

Pululu

Time limit: 1 s

Memory limit: 32 MB

Pululu adalah spesies hewan baru yang ditemukan di Kepulauan Galalala. Hewan unik ini memiliki dua mata yang bisa digerakkan secara terpisah. Dengan kata lain, pululu dapat melihat ke paling banyak dua arah (atas, bawah, kiri, atau kanan) yang berbeda. Walaupun pululu biasanya memakan hewan-hewan lain yang lebih kecil darinya, pululu memiliki kecenderungan untuk melakukan kanibalisme jika ada pululu lain yang lengah.

Pak Chanek, seorang pecinta hewan unik, ingin memelihara sebanyak mungkin pululu di halamannya. Akan tetapi, karena ia sangat mencintai hewan, ia tidak ingin melihat hewan peliharaannya memakan satu sama lain. Halaman Pak Chanek berbentuk persegi panjang dengan ukuran NN petak \times MM petak. Karena ukurannya, setiap pululu yang diletakkan pada halaman Pak Chanek akan menempati petak berukuran 1×1 dan tidak mungkin meletakkan dua atau lebih pululu pada petak yang sama. Pululu akan memakan sesamanya jika ada pululu yang berhadapan dengan lebih dari dua pululu lain. Dua pululu dikatakan berhadapan jika mereka berada pada kolom atau baris yang sama pada halaman Pak Chanek dan tidak ada pululu lain yang berada di antara kedua pululu tersebut.

Jika diketahui ukuran halaman Pak Chanek, hitunglah jumlah maksimal pululu yang bisa Pak Chanek pelihara.

Format Masukan

Masukan terdiri dari satu baris yang berisi dua buah bilangan bulat NN dan MM , dipisahkan oleh spasi.

Format Keluaran

Keluaran berupa sebuah bilangan bulat yang menyatakan jumlah maksimal pululu yang bisa dipelihara Pak Chanek di halamannya.

Contoh Masukan

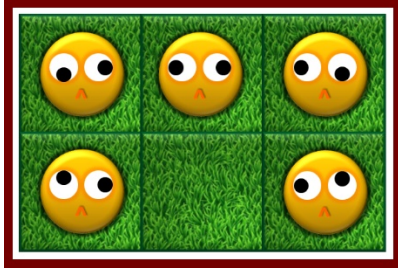
```
2 3
```

Contoh Keluaran

```
5
```

Penjelasan

Pak Chanek dapat menempatkan pululu dengan konfigurasi sebagai berikut.



Batasan

Terdapat dua subsoal pada soal ini.

Subsoal 1 (28 poin)

- $1 \leq N, M \leq 4$

Subsoal 2 (72 poin)

- $1 \leq N, M \leq 1000000$

XOR A B

Time limit: 1 s

Memory limit: 32 MB

Setelah beberapa kali gagal mengikuti ujian saringan masuk beberapa perguruan tinggi negeri unggulan, Pak Chanek akhirnya bekerja sebagai seorang *lighting technician* di suatu konser musik. Tugas yang dikerjakan Pak Chanek cukup sederhana, yaitu mengatur konfigurasi saklar-saklar lampu agar sesuai dengan petunjuk yang diberikan oleh manajernya.

Sayangnya, karena keterbatasan waktu, manajer Pak Chanek tidak dapat memberitahukan dengan terperinci saklar-saklar lampu mana saja yang harus dihidupkan. Ia hanya memberikan dua buah bilangan bulat A dan B di secarik kertas dan mengatakan bahwa konfigurasi saklar-saklar lampu yang benar dapat diketahui dengan menghitung nilai biner dari hasil

operasi $A \oplus (A+1) \oplus (A+2) \oplus \dots \oplus B$ atau $B \oplus (A+1) \oplus (A+2) \oplus \dots \oplus B$. (Simbol \oplus menyatakan operasi bitwise XOR.)

