**Problema A: Dan o Troco**

*TL: 1 segundo (~50 casos)*

*ML: 128 MB*

Dan é o dono do mercadinho da esquina. Ele não é muito bom de contas, e pediu para você fazer uma calculadora para calcular o troco das compras. As moedas existentes são a de 1, 5, 10, 20.

Dado um preço de **N** unidades, imprima as moedas de forma ordenada, da maior para a menor, de forma que o número de moedas seja o mínimo possível!

**Entrada**

A primeira linha de cada caso de teste contém um número **N** (1 <= **N** <= 10^3), indicando o preço da compra.

**Saída**

Imprima os valores das moedas separados por um sinal de +.

|  |  |
| --- | --- |
| **Entrada de Teste**  1  11  21  16  60 | **Saída de Teste**  1  10+1  20+1  10+5+1  20+20+20 |



**Problema B: Promoção**

*TL: 1 segundo (~50 casos)*

*ML: 128 MB*

Está tendo uma promoção muito interessante no Tiquinho (é um supermercado de doces e salgadinhos que tem na Avenida São Carlos!). A promoção é a seguinte: há **N** itens no seu carrinho, cada um com um preço (podemos ter preços iguais). Ao passar os itens no caixa você deve escolher **3** itens do seu carrinho e pagar só pelos **2** itens mais caros, pegar os próximos **3** e pagar só os **2** mais caros e assim por diante até que sobre **menos** de **3** itens, caso isso aconteça você deve pagar pelo preço dos itens restantes.

O que os donos não perceberam, é que dependendo da ordem em que se é passado os itens, podemos obter um preço maior ou menor. Você é capaz de dizer qual o menor preço que você pode pagar se escolher a melhor estratégia?

**Entrada**

A primeira linha de cada caso de teste contém um número **N** (1 <= **N** <= 10^4), indicando o número de items no seu carrinho. Na linha seguinte há **N** números inteiros (de 1 até 10^5) separados por espaços, indicando o preço de cada item.

**Saída**

Imprima o menor preço possível.

|  |  |
| --- | --- |
| **Entrada de Teste**  1  3  2  4 3  3  5 4 3  6  1 1 2 1 2 3  7  4 8 1 3 5 6 10 | **Saída de Teste**  3  7  9  7  28 |

No penúltimo caso, a estratégia ótima é de se pegar os conjuntos: (2 2 3) e (1 1 1).

**Problema C: Números Parecidos**

*TL: 1 segundo (~1000 casos)*

*ML: 128 MB*

Dizemos que dois números são parecidos se eles são diferentes por apenas um dígito na notação decimal, com possíveis zeros na frente (ex: 000123). Por exemplo, os números 128 e 028 são parecidos, pois diferem de apenas um dígito. Mas considere outro exemplo, 2047 e 0040 diferem por dois dígitos, e não são considerados parecidos!!

Dado um número **N**, ache o menor número maior ou igual a zero que seja parecido com ele! Imprima-o sem os zeros na frente!

**Entrada**

Há vários testes. Cada caso de teste contém um número **N** (1 <= **N** <= 10^9), indicando o número do enunciado. Há por volta de 1000 casos de teste!

**Saída**

Para cada caso de teste, imprima o menor número que é parecido com **N**.

|  |  |
| --- | --- |
| **Entrada de Teste**  128  2047  30000  9  123 | **Saída de Teste**  28  47  0  0  23 |



**Problema D: Poluição Auditiva**

*TL: 3 segundos (~50 casos)*

*ML: 128 MB*

No cotidiano sofremos com os altos níveis de ruídos vindo de carros, construções, entre outros. Nos devemos preocupar com esta poluição sonora, pois ela pode causar danos permanentes à nossa audição.

Por este motivo você deseja saber qual **o maior nível de ruído** (e não a soma!!) que você deve enfrentar ao ir de um destino à outro qualquer da cidade se você traçar a melhor rota possível, ou seja, se você somente andar por ruas com pouco ruído (isso para todos os pares de destinos!!).

Você tem em mãos o mapa da cidade, com as ruas (arestas) e intersecções (vértices) e cada rua tem um nível de ruído. Salve sua audição! É garantido que há caminhos entre todos os pontos da cidade.

