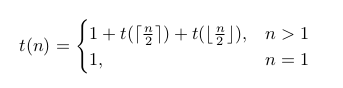
**A. Analisis Memori**

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

**Deskripsi**

Struktur data lanjutan yang banyak aplikasinya di *Competitive Programming* adalah Segment Tree. Tenang, anda tidak diminta untuk mengimplementasikan Segment Tree. Kali ini, anda akan menganalisis memori yang digunakan di Segment Tree. Secara formal, Segment Tree yang memiliki n buah data akan membutuhkan memori sesuai dengan fungsi berikut:



Catatan: itu merupakan lambang *ceiling* (pembulatan ke atas) dan *floor* (pembulatan ke bawah).

Pertanyaannya, jika anda memiliki N buah data, berapakah memori yang akan anda butuhkan?

**Format Masukan**

Satu baris berisi sebuah bilangan bulat N, banyaknya data.

**Format Keluaran**

Satu baris berisi sebuah bilangan bulat, banyaknya memori yang diperlukan.

**Contoh Masukan**

2

**Contoh Keluaran**

3

**Batasan**

* 1 ≤ N ≤ 105

**B. Banyak Jalan Menuju Pintu Keluar**

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

**Deskripsi**

Anda sedang terjebak di sebuah *maze* yang untungnya, tidak terlalu besar. *Maze*ini berbentuk persegi panjang berukuran N\*M, dan terdiri dari petak-petak berukuran 1\*1. Sehingga, terdapat N baris dan M kolom di *maze*ini. Pojok kiri atas berada di petak (1,1), sedangkan kanan bawah berada di petak (N,M). *Maze* ini cukup aneh, karena di dalam *maze* ini, anda hanya dapat berpindah ke petak yang berada tepat di kanan atau tepat di bawah (tentunya jika petak itu ada!). Saat ini, anda berada di pintu masuk, alias petak (1,1). Pintu keluar berada di petak (N,M). Tentukan, ada berapa cara berbeda anda bisa mencapai pintu keluar?

**Format Masukan**

Satu baris berisi 2 buah bilangan bulat N dan M, yang menyatakan ukuran *maze*.

**Format Keluaran**

Satu baris berisi sebuah bilangan bulat, banyak cara mencapai pintu keluar.

**Contoh Masukan**

3 2

**Contoh Keluaran**

3

**Batasan**

* 1 ≤ N, M ≤ 10

**C. Menara Hanoi**

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

**Deskripsi**

Apakah anda tahu tentang Menara Hanoi? Menara Hanoi merupakan permainan dengan 3 tiang dan N cakram, yang ukurannya berbeda-beda semua. Awalnya, semua cakram berada pada salah satu tiang, dan terurut, dengan cakram terbesar berada di paling bawah dan cakram terkecil berada di paling atas. Sebut saja tiang ini tiang A, dan dua tiang lainnya tiang B dan tiang C. Tujuan permainan ini adalah memindahkan N cakram tersebut ke tiang C dengan aturan berikut:

* Pada satu waktu, hanya satu cakram yang boleh dipindahkan.
* Cakram yang boleh dipindahkan merupakan cakram yang berada di posisi paling atas salah satu tiang. Cakram akan dipindahkan ke posisi paling atas salah satu tiang yang lain.
* Pada saat memindahkan, cakram yang dipindahkan tidak boleh lebih besar dari cakram teratas tiang tujuan.

Tentunya, pada akhirnya cakram-cakram itu akan terurut seperti semula, hanya saja berada pada tiang C. Diberikan N, banyak cakram, anda harus mengeluarkan solusi pemindahan cakram-cakram tersebut yang membutuhkan langkah paling sedikit.

**Format Masukan**

Satu baris berisi sebuah bilangan bulat N, banyaknya cakram. Cakram-cakram tersebut memiliki ukuran 1 sampai N.

**Format Keluaran**

Beberapa baris, dengan setiap baris memiliki format "X Y Z", yang berarti memindahkan cakram berukuran X dari puncak tiang Y ke puncak tiang Z.

**Contoh Masukan**

3

**Contoh Keluaran**

1 A C

2 A B

1 C B

3 A C

1 B A

2 B C

1 A C

**Batasan**

* 1 ≤ N ≤ 10

**D. K-Permutasi**

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

**Deskripsi**

Permutasi adalah penyusunan kembali suatu kumpulan objek dalam urutan yang berbeda dari urutan yang semula. Sebagai contoh, 1234 dapat disusun kembali menjadi 4231. Terdapat 24 permutasi berbeda dari 1234. Diberikan sebuah N, anda ingin membangkitkan semua permutasi 123...N yang valid. Hanya saja, karena itu membosankan, anda menambahkan batasan tambahan yaitu K. Sekarang, semua permutasi yang dibangkitkan valid hanya jika selisih setiap elemen yang bersebelahan tidak melebihi K.

**Format Masukan**

Satu baris berisi 2 buah bilangan bulat, N dan K.

**Format Keluaran**

Beberapa baris, dimana setiap baris berisi sebuah permutasi yang valid. Urutkan permutasi berdasarkan urutan leksikografis. Lalu, pisahkan setiap bilangan dalam sebuah permutasi menggunakan spasi.

**Contoh Masukan 1**

3 2

**Contoh Keluaran 1**

1 2 3

1 3 2

2 1 3

2 3 1

3 1 2

3 2 1

**Contoh Masukan 2**

3 1

**Contoh Keluaran 2**

1 2 3

3 2 1

**Batasan**

* 1 ≤ N, K ≤ 8