Ketidaktersediaan kertas dan alat tulis di lapangan membuat Pak Chanek bingung bukan kepalang. Ia tidak dapat membuat penghitungan di tempat kerjanya! Untuk memperburuk keadaan, konser akan dimulai beberapa menit lagi dan pengunjung sudah memadati lokasi konser. Di tengah keputusasaannya, Pak Chanek hanya dapat berharap kepada Anda, programmer andal yang sedang membaca soal ini di layar komputer, untuk membantunya. Bantulah Pak Chanek menyelesaikan tugasnya!

Format Masukan

Masukan terdiri atas satu baris yang berisi dua buah bilangan bulat A dan B, dipisahkan oleh spasi.

Format Keluaran

Keluaran berupa sebuah bilangan biner sesuai dengan deskripsi soal, tanpa *leading zero*.

Contoh Masukan

```
2 10
```

Contoh Keluaran

```
1010
```

Batasan

Terdapat tiga subsoal pada soal ini. Untuk setiap subsoal berlaku

- $0 \leq A \leq B \leq 2^{64}-1$ $0 \leq A \leq B \leq 2^{64}-1$.

Subsoal 1 (30 poin)

- $B-A \leq 1000000$ $B-A \leq 1000000$

Subsoal 2 (30 poin)

- $A=1$ $A=1$

Subsoal 3 (40 poin)

- Tidak ada batasan tambahan.

Chain Frost

Time limit: 1 s

Memory limit: 32 MB

Lich adalah salah satu *hero* dalam permainan DOTA yang menggunakan kekuatan es untuk menghancurkan musuh-musuhnya. Dari berbagai macam kekuatan yang ia miliki, kekuatan yang paling ditakuti oleh *hero-hero* lainnya adalah *Chain Frost* karena dapat merambat dari target yang satu ke target yang lain.

Serangan *Chain Frost* dapat merambat sebanyak K kali, dimulai dari target T_1 . Untuk $1 \leq i \leq K$, saat *Chain Frost* mengenai target T_i , *Chain Frost* akan merambat ke target T_{i+1} , yaitu target lain yang terdekat dari T_i dan yang *bukan* terakhir kali merambatkan serangan (*i.e.* target T_{i-1} , saat $i \geq 2$). Jika *Chain Frost* dapat merambat ke dua atau lebih target, *Chain Frost* akan merambat ke target dengan nomor terkecil di antara target-target tersebut.

Karena kondisi geografis pada peta, mungkin saja suatu target tidak dapat merambatkan secara langsung serangan *Chain Frost* ke target-target tertentu, walaupun target-target tersebut tidak melanggar syarat perambatan yang dijelaskan sebelumnya. Untuk target-target yang terpisah secara geografis seperti ini, jarak antara keduanya tidak didefinisikan. Jika saat ini terdapat N buah target pada peta yang dinomori $0, 1, 2, \dots, N-1$, hitunglah jarak yang ditempuh oleh *Chain Frost* setelah memantul paling banyak K kali.

Format Masukan

Masukan terdiri dari dua bagian.

- Bagian pertama terdiri dari satu baris yang berisi empat buah bilangan bulat N , M , K , dan T . Nilai M menunjukkan jumlah data jarak antartarget akan diperinci pada bagian kedua masukan.
- Bagian kedua terdiri dari M baris. Untuk setiap $1 \leq i \leq M$, baris ke- i pada bagian ini berisi tiga buah bilangan bulat A_i , B_i , dan C_i yang secara berturut-turut menyatakan bahwa jarak antara target A_i dan B_i adalah C_i .

Format Keluaran

Keluaran berupa sebuah bilangan bulat yang menyatakan jarak lintasan yang ditempuh oleh *Chain Frost* setelah memantul paling banyak K kali.

Contoh Masukan

```
4 5 6 0
0 1 1
1 2 2
2 3 5
2 0 3
3 1 3
```

Contoh Keluaran

```
12
```

Penjelasan

Rute yang ditempuh *Chain Frost* adalah $0-1-2-0-1-2-0-1-2-0-1-2-0$.

Batasan

Terdapat tiga subsoal pada soal ini. Untuk setiap subsoal berlaku

- $0 \leq T, A, B < N \leq T, A, B < N$.

