

Ini Ibu Budi

Time Limit: 1 second(s)
Memory Limit: 32 megabyte(s)

Belajar matematika itu menyenangkan! Setidaknya, itulah yang dirasakan oleh Budi yang sedang duduk di bangku SD kelas 1 ketika diperkenalkan dengan pelajaran matematika. Andi memiliki dua buah apel di tangannya, jika ibu guru memberi lima buah apel lagi kepada Andi, berapa apel yang dimiliki Andi? Tentunya tujuh (dua ditambah lima)! Konsep penjumlahan seperti ini sangat menarik bagi Budi.

Penasaran dengan penjumlahan, Budi mencari soal-soal latihan dan menemukan soal yang lebih menarik lagi. Setiap hari ibu guru memberikan 3 buah apel kepada Andi, setelah 5 hari berlalu, berapakah jumlah apel yang diberikan oleh ibu guru kepada Andi? Budi mulai berhitung; hari pertama ibu guru memberikan 3 buah apel, di hari kedua juga 3 apel, di hari ketiga kembali 3 apel... aha! jumlah apel yang diberikan ibu guru adalah $3 + 3 + 3 + 3 + 3$, yaitu... lima belas!

Budi dengan segera mengerjakan soal-soal latihan serupa yang ia temukan, dan sekarang ia meminta bantuan anda untuk memeriksa hasil pekerjaannya. Soal latihan yang ia kerjakan berbentuk: *"Setiap hari ibu guru memberikan A buah apel kepada Andi, setelah B hari berlalu, berapakah jumlah seluruh apel yang diberikan oleh ibu guru kepada Andi?"*

Input

Baris pertama dari input adalah sebuah bilangan bulat T ($T \leq 100$) yang menyatakan banyaknya kasus yang harus anda tangani. Setiap kasus terdiri dari dua buah bilangan bulat A dan B ($1 \leq A, B \leq 20$) yang dipisahkan oleh sebuah spasi. A dan B ini merepresentasikan soal yang diberikan Budi pada deskripsi permasalahan di atas.

Output

Untuk setiap kasus, output dalam satu baris, "Kasus #X: Y" (tanpa kutip) dengan X adalah nomor kasus dimulai dari 1 secara berurutan, dan Y adalah jawaban dari soal Budi untuk kasus yang bersangkutan.

Sample Test

Input

```
4
3 5
2 4
5 2
7 9
```

Output

```
Kasus #1: 15
Kasus #2: 8
Kasus #3: 10
Kasus #4: 63
```

Notes

Penjelasan untuk contoh input 1:

Kasus ini sesuai dengan contoh pada deskripsi soal di atas. Setiap hari ibu guru memberikan A (= 3) buah apel kepada Andi, setelah B (= 5) hari berlalu, berapakah jumlah apel yang diberikan oleh ibu guru kepada Andi? Jawabannya: 15.

Penjelasan untuk contoh input 2:

$$2 + 2 + 2 + 2 = 8$$

Penjelasan untuk contoh input 3:

$$5 + 5 = 10$$

Penjelasan untuk contoh input 4:

$$7 + 7 + 7 + 7 + 7 + 7 + 7 + 7 + 7 = 63$$

Penggemar Berat

Time Limit: 1 second(s)

Memory Limit: 32 megabyte(s)

Babak penyisihan BPCHS 2015 (Bamboo Programming Contest for High School Students 2015) akan segera digelar. Total 500 peserta dari berbagai penjuru Indonesia akan bertempur menggunakan logika mereka untuk memperebutkan 10 peringkat teratas. Ya, kontes tahun ini memang salah satu kontes paling ramai dalam sejarah BPCHS.

Sebelumnya, memang tidak banyak yang mengetahui tentang BPCHS ini. Namun berkat kerja keras dari panitia BPCHS, akhirnya mereka mendapatkan *official endorsement* dari seorang tokoh masyarakat, Pak Kumis. Sangat sulit untuk mendapatkan endorsement dari bapak ini karena jamnya yang sibuk, serta standarnya yang sangat tinggi (tidak jarang beliau menolak untuk meng-endorse kontes yang lain).

Perjanjian antara Pak Kumis dan panitia BPCHS adalah, Pak Kumis harus menuliskan promosi tentang BPCHS pada akun sosial medianya, *Lovebook*, sebanyak 2 kali sehari selama 2 minggu. Tidak heran selama 2 minggu tersebut, jumlah peserta meningkat dari 12 orang menjadi 500 orang.

