subsequence*dari S dan T adalah subsekuens terpanjang yang terdapat baik di S maupun di T.

**Format Masukan**

Baris pertama berisi string S dan baris kedua berisi string T, dimana S dan T adalah dua string yang diterima.

**Format Keluaran**

Satu baris berisi sebuah bilangan bulat, panjang pesan hasil ide aneh anda.

**Contoh Masukan**

aczad

azca

**Contoh Keluaran**

3

**Batasan**

* 1 ≤ |S|, |T| ≤ 1.000
* S dan T hanya akan terdiri dari karakter 'a' sampai 'z'.

**G. Maling Ayam**

|  |  |
| --- | --- |
| Time limit | 1 s |
| Memory limit | 64 MB |

**Deskripsi**

Setelah berbagai rencana jahatnya digagalkan Pak Chanek, sekarang Pak Kanech sudah tidak punya uang! Berhubung ia sangat kelaparan, ia memutuskan untuk mencuri beberapa ayam di kampungnya. Ketika memasuki salah satu kandang ayam, ia terkejut karena di dalamnya terdapat banyak sekali ayam, yang jumlahnya mungkin tidak terbatas!

Ayam-ayam di kandang tersebut dibagi ke dalam beberapa ruang di kandang tersebut. Setelah dihitung, ternyata ada N ruang di kandang tersebut. Pak Kanech menyadari bahwa ayam-ayam di ruang yang sama memiliki berat badan, serta harga yang sama. Ayam-ayam di ruang ke-i memiliki berat Wi dan harga Ci.

Sejujurnya, Pak Kanech tidak menyangka bahwa ayam di kandang tersebut jumlahnya tidak terbatas. Sayang, ia hanya membawa karung yang kuat menahan berat paling banyak M. Sekarang, Pak Kanech ingin memaksimalkan total harga ayam yang ia curi. Bantulah Pak Kanech!

**Format Masukan**

Baris pertama berisi dua buah bilangan bulat N dan M, banyak ruang dan berat maksimum yang dapat ditahan karung Pak Kanech.

N baris selanjutnya berisi Wi dan Ci, berat dan harga ayam-ayam yang berada di ruang ke-i.

**Format Keluaran**

Satu baris berisi sebuah bilangan bulat, total harga ayam maksimum yang bisa Pak Kanech curi.

**Contoh Masukan**

3 10

2 3

3 4

4 7

**Contoh Keluaran**

17

**Batasan**

* 1 ≤ N ≤ 1.000
* 1 ≤ M ≤ 5.000
* 1 ≤ Wi ≤ 5.000
* 1 ≤ Ci ≤ 5.000

**Catatan**

Pada dasarnya, mencuri bukanlah perbuatan yang baik. Begitu juga membantu pencuri seperti Pak Kanech.

**H. Pusing Makan-Makan**

|  |  |
| --- | --- |
| Time limit | 1 s |
| Memory limit | 64 MB |

**Deskripsi**

Minggu ini merupakan minggu yang sibuk. Bayangkan saja, Anda mendapatkan N undangan untuk berbagai pesta makan! Pesta ke-i memiliki durasi yang pasti, dan berjalan dari waktu Si sampai Ei inklusif. Setiap pesta memiliki aturan, bahwa setiap hadirin pesta tersebut harus datang tepat pada waktu Si, serta keluar pada waktu Ei. Sekarang, bukannya bingung memikirkan menghadiri pesta yang mana saja, anda justru memikirkan hal yang lain. Ada berapa cara berbeda untuk menghadiri setidaknya 1 pesta, tanpa melanggar aturan tiap pesta tersebut? Karena angka ini bisa sangat besar, keluarkan hasilnya modulo 1.000.000.007.

**Format Masukan**

Baris pertama berisi sebuah bilangan bulat N, banyaknya pesta.

N baris selanjutnya berisi dua buah bilangan bulat Si dan Ei, masing-masing menyatakan waktu pesta dimulai dan waktu pesta berakhir.

**Format Keluaran**

Satu baris berisi sebuah bilangan bulat, banyak cara menghadiri setidaknya 1 pesta modulo 1.000.000.007.

**Contoh Masukan**

3

1 3

2 4

5 9

**Contoh Keluaran**

5

**Batasan**

* 1 ≤ N ≤ 100.000
* 1 ≤ Si ≤ Ei ≤ 100.000
* Untuk tiap i < j, Si≠ Sj
* Untuk tiap i < j, Ei≠ Ej

**I. Takada, Kembalian**

|  |  |
| --- | --- |
| Time limit | 1 s |
| Memory limit | 32 MB |

**Deskripsi**

Pak Chanek sedang berbelanja di suatu supermarket. Ketika mau membayar, ia berkenalan dengan kasir supermarket tersebut, yaitu Takada. Total pembelian Pak Chanek adalah M rupiah. Ketika mengecek alatnya, Takada berkata "maaf Pak Chanek, uang pas ya. Tak ada kembalian". Pak Chanek lalu merogoh dompetnya, dan mendapati bahwa ia membawa N koin. Koin ke-i bernilai Ci rupiah. Sekarang, Pak Chanek ingin membayar tepat M rupiah dengan sesedikit mungkin koin yang digunakan. Bantulah Pak Chanek mencari jumlah minimal koin yang diperlukan, karena antrian sudah semakin menumpuk!