Subsoal 1 (22 poin)

- $0 \leq K \leq 20$
- $1 \leq N \leq 10$
- $0 \leq M \leq 20$
- Untuk setiap $1 \leq i \leq M$, $C_i = 1$
- Setiap target terhubung satu sama lain

Subsoal 2 (33 poin)

- $0 \leq K \leq 2000$
- $1 \leq N \leq 1000$
- $0 \leq M \leq 2000$
- Untuk setiap $1 \leq i \leq M$, $1 \leq C_i \leq 1000000$

Subsoal 3 (45 poin)

- $0 \leq K \leq 200000$
- $1 \leq N \leq 100000$
- $0 \leq M \leq 200000$
- Untuk setiap $1 \leq i \leq M$, $1 \leq C_i \leq 1000000$

Jalan-Jalan

Time limit: 2 s

Memory limit: 64 MB

Pada suatu hari, hiduplah seorang koboi bernama Tarzan. Ia sedang berkelana mencari sapinya yang hilang.

Sapi Tarzan hilang saat ia sedang tertidur lelap di bawah rindangnya pohon, di tengah hembusan angin sepoi-sepoi. Sudah 2089 hari ia berkelana dan masih belum ditemukan juga tanda-tanda keberadaan sang sapi, Nabilah. Perlu diketahui, meskipun diberi nama Nabilah, sapi tersebut jantan. Kejombloan Tarzanlah yang menyebabkan dirinya memberikan nama seorang wanita kepada sapinya.

Memasuki hari pencarian ke-2090, Tarzan pun menginjak usia 95 tahun. Tubuhnya yang sudah renta dan lemah menyebabkan ia hanya mampu berjalan sejauh 100 meter setiap jamnya. Saat sedang beristirahat di sebuah pondok, Tarzan pun berpikir, "Jika saya bergerak selambat ini, bagaimana mungkin saya akan menemukan Nabilah?"

Tiba-tiba, muncullah sosok lelaki tua bernama Nazrat. Nazrat memang sudah tua, tetapi ia memiliki tubuh yang sama kuat dan sama bugarnya dengan pemuda berusia 40 tahun. Melihat tubuh Nazrat yang kekar, Tarzan pun berandai-andai. Andai saja Tarzan dapat menjadi seperti Nazrat. Ia pasti akan mampu melanjutkan perjalanannya hidupnya mencari Nabilah.

Demikianlah hari berganti hari, bulan berganti bulan, dan musim berganti musim hingga... wow! Ternyata keinginan Tarzan dikabulkan oleh Dewi Koboi. Tarzan berubah menjadi seperti Nazrat! Berbekal kekuatan yang baru, Tarzan pun dapat kembali melanjutkan perjalanannya mencari Nabilah. Ia ingin dapat kembali ke kota, hidup bersama dengan Nabilah, setelah ia menemukan sapinya tersebut.

Sayang sekali, belum lama menikmati kekuatan yang baru, Tarzan segera menyadari kekurangan yang dimiliki Nazrat. Setelah berubah menjadi seperti Nazrat, Tarzan kini hanya dapat berjalan ke arah timur dan selatan. Tarzan pun menjadi sering linglung, sehingga ia hanya dapat menentukan arah berjalan sebanyak N kali. Untuk $1 \leq i \leq N$, setelah menentukan arah untuk ke- i kalinya, ia akan berjalan lurus ke arah tersebut sejauh D_i satuan.

Walaupun harus berhadapan dengan kekurangan seperti ini, Tarzan tidak kehilangan semangatnya untuk bertemu dengan Nabilah. Agar pencariannya menjadi efisien, ia ingin agar luas daerah yang dicakup oleh perjalanannya seluas mungkin. Jika koordinat awal dan koordinat akhir Tarzan secara berturut-turut adalah $(0,0)$ dan (x,y) , maka luas daerah yang dicakup oleh perjalanan Tarzan adalah $x \cdot y$. Karena Tarzan sudah tua, bantulah Tarzan menentukan luas daerah terbesar yang dapat dicakup dan pilihan arah untuk mencakup luas daerah cakupan terbesar tersebut.