Salah satu peserta yang ternyata juga mengidolakan Pak Kumis adalah Sandi. Awalnya dia tidak tahu tentang kontes BPCHS, namun ketika sang idola mempromosikan tentang BPCHS, Sandi sangat terinspirasi untuk memenangkan BPCHS tahun ini karena dia tahu Pak Kumis akan menyerahkan hadiah kepada sang pemenang. Tentu ini akan menjadi hal yang diidamkan oleh Sandi, bisa bersalaman dan foto bersama sang idola.

Sandi yang sangat senang terhadap *post* dari Pak Kumis di *Lovebook*, tanpa sadar menekan *tombol Cinta* berkali kali. Tombol Cinta di *Lovebook* bekerja sebagai berikut. Terdapat 2 kemungkinan status cinta: Sandi mencintai post Pak Kumis, atau Sandi tidak mencintai post Pak Kumis. Setiap kali tombol Cinta ditekan, status cinta Sandi akan bertukar. Yaitu dari cinta menjadi tidak cinta, atau dari tidak cinta menjadi cinta. Tentunya status awal dari tombol ini adalah tidak cinta.

Sandi yang terlalu senang, menekan tombol Cinta sebanyak N kali. Namun dia tidak tahu yang sebenarnya terjadi apakah dia mencintai post Pak Kumis atau malah tidak mencintainya.

Input

Baris pertama berisi sebuah bilangan T ($T \leq 1.000$) yang menyatakan banyaknya kasus yang harus ditangani. Untuk setiap kasus berisi sebuah bilangan N ($1 \leq N \leq 1.000$) yang menyatakan berapa kali Sandi menekan tombol Cinta.

Output

Untuk setiap kasus, cetak " Kasus \#X: Y " dimana X adalah nomor kasus dimulai dari 1, dan Y adalah "*CINTA*" (tanpa kutip) jika status Sandi adalah cinta post Pak Kumis, atau "*TIDAK CINTA*" (tanpa kutip) jika status Sandi adalah tidak cinta post Pak Kumis.

Sample Tests

Input

2
1
2

Output

Kasus #1: CINTA
Kasus #2: TIDAK CINTA

Notes

Penjelasan untuk contoh input 1:

Sandi menekan tombol Cinta sebanyak satu kali, pada penekanan pertama, status Sandi menjadi mencintai post Pak Kumis.

Penjelasan untuk contoh input 2:

Sandi menekan tombol Cinta sebanyak dua kali, artinya status Sandi akan berubah: (mula-mula) tidak cinta → (setelah penekanan pertama) cinta → (setelah penekanan kedua) tidak cinta.

Deduksi Peringkat

Time Limit: 1 second(s)

Memory Limit: 32 megabyte(s)

Akhir-akhir ini Tedy sangat gemar memainkan Makanio, sebuah permainan *online* yang ia temukan secara tidak sengaja beberapa bulan yang lalu. Pada permainan ini, peringkat pemain ditentukan hanya berdasarkan poin yang berhasil diperoleh pemain; semakin besar poin yang diraih, maka peringkat pemain tersebut akan semakin tinggi (semakin mendekati peringkat 1).

Suatu hari Tedy berhasil mendapatkan poin sebesar M pada permainan Makanio, dan Tedy ingin segera mengetahui peringkat yang ia dapatkan. Namun, peringkat resmi pada permainan ini hanya diumumkan seminggu sekali, sehingga Tedy harus menunggu beberapa hari hingga waktu pengumumannya.

Setelah bermain-main untuk beberapa waktu, Tedy menyadari bahwa ia bisa mendapatkan informasi poin dari seluruh pemain yang ada (tentunya dengan cara yang legal). Dengan demikian, tentunya, secara teori, ia bisa menghitung sendiri peringkat yang ia peroleh berdasarkan poin-poin yang ada. Tetapi jumlah pemain Makanio sangatlah banyak, sehingga ia kewalahan menghitung peringkatnya. Tedy memang kurang pandai, tapi ia baik hati; bantulah Tedy dengan menulis program untuk menentukan peringkatnya.

Perlu diperhatikan, jika terdapat pemain dengan poin yang sama, maka pemain-pemain tersebut akan mendapatkan peringkat yang sama. Namun, peringkat berikutnya akan diloncati tergantung berapa banyak orang yang memiliki poin yang sama tadi. Dengan kata lain, peringkat seseorang adalah R jika terdapat tepat $R - 1$ pemain lain yang poinnya lebih baik dari dirinya. Untuk lebih jelasnya, perhatikan penjelasan contoh input di bawah.