**Entrada**

Há vários testes. A primeira linha de cada caso de teste contém os números **N** (1 <= **N** <= 10^3) e **M** (1 <= M <= 10^5), indicando o número de interseccções/destinos. Nas próximas **M** linhas haverá três números inteiros **a**, **b** (1 <= **a,b** <= **N**) e **c** (1 <= **c** <= 10^9), indicando que os destinos **a** e **b** estão conectados por uma rua de ruído **c**.

**Saída**

Para cada caso de teste, imprima o maior valor

|  |  |
| --- | --- |
| **Entrada de Teste**  4 4  1 2 1  2 3 2  1 4 5  3 4 1  3 3  1 2 10  2 3 5  1 1 1 | **Saída de Teste**  2  10 |

**Problema E: Dia de Atividades**

*TL: 2 segundos (~100 casos)*

*ML: 128 MB*

Se organizar durante um dia com tantas atividades não é fácil. Como um ser preguiçoso, você quer ao mesmo tempo ocupar o dia inteiro com tarefas e executar o mínimo de tarefas possível.

Cada dia começa no tempo **0** e vai até o tempo **M**. Dado o intervalo que cada tarefa ocupa, organize seu dia de modo que elas cubram o dia inteiro e ao mesmo tempo o número de tarefas seja o mínimo possível. Como você caiu em um barril radioativo quando criança e agora tem oito braços, você consegue fazer quantas tarefas quiser ao mesmo tempo (intersecção de tarefas!). É garantido que exista um conjunto de tarefas que cubra o dia inteiro.

**Entrada**

Há vários testes. A primeira linha de cada caso de teste contém os números **N** (1 <= **N** <= 10^4) e **M** (1 <= M <= 10^9), indicando o número de tarefas existentes e a duraçao do dia. Nas próximas **N** linhas haverá dois números inteiros **a**, **b** (0 <= **a** <= **b** <= **M**), indicando que existe uma tarefa que começa em **a** e termina em **b**, podendo inclusive começar e terminar no mesmo tempo.

**Saída**

Para cada caso imprima a quantidade mínima de tarefas para cobrir o dia inteiro.

|  |  |
| --- | --- |
| **Entrada de Teste**  2 4  0 3  2 4  3 4  0 3  2 4  1 3  2 5  0 5  2 3 | **Saída de Teste**  2  2  1 |



**Problema F: Jogo das Predas**

*TL: 1 segundo (~1000 casos)*

*ML: 128 MB*

Chaves e Kiko estão jogando o Jogo das Predas. O jogo consiste em uma pilha de **N** pedras. Os dois jogam no mesmo monte de pedras. A cada turno um deles deve escolher um número de pedras **x** qualquer (**x** deve ser menor ou igual ao número de pedras do monte e maior do que zero) e remover estas pedras da pilha.

Eles se alternam no jogo, ou seja, cada hora é o turno de alguém. Quando um jogador decide tirar **x** pedras do monte, então em nenhum outro turno nenhum outro jogador poderá tirar **x** pedras, ou seja, o movimento de se tirar **x** pedras agora é inválido.

Chaves joga primeiro, e no primeiro turno ele não pode tirar todas as pedras para ser justo com Kiko. O jogador que não pode fazer uma jogada perde. Dado o tamanho da pilha de pedras, quem vence caso os dois sempre joguem tentando ao máximo ganhar o jogo?

Considere o caso onde temos 2 pedras. Uma das estratégias ótimas é se Chaves tirar 1 pedra (no caso é a única jogada possível dele), deixando o monte com 1 pedra. Agora Kiko não tem jogadas e portanto perde e Chaves ganha!

Considere o caso onde temos 3 pedras. Não importa qual seja a jogada de Chaves, Kiko sempre vai ganhar. Se Chaves retirar 2 pedras então Kiko retira 1 pedra e o monte fica vazio, e Chaves não tem nenhuma jogada válida. Se Chaves retirar 1 pedra então Kiko retira 2 pedras e o monte fica vazio, e mais uma vez Chaves perde.

**Entrada**

A primeira linha de cada caso de teste contém o números **N** (1 <= **N** <= 10^9), indicando o número de pedras no monte. Temos por volta de 1000 casos de teste!

**Saída**

Imprima **Isso Isso Isso** caso Chaves vença, ou **Kiko Kiko RaRaRa** caso Kiko vença.

|  |  |
| --- | --- |
| **Entrada de Teste**  1  2  3  8 | **Saída de Teste**  Kiko Kiko RaRaRa  Isso Isso Isso  Kiko Kiko RaRaRa  Isso Isso Isso |