Format Masukan

Baris pertama masukan berisi satu bilangan bulat T yang menyatakan banyaknya kasus uji.
Untuk setiap kasus uji

- baris pertama berisi sebuah bilangan bulat N ;
- baris kedua berisi N buah bilangan bulat D_1, D_2, \dots, D_N .

Format Keluaran

Keluaran untuk setiap kasus uji terdiri dari dua baris. Baris pertama berisi satu bilangan bulat AA yang menyatakan luas daerah terbesar yang dapat dicakup. Baris kedua berisi NN buah karakter yang menyatakan cara memilih arah jalan agar luas daerah yang dicakup sama dengan AA . Untuk setiap $1 \leq i \leq N$, cetak **R** jika Tarzan harus berjalan ke timur pada pemilihan arah jalan ke- i atau **D** jika Tarzan harus berjalan ke selatan pada pemilihan arah jalan ke- i . Jika terdapat dua atau lebih konfigurasi pemilihan arah jalan yang sama-sama mencakup daerah seluas AA , keluarkan yang mana saja.

Contoh Masukan

```
2
4
1 2 3 4
4
2 3 4 5
```

Contoh Keluaran

```
25
RDDR
49
DRRD
```

Batasan

- $1 \leq T \leq 50$
- $1 \leq N \leq 200$
- $1 \leq D_i \leq 500$, untuk setiap $1 \leq i \leq N$

Relokasi Warga KaliJodie

Time limit: 500 ms

Memory limit: 64 MB

Deskripsi

Pak Chanek saat ini sedang pusing memikirkan rencana relokasi warga di daerah KaliJodie, pasalnya untuk merelokasi rumah-rumah tersebut membutuhkan waktu dan dana yang cukup besar. Untuk memudahkan pemindahan, Pak Chanek memutuskan untuk bekerjasama dengan Mba Jodie, seorang agen FBI sekaligus *centeng* daerah KaliJodie untuk memindahkan semua warga secara paksa ke daerah yang disediakan.

Daerah KaliJodie merupakan jalan lurus yang bisa direpresentasikan menjadi sebuah bidang lurus satu dimensi. Terdapat NN warga, dimana warga ke- i menetap di rumah yang terletak pada koordinat X_i .

Pada daerah tersebut juga akan ditentukan batas daerah yaitu koordinat LL dan RR yang menjadi tempat tujuan relokasi, semua warga yang tidak berada pada daerah tersebut akan dipindahkan secara paksa menuju batas daerah tersebut. Karena minimnya biaya, Pak Chanek hanya mampu menyediakan daerah relokasi sepanjang KK meter. Artinya, jarak dari LL ke RR haruslah kurang dari atau sama

dengan KK. Pak Chanek juga ingin meminimalisasi biaya penggusuran, biaya penggusuran rumah yang berada pada koordinat MM adalah kuadrat dari jarak terdekat titik MM ke daerah relokasi yaitu koordinat LL sampai dengan RR inklusif.

Pak Chanek dan Mbak Jodie sangat sibuk bernegosiasi dengan warga setempat saat ini, sehingga tidak sempat memikirkan lokasi relokasi terbaik untuk menekan total dana penggusuran. Oleh karena itu bantulah Pak Chanek dan Mbak Jodie menyelesaikan permasalahan ibu kota yang tidak berujung ini dengan menentukan biaya minimum yang harus mereka persiapkan.

Format Masukan

Baris pertama berisi dua buah bilangan bulat NN dan KK yang dipisahkan oleh spasi. Baris berikutnya berisi NN buah bilangan bulat X_i ($1 \leq i \leq N$) yang dipisahkan dengan spasi.