Input

Baris pertama berisi sebuah bilangan bulat T ($T \leq 100$) yang menyatakan banyaknya kasus yang harus ditangani. Input untuk setiap kasus dimulai dengan dua buah bilangan bulat N dan M ($1 \leq N \leq 100$; $1 \leq M \leq 100.000$) yang menyatakan banyaknya pemain Makanio **selain** Tedy dan poin yang diperoleh Tedy secara berturut-turut. Baris berikutnya berisi N buah bilangan bulat P_i yang menyatakan poin dari pemain ke- i ($1 \leq P_i \leq 100.000$).

Output

Untuk setiap kasus, cetak "Kasus #K: X" dimana K adalah nomor kasus dimulai dari 1, dan X adalah peringkat Tedy yang mendapatkan poin sebesar M .

Sample Tests

Input

```
4
6 5
2 5 3 6 3 3
6 3
2 5 3 6 3 3
6 3
2 5 3 6 5 3
14 58477
11447 51296 22546 99789 50634 83232 15328 67370 9511 81785 92744 45504
39069 82813
```

Output

```
Kasus #1: 2
Kasus #2: 3
Kasus #3: 4
Kasus #4: 7
```

Notes

Penjelasan untuk contoh input 1:

Total ada 7 pemain (Tedy dan 6 pemain lainnya). Poin yang diperoleh Tedy adalah 5, sedangkan 6 pemain lainnya masing-masing memperoleh poin: 2, 5, 3, 6, 3, 3. Berikut adalah peringkat masing-masing pemain jika diurutkan:

#	Poin	Peringkat
1	6	1
2	5	2
3	5	2
4	3	4
5	3	4
6	3	4
7	2	7

Dari tabel di atas, Tedy yang mendapatkan poin sebesar 5 akan menduduki peringkat 2 bersama satu pemain lainnya.

Kacamata Anti-Akik

Time Limit: 1 second(s)

Memory Limit: 32 megabyte(s)

Selain menyukai Matematika, Budi juga sangat gemar membaca. Meskipun ia baru duduk di bangku kelas 1 SD, ia sudah menghabiskan (membaca habis) beberapa buku novel tebal, seperti serial Sherlock Holmes dan Harry Potter. Tentunya selain novel, ia juga membaca buku-buku ilmiah seperti Concrete Mathematics dan Introduction to Algorithms. Ibunya pun bahkan tidak mengetahui apakah Budi benar-benar mengerti apa yang ia baca atau tidak. Namun dampak dari kegemarannya membaca buku ini mulai terasa, penglihatannya mulai terganggu dengan rabun jauh (mata minus). Budi mau tidak mau terpaksa menggunakan teknologi hasil dari penemuan di abad ke-13: kacamata.

Kacamata yang dimiliki Budi tergolong unik (dan ajaib); kacamataanya mampu mengabaikan keberadaan batu akik dari pandangan Budi (ia namakan kacamata anti-akik). Untuk mempermudah permasalahan, kita fokuskan saja permasalahan ini pada bidang 1D (1 dimensi). Misalnya terdapat beberapa batu akik yang dibariskan secara berderetan/bersebelahan dalam satu barisan. Apabila jumlah batu akik yang bersebelahan tersebut adalah genap, maka Budi (dengan kacamataanya) tidak akan bisa melihat satupun batu akik tersebut. Apabila jumlahnya ganjil, maka Budi akan melihat hanya satu buah batu akik yang tepat berada di tengah barisan.

Contoh. 'X' merepresentasikan sebuah batu akik, dan '.' artinya kosong (tidak ada batu akik).

```
Susunan   : XXXXXX
Terlihat  : .....
```

Ada sejumlah genap batu akik yang berderetan, sehingga tidak ada satupun batu yang terlihat melalui kacamata anti-akik Budi.

```
Susunan   : XXXXX
Terlihat  : ..X..
```

Ada sejumlah ganjil batu akik yang berderetan, sehingga hanya batu akik yang berada di tengah barisan yang terlihat.

Jika ada lebih dari satu deret batu akik yang bersebelahan, maka batu-batu tersebut akan diperlakukan sesuai dengan kelompok/deretnya masing-masing.