**Format Masukan**

Baris pertama berisi dua buah bilangan bulat N dan M, banyak koin di dompet Pak Chanek dan total pembelian Pak Chanek.

Baris kedua berisi N buah bilangan bulat Ci, nilai dari koin ke-i.

**Format Keluaran**

Satu baris berisi sebuah bilangan bulat, banyak koin minimal yang diperlukan, atau "-1" (tanpa tanda petik) jika ternyata tidak ada kombinasi koin manapun yang berjumlah M rupiah.

**Contoh Masukan**

5 7

1 1 3 4 5

**Contoh Keluaran**

2

**Batasan**

* 1 ≤ N ≤ 1.000
* 1 ≤ M ≤ 30.000
* 1 ≤ Ci ≤ 30.000

**J. Mengambil Batu**

|  |  |
| --- | --- |
| Time limit | 2 s |
| Memory limit | 64 MB |

**Deskripsi**

Pak Chanek adalah seorang ahli geologi. Pada suatu saat, dia pergi ke sebuah hutan untuk melakukan risetnya. Tak disangka, dia terperosok di suatu tempat dan terjebak di dalam gua! Untungnya, gua tersebut dipenuhi banyak batu yang dibutuhkan Pak Chanek untuk melengkapi risetnya. Setelah dihitung, dalam gua tersebut terdapat N batu yang dinomori dari 1 sampai N. Masing-masing batu memiliki berat Wi dan nilai Vi yang merupakan ukuran keberhargaan suatu batu bagi ahli geologi. Tentu Pak Chanek ingin mengambil sebanyak mungkin batu. Sayangnya, tas yang dimiliki Pak Chanek hanya mampu mengangkut beban maksimum M kg. Pak Chanek ingin mengambil batu-batu tersebut sehingga nilai total yang didapatkannya maksimal. Karena batu-batu tersebut sangat banyak, Pak Chanek bingung batu mana saja yang harus diambil. Sebagai temannya, Anda diminta untuk membantu Pak Chanek memilih batu-batu yang harus diambil sehingga nilai total dari batu-batu tersebut maksimal.

**Format Masukan**

Baris pertama berisi dua buah bilangan bulat, N dan M, banyak batu dan beban maksimum yang dapat dibawa tas Pak Chanek.  
Baris kedua berisi N bilangan bulat, Wi, yang merupakan berat batu ke-i.  
Baris ketiga berisi N bilangan bulat, Vi, yang merupakan nilai batu ke-i.

**Format Keluaran**

Sebuah bilangan bulat yang merupakan nilai total maksimum yang dapat diperoleh Pak Chanek.

**Contoh Masukan 1**

4 10

100 5 3 6

100 1 3 2

**Contoh Keluaran 1**

5

**Batasan**

* 1 <= N <= 100
* 1 <= M <= 109
* 1 <= Wi <= 107
* 1 <= Vi <= 1000

**K. Menyusun Barang**

|  |  |
| --- | --- |
| Time limit | 2 s |
| Memory limit | 64 MB |

**Deskripsi**

Pak Chanek memiliki N buah barang yang dinomori dari 1 sampai N. Barang ke-i memiliki berat Wi. Pak Chanek ingin menyusun barang-barangnya sehingga barang-barang yang dipilih memiliki total berat tepat M. Karena Pak Chanek akan merasa lelah apabila menyusun barang yang terlalu banyak maka Pak Chanek juga ingin meminimalkan banyak barang yang dipilih. Bantulah Pak Chanek untuk menentukan barang-barang yang harus dipilihnya!

**Format Masukan**

Baris pertama berisi dua buah bilangan bulat N dan M, masing-masing menyatakan banyak barang yang dimiliki Pak Chanek dan total berat yang diinginkan Pak Chanek.

Baris berikutnya berisi N buah bilangan bulat yang menyatakan berat barang ke-i.

**Format Keluaran**

Baris pertama berisi sebuah bilangan bulat X yang menyatakan jumlah barang paling sedikit yang bisa dipilih Pak Chanek sehingga total berat barang yang dipilih adalah M.

Baris berikutnya berisi X buah bilangan bulat yang menyatakan barang-barang yang dipilih Pak Chanek. Jika terdapat lebih dari satu kemungkinan konfigurasi pemilihan barang, keluarkan konfigurasi yang terkecil secara leksikografis.

Apabila tidak ada konfigurasi pemilihan yang mungkin, keluarkan "-1" (tanpa tanda kutip).

Sebuah urutan A dikatakan lebih kecil secara leksikografis dari urutan B yang memiliki panjang sama apabila terdapat i, sehingga Aj = Bj untuk 1 ≤j < i dan Ai < Bi.

**Contoh Masukan 1**

5 2

1 1 2 2 3

**Contoh Keluaran 1**

1

3

**Contoh Masukan 2**

3 6

1 2 7

**Contoh Keluaran 2**

-1

**Contoh Masukan 3**

3 6

1 2 4

**Contoh Keluaran 3**

2

2 3

**Batasan**

* 1 ≤ N, M ≤ 1.000
* 1 ≤ Wi ≤ 1.000