



116947 - Programação Competitiva - Turma A

Torneio 2

Prof. José Carlos Loureiro Ralha

1 Google

(Limite: 1 segundo – Arquivo: A.{c|cpp|java|py|py3|cs} – Autor: Matheus Pimenta)

As maiores empresas de *software* e tecnologia do mundo resolveram se juntar para fazer uma grande prova, que testa habilidades em solucionar problemas. Elas decidiram fazer isto, porque concordam que esta é a melhor maneira de selecionar novos empregados.

Google está desenvolvendo um novo *software*, cuja segurança irá depender da implementação de primitivas criptográficas. Para saber se você é capaz de participar da equipe deste projeto, a empresa te propôs o seguinte problema.

Um *corpo* (uma estrutura matemática com várias propriedades interessantes) é um conjunto e duas operações binárias (operações de dois operandos). Cada elemento do conjunto de um corpo possui um elemento *inverso*, em relação a cada uma das duas operações. Por exemplo, no corpo dos números reais com a soma e a multiplicação, o inverso multiplicativo de 5 é 0.2, pois $5 \cdot 0.2 = 1$ (1 é o elemento *neutro* da multiplicação). Isto quer dizer que, na verdade, um corpo tem quatro operações! Soma e sua inversa (subtração) e multiplicação e sua inversa (divisão). Subtrair x é somar o inverso aditivo de x . Dividir por x é multiplicar pelo inverso multiplicativo de x .

Em particular, existem corpos que são extensamente utilizados em criptografia. Estamos falando dos *corpos finitos*, também conhecidos como *corpos de Galois*. Se p é um número primo, então o conjunto $\{0, 1, 2, \dots, p-1\}$ e as operações modulares de soma e multiplicação formam um corpo finito.

Neste problema, dados os inteiros a , b e p , onde p é um número primo, você deve encontrar o valor de $a/b \bmod p$. Em outras palavras, dê o resto da divisão de a/b por p .

Se p é primo, então, pelo pequeno teorema de Fermat, o inverso multiplicativo de x módulo p é

$$x^{p-2} \bmod p$$

Lembre-se que $ab \bmod p$ é igual a $(a \bmod p)(b \bmod p) \bmod p$.

1.1 Entrada

A entrada é composta por vários casos de teste e termina com fim de arquivo.

Cada caso de teste é formado por uma única linha, com os inteiros a , b e p .

1.2 Saída

Para cada caso de teste, imprima uma linha com o valor de $a/b \bmod p$.

1.3 Restrições

- $1 \leq a, b, p \leq 2^{30}$
- p sempre é primo
- Cuidado com os tipos das suas variáveis!

1.4 Exemplos

Entrada	Saída
12 16 7	6
31 28 13	9

2 Facebook

(Limite: 0 segundos – Arquivo: B.{c|cpp|java|py|py3|cs} – Autor: Matheus Pimenta)

DP temporal

2.1 Entrada

A entrada é composta por vários casos de teste e termina com fim de arquivo.

2.2 Saída

Para cada caso de teste, imprima uma única linha com o número máximo de .

2.3 Restrições

- $1 \leq N \leq 25$

2.4 Exemplos

Entrada	Saída
4 6	5
0-0-0-0-0	6
0-0-0-*-0	
*-0-0-0-0	
--0-0-0	
0-*-0-0-0	
3 7	
0-*-0-0	
-0--0	
--0-0	
0-*-0-0	

3 Amazon

(Limite: 3 segundos – Arquivo: `C.{c|cpp|java|py|py3|cs}` – Autor: Matheus Pimenta)

Amazon oferece serviços de computação em nuvem. Acontece que muitos clientes agendaram a utilização dos recursos centrais para o dia de hoje. No entanto, Amazon possui a política de que só um cliente pode usar os recursos centrais de cada vez, para que as aplicações possam ter desempenho máximo. Então, a empresa pediu a sua ajuda para determinar a maior quantidade de agendamentos que podem ser atendidos. Nesta situação, Amazon quer deixar feliz a maior quantidade de clientes que puder.

3.1 Entrada

A entrada é composta por vários casos de teste e termina com fim de arquivo.

A primeira linha de um caso de teste é composta por um único inteiro N .

Em seguida virão N linhas. A i -ésima linha é composta por dois inteiros s_i e f_i , que indicam respectivamente os tempos de início e fim do i -ésimo agendamento.

3.2 Saída

Para cada caso de teste, imprima uma única linha com o maior número de agendamentos que podem ser atendidos no dia de hoje.

3.3 Restrições

- $1 \leq N \leq 10^5$
- $1 \leq s_i < f_i \leq 10^9$, para $1 \leq i \leq N$
- Um agendamento pode iniciar no mesmo instante em que outro termina.

3.4 Exemplos

Entrada	Saída
2	2
5 10 3 5 6 20 25 2 5 7 10 12 20 5 10 6 10	4

4 Sony

(Limite: 4 segundos – Arquivo: D. {c|cpp|java|py|py3|cs} – Autor: Matheus Pimenta)

A empresa Sony resolveu colaborar com o desenvolvimento de um jogo altamente promissor que será publicado para o PlayStation. Estamos falando, nada mais, nada menos do que do jogo de Dragon Ball Super, a nova série de Dragon Ball escrita pelo próprio Akira Toriyama! Para saber se você é capaz de participar da equipe, a empresa te desafiou a resolver o seguinte problema.

Goku, ao se transformar em Deus Super Saiyajin, terá uma habilidade especial ao disparar ataques de energia. A força do ataque aumenta, se ele atravessar uma sequência de inimigos tal que o poder de luta do próximo é maior que o do atual. Se o ataque atinge um inimigo com poder de luta menor que o do inimigo anterior, a energia é completamente dissipada e o ataque some. Além disso, se o ataque atinge um inimigo mais forte que o anterior, ele absorve a energia vital deste inimigo (que não tem relação alguma com o seu poder de luta). Dada uma sequência de inimigos dispostos em linha reta, as relações de poder de luta entre eles e seus pontos de energia vital, Sony quer saber se você é capaz de calcular a quantidade máxima de energia que um ataque pode absorver, permitindo retirar zero ou mais inimigos da sequência.

4.1 Entrada

A entrada é composta por vários casos de teste e termina com fim de arquivo.

A primeira linha de um caso de teste contém um único inteiro N , o número de inimigos, que são numerados de 1 a N .

Em seguida virão N linhas. A i -ésima linha contém dois inteiros x_i e v_i , que são respectivamente a coordenada unidimensional e os pontos de energia vital do inimigo i . Para qualquer par de inimigos distintos i e j , $x_i \neq x_j$.

As próximas $N - 1$ linhas dão as relações de poder de luta entre os inimigos. A i -ésima linha diz quem é o inimigo imediatamente mais fraco que $i + 1$. O inimigo 1 é sempre o mais fraco de todos. Observe que a relação é transitiva, ou seja, se i é mais fraco que j e j é mais fraco que k , então i é mais fraco que k . É garantido que a relação não é cíclica, ou seja, não haverá na entrada uma relação, por exemplo, em que i é mais fraco que j e j é mais fraco que i . Por fim, observe que se não é possível determinar a relação de poder de luta entre i e j , então não é possível que ambos i e j estejam na sequência ótima.

4.2 Saída

Para cada caso de teste, imprima uma única linha com a resposta.

4.3 Restrições

- $1 \leq x_i \leq N \leq 10^5$
- $1 \leq v_i \leq 10^4$

4.4 Exemplos

Entrada	Saída
1	7
1 7	15
3	
2 13	
1 3	
3 2	
3	
1	