Contoh.

```
Susunan   : ...XXX..XX.X.XXXXX..
Terlihat  : ....X.....X...X....
```

Ada 4 deret batu akik yang terpisah. Deret pertama berjumlah ganjil (3 batu), deret kedua berjumlah genap (2 batu), deret ketiga berjumlah ganjil (1 batu), dan deret keempat berjumlah ganjil (5 batu).

Susunan : ...XXXX.XX.XXX.X...XX
Terlihat :X..X.....

Ada 5 deret batu akik yang terpisah. Deret pertama berjumlah genap (4 batu), deret kedua berjumlah genap (2 batu), deret ketiga berjumlah ganjil (3 batu), deret keempat berjumlah ganjil (1 batu), dan deret kelima berjumlah genap (2 batu).

Ibu Budi tentu saja penasaran dengan kacamata baru Budi; untuk setiap susunan batu akik dalam satu baris, ia ingin mengetahui bagaimana (apa) yang terlihat oleh Budi melalui kacamata anti-akik nya.

Input

Baris pertama dari input adalah sebuah bilangan bulat T ($T \leq 100$) yang menyatakan banyaknya kasus yang harus anda tangani. Setiap kasus terdiri dari sebuah string S yang merepresentasikan susunan batu pada sebuah barisan. S hanya disusun atas dua karakter: 'x' yang merepresentasikan sebuah batu akik, dan '.' yang artinya kosong (tidak ada batu akik). Panjang barisan untuk setiap input tidak akan melebihi 100.

Output

Untuk setiap kasus, cetak dalam satu baris, "Kasus #X: Y" (tanpa kutip) dengan X adalah nomor kasus dimulai dari 1 secara berurutan, dan Y adalah susunan batu akik yang dilihat oleh Budi melalui kacamata anti-akik nya. Perhatikan bahwa Y dan S (input) harus memiliki panjang yang sama.

Sample Tests

Input

```
2
...xxx..xx.xxxxxx..
...xxxxxxxxxx..
```

Output

```
Kasus #1: ....X.....X...X....
Kasus #2: .....
```


Notes

Penjelasan untuk contoh input 1:

Kasus ini sesuai dengan contoh yang diberikan pada penjelasan permasalahan di atas. Terdapat 4 deret/kelompok batu akik berurutan pada kasus, yaitu "XXX", "XX", "X", dan "XXXXX".

- "XXX" karena jumlahnya ganjil, yang terlihat adalah ".X."
- "XX" karena jumlahnya genap, yang terlihat adalah ". ."
- "X" karena jumlahnya ganjil, yang terlihat adalah "X"
- "XXXXX" karena jumlahnya ganjil, yang terlihat adalah ". .X. ."

Penjelasan untuk contoh input 2:

Hanya ada 1 kelompok batu akik yang berurutan pada kasus ini. Karena jumlah batu akik pada kelompok ini adalah genap (10 batu), maka tidak ada batu akik yang akan terlihat melalui kacamata anti-akik Budi.

Roda Gigi

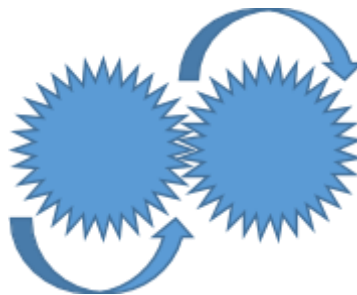
Time Limit: 1 second(s)

Memory Limit: 32 megabyte(s)

Vincent baru-baru ini memulai usaha baru sebagai seorang pengusaha bengkel. Ketertarikannya terhadap mesin bermotor, membuat dia akhirnya memutuskan untuk membuka usaha tersebut. Sebagai pengusaha baru, tentu Vincent membutuhkan karyawan dalam membantu tugasnya. Namun karena Vincent belum memiliki uang yang cukup banyak, akhirnya Vincent mengangkat anda sebagai karyawan satu-satunya.

Dengan bermodalkan kenalan dekat, akhirnya pelanggan pertama datang. Pelanggan ini mempunyai mesin dengan N buah roda gigi (*gear*) yang tersusun berderet dan bersentuhan dari kiri ke kanan. Roda gigi ini memiliki diameter yang sama, serta jumlah gerigi yang sama. Permasalahannya adalah, terdapat beberapa roda gigi yang berputar tidak sesuai arahnya ketika mesin dinyalakan.

