**Problema A: Primos**

*TL: 1 segundo (~1000 casos)*

*ML: 128 MB*

Números primos são usados em várias aplicações na computação e na matemática e ainda são objetos de muito estudo e pesquisa. Um número é considerado primo se ele for apenas divisível por 1 e por ele mesmo. Seu objetivo aqui é dado um número **Y**, imprimir o primeiro primo maior ou igual a **Y**.

Alguns exemplos de primo: 2, 3, 5, 7, 11…

Alguns exemplos de não primos: 4, 10, 20, 49...

**Entrada**

Temos por volta de 1000 casos de teste. Cada caso contém um número **Y** (1 <= **Y** <= 10^4), indicando o número de teste.

**Saída**

Imprima o primeiro primo maior ou igual a **Y**.

|  |  |
| --- | --- |
| **Entrada de Teste**  1  8  123 | **Saída de Teste**  2  11  127 |

**Problema B: Anagramas**

*TL: 1 segundo (~50 casos)*

*ML: 128 MB*

Um anagrama de uma palavra é uma outra palavra que pode ser obtida da primeira ao reordenarmos as letras dela, sem remover ou adicionar nenhuma outra. Por exemplo, “aabb” é um anagrama de “baba”, enquanto “baaa” **não** é anagrama de “bbaa” (uma palavra tem um ‘b’ a mais, e a outra tem um ‘a’ a mais).

Dado algumas palavras, calcule o tamanho do maior conjunto de palavras onde nenhum par delas é um anagrama da outra!

**Entrada**

A primeira linha de cada caso de teste contém um número **N** (1 <= **N** <= 10^3), indicando a quantidade de palavras. Nas próximas **N** linhas haverá palavras com letras do alfabeto minúsculas de tamanho máximo 20.

**Saída**

Imprima o tamanho do maior conjunto.

|  |  |
| --- | --- |
| **Entrada de Teste**  3  aabb  bbaa  abbb  3  a  b  c  3  aaa  aa  a  3  abc  cba  bac | **Saída de Teste**  2  3  3  1 |

**Problema C: Brock o Aviador**

*TL: 1 segundo (~50 casos)*

*ML: 128 MB*

Brock tem seu próprio avião. Inicialmente ele tem **F** de combustível disponível. Ele vai fazer vários voos consecutivos. Cada voo gasta uma certa quantidade de combustível, e só pode ser feito caso a quantidade de combustível seja suficiente. **Antes** de decolar ele poderá reabastecer o seu avião com uma quantidade de combustível que depende do voo que ele escolheu.

Dado **F** e as informações de cada voo (quantidade necessária de combustível e quantidade de combustível que iremos reabastecer), qual o número máximo de voos que Brock pode fazer? Ele só pode fazer cada voo uma única vez e a quantidade que podemos reabastecer sempre é menor ou igual à quantidade que gastamos no voo.

Por exemplo, suponha que **F** = 5 e temos 3 voos: (10, 8), (4, 1) e (5, 3). A resposta será 2, pois podemos pegar o primeiro voo e reabastecer com 8 unidades, indo para **F** = 13 de combustível. Ao completar o primeiro voo teremos **F** = 3 de combustível. Depois pegamos o segundo, iremos reabastecer com 1 unidade, indo para **F** = 4. Ao completar o segundo teremos **F** = 0 de combustível, não tendo como executar um terceiro voo. Lembre que podemos escolher os voos em qualquer ordem!!

**Entrada**

A primeira linha de cada caso de teste contém os números **N** (1 <= **N** <= 10^4) e **F** (1 <= **F** <= 10^9), indicando o número de voos disponíveis e o combustível. Na linha seguinte, há **N** valores separados por espaços (de 1 a 10^9), que indicam a quantidade necessária para executar cada voo. Na outra linha há mais **N** valores correspondentes à quantidade a ser reabastecida em cada voo.

**Saída**