Format Keluaran

Keluarkan sebuah baris yang merupakan total biaya relokasi minimum yang harus dikeluarkan Pak Chanek. Jawaban dengan kesalahan relatif kurang dari 10^{-6} akan dianggap benar. Catatan:

misal aa adalah jawaban juri dan bb adalah jawaban peserta, maka jawaban peserta akan dianggap benar jika dan hanya jika $|b-a| \max(1, a) \leq 10^{-6}$.

Contoh Masukan 1

```
3 1
1 3 5
```

Contoh Keluaran 1

```
4.5
```

Contoh Masukan 2

```
3 10
1 3 5
```

Contoh Keluaran 2

```
0
```

Penjelasan Contoh

Untuk kasus uji pertama, daerah relokasi paling optimal adalah pada titik 2.5 sampai dengan 3.5 (panjang 1 meter).

Biaya relokasi adalah $1.5^2 + 1.5^2 = 4.5$ (memindahkan warga pada titik 1 ke 2.5 dan warga pada titik 5 ke 3.5, warga pada titik 3 tidak perlu dipindahkan karena sudah terletak diantara 2.5 dan 3.5)

Untuk kasus uji kedua, jika daerah relokasi terletak pada titik 1 sampai dengan 5 (panjang 4 meter) relokasi tidak dibutuhkan.

Subsoal

Untuk semua subsoal berlaku

- $1 \leq K \leq 20.000$
- $0 \leq X_i \leq 20.000$

Subsoal 1 (15 Poin)

- $1 \leq N \leq 20$
- $1 \leq K \leq 20$
- $0 \leq X_i \leq 20$

Subsoal 2 (15 Poin)

- $1 \leq N \leq 20$

Subsoal 3 (30 Poin)

- $1 \leq N \leq 2.000$

Subsoal 4 (40 Poin)

- $1 \leq N \leq 20.000$

Odd Bubble Sort II

Time limit: 3 s

Memory limit: 64 MB

Anda sedang asyik membaca buku di taman. Tiba-tiba, seorang teman Anda datang sambil membawa sepotong kertas dengan wajah bingung. Anda pun bertanya ada masalah apa. Rupanya teman Anda diberikan sebuah *pseudocode* dari suatu algoritme bernama *oddBubbleSort2*. Karena teman Anda tahu Anda sangat menyukai algoritme dan pemrograman, dia memberikan kertas tersebut kepada Anda; siapa tahu Anda berminat untuk memahami kertas tersebut. Nah, berikut adalah *pseudocode* algoritme *oddBubbleSort2* tersebut dari teman Anda.

```
procedure oddBubbleSort2(A : array yang merupakan permutasi dari N buah bilangan berbeda dari 1 hingga N)
    N = length(A)
    print "N = "
    print N
    print newline
```



```

    for i = 1 to 1 inclusive do
        print "i = "
        print i
        print " : "
        for j = 2 to N inclusive do
            if A[j-1] > A[j] then
                swap(A[j-1], A[j])
            else if
                print space
                print j
            end if
        end for
        print newline
    end for
    print "---"
    print newline
end procedure

```

Melihat Anda bersemangat memahami algoritme tersebut, teman Anda pun ikut-ikutan mempelajarinya. Kemudian, dia memberikan Anda beberapa contoh keluaran dari prosedur tersebut dan meminta Anda menentukan berapa banyak kemungkinan isi awal *array* A sebelum diurutkan.

Format Masukan

Baris pertama berisi bilangan bulat T, yaitu banyaknya kasus uji. Beberapa baris berikutnya merupakan keluaran dari algoritme *oddBubbleSort2* tersebut. Perhatikan bahwa secara tidak langsung setiap kasus uji dipisahkan oleh string "---".

Format Keluaran

Untuk setiap kasus uji, cetaklah banyak kemungkinan isi awal *array* A setelah di-*modulo* dengan 1000000007.

Contoh Masukan

```

3
N = 5
i = 1 : 3
---
N = 3

```

```
i = 1 : 2 3
---
N = 3
i = 1 :
---
```

Contoh Keluaran

```
12
1
2
```

Batasan

- $T \leq 128$
- $1 \leq N \leq 100000$