Setiap roda gigi bisa berputar dalam dua arah: searah jarum jam (CW: *clockwise*) dan berlawanan dengan arah jarum jam (CCW: *counter clockwise*). Jika roda gigi pertama berputar ke arah CW dan roda gigi kedua juga berputar ke arah CW, maka kedua roda gigi tersebut akan tersendat dan mesin tersebut tidak akan jalan. Begitu juga sebaliknya, jika kedua roda gigi yang bersebelahan berputar ke arah CCW, maka kedua roda gigi tersebut juga akan tersendat. Mesin tersebut dapat bekerja dengan benar hanya jika setiap roda gigi yang bersebelahan berputar ke arah yang berbeda. Gambar berikut menjelaskan bagaimana keadaan dua roda gigi yang bersebelahan jika arah putarannya berbeda (tidak tersendat).



Kalian diberikan N buah roda gigi dengan kondisi berputar dari masing masing roda gigi ketika mesin dinyalakan. Bantu Vincent menghitung berapa roda gigi minimum yang harus Vincent ubah putarannya agar mesin tersebut berjalan secara normal.

Input

Baris pertama berisi sebuah bilangan T ($T \leq 100$) yang menyatakan banyaknya kasus yang harus ditangani. Untuk setiap kasus, berisi sebuah bilangan N ($1 \leq N \leq 1.000$) yang menyatakan jumlah roda gigi. Baris berikutnya berisi string dengan panjang N yang berisi karakter 'R' yang menyatakan roda gigi tersebut berputar searah jarum jam (CW), atau 'L' yang menyatakan roda gigi tersebut berputar berlawanan dengan arah jarum jam (CCW).

Output

Untuk setiap kasus, cetak "Kasus #K: M" dimana K adalah nomor kasus dimulai dari 1, dan M adalah jumlah roda gigi minimum yang harus Vincent ubah putarannya.

Sample Tests

Input

```
3
2
LR
2
LL
3
LLL
```

Output

```
Kasus #1: 0
Kasus #2: 1
Kasus #3: 1
```

Notes

Penjelasan contoh kasus 1:

Ada dua roda gigi yang bersebelahan dan arah putaran mereka berbeda, sehingga tidak perlu ada perubahan

Penjelasan contoh kasus 2:

Vincent dapat mengubah putaran pada roda gigi pertama menjadi CW/R – RL, ATAU mengubah putaran roda gigi kedua menjadi CW/R – LR.

Penjelasan contoh kasus 3:

Vincent harus mengubah putaran pada roda gigi kedua menjadi CW/R – LRL. Vincent juga bisa mengubah putaran pada roda gigi pertama dan ketiga menjadi CW/R – RLR, agar mesin tersebut bekerja; namun cara ini membutuhkan 2 roda gigi untuk diubah putarannya.

ROKET-1

Time Limit: 1 second(s)

Memory Limit: 32 megabyte(s)

Pada tahun 1969, NASA berhasil memberangkatkan manusia pertama kali ke bulan dan itu merupakan titik balik dalam penjelajahan luar angkasa. Tidak mau kalah, Vincent, pendiri Pusat Penelitian Antariksa Bambu University (PPABU) juga ingin menjelajahi luar angkasa. Dengan teknologi seadanya, Vincent bersama timnya berhasil menciptakan sebuah roket yang diberi nama ROKET-1 (ROKET karena bentuknya yang seperti roket, dan 1 karena itu adalah roket pertama).

ROKET-1 akhirnya diberi ijin untuk meluncur pada tanggal 2 Agustus 2015 setelah ditunda selama 8 tahun karena astronotnya belum memiliki SIM (Surat Ijin Meluncur). Ternyata permasalahan PPABU tidak hanya sampai di situ. Tugas utama ROKET-1 adalah mencatat jarak planet-planet yang dekat dengan bumi, namun karena keterbatasan teknologi, ROKET-1 tersebut hanya akan mencatat perkiraan jarak (dalam kilometer) planet tersebut dan melaporkannya dalam bentuk 10^N (sepuluh pangkat N).

Ternyata keterbatasan teknologi bukan hanya terjadi pada ROKET-1, namun ruang kendali ROKET-1 yang terdapat di bumi juga mengalami kendala. Sistem PPABU hanya bisa menerima angka yang kurang dari M. Dengan kata lain, jika ROKET-1 mengirimkan data jarak suatu planet ke bumi yang nilainya lebih besar atau sama dengan M, maka data tersebut tidak akan bisa disimpan pada sistem PPABU.

Vincent memiliki cara untuk mengakali agar data yang dikirim dari ROKET-1 tetap dapat dimasukkan. Cara tersebut adalah, dengan hanya menyimpan sisa hasil bagi dari jarak planet tersebut ke bumi dengan M. Hal ini akan membuat data yang disimpan tidak akan mencapai M. Karena data yang disimpan bukan data sebenarnya, Vincent hanya berharap teknologi masa depan dapat mengembalikan data tersebut ke nilai aslinya.

Vincent akan menyimpan data tersebut dengan cara sebagai berikut. Kita asumsikan J adalah data jarak antar suatu planet dengan bumi yang dikirimkan oleh ROKET-1 ke sistem PPABU, dan M adalah data maksimal yang dapat disimpan, serta X adalah data yang akan disimpan. Maka $J = M * P + X$ di mana P adalah hasil bagi J dengan M yang dibulatkan kebawah, serta X adalah sisa pembagian tersebut. Mari kita sebut operasi sisa pembagian tersebut dengan operasi modulo (mod), di mana $X = J \bmod M$ (sama halnya dengan 10 dibagi 6 hasilnya 1 dan bersisa 4. Sehingga $10 \bmod 6 = 1$ dan $10 \bmod 6 = 4$). Vincent tahu angka yang diberikan ROKET-1 sangat besar, sehingga dia memberikan teori bahwa:

$$A = (B * C) \bmod M \text{ jika dan hanya jika } A = ((B \bmod M) * (C \bmod M)) \bmod M$$

Teori ini juga berlaku untuk penjumlahan.

$$A = (B + C) \bmod M \text{ jika dan hanya jika } A = ((B \bmod M) + (C \bmod M)) \bmod M$$

Contoh penggunaan:

```
1 = 10 mod 3
1 = (5 + 5) mod 3
1 = ((5 mod 3) + (5 mod 3)) mod 3
1 = (2 + 2) mod 3
1 = 4 mod 3
1 = 1
```

Atau bisa juga:

```
1 = (2 * 5) mod 3
1 = ((2 mod 3) * (5 mod 3)) mod 3
1 = (2 * 2) mod 3
1 = 4 mod 3
1 = 1
```

Dengan teori yang diberikan Vincent, bantu dia untuk mengetahui berapa nilai yang harus disimpan pada sistem PPABU.

Input

Baris pertama berisi sebuah bilangan T ($T \leq 1.000$) yang menyatakan banyaknya kasus yang harus ditangani. Untuk setiap kasus, berisi dua buah bilangan N dan M ($0 \leq N \leq 1.000$; $1 \leq M \leq 10.007$) yang menyatakan nilai pangkat yang diberikan ROKET-1 dan data maksimal yang dapat disimpan pada sistem PPABU secara berurutan.

Output

Untuk setiap kasus, cetak "Kasus #K: X" dimana K adalah nomor kasus dimulai dari 1, dan X adalah nilai dari $10^N \bmod M$ untuk kasus tersebut.

Sample Tests

Input

```
3
1 7
3 7
100 127
```

Output

```
Kasus #1: 3
Kasus #2: 6
Kasus #3: 122
```

Notes

Penjelasan untuk contoh input 1:

Data yang diberikan ROKET-1 adalah 10^1 dan data yang bisa disimpan harus dibawah 7. Sehingga $10 \bmod 7$ adalah 3.

Penjelasan untuk contoh input 2:

Data yang diberikan ROKET-1 adalah 10^3 dan data yang bisa disimpan harus dibawah 7. Oleh karena itu, $1000 \bmod 7 = 6$. Namun jika kita mengaplikasikan teori tersebut, kita bisa melakukan perhitungan sebagai berikut:

```
= 1000 mod 7
= ((10 * 10) * 10) mod 7
= (((10 * 10) mod 7) * (10 mod 7)) mod 7
= (((10 mod 7) * (10 mod 7)) mod 7) * (10 mod 7) mod 7
= ((3 * 3) mod 7) * 3 mod 7
= (9 mod 7) * 3 mod 7
= (2 * 3) mod 7
= 6 mod 7
= 6
```

Perhatian: Operasi modulus pada Pascal menggunakan operator 'mod' (tanpa tanda kutip), dan operasi modulus pada Java/C/C++ menggunakan operator '%' (tanpa tanda kutip).

Jangan menghitung langsung nilai dari 10^N . 10^{1000} merupakan angka yang sangat besar dengan 1000 digit yang tidak bisa ditangani tipe data longint/int64 pada Pascal atau int/long long pada Java/C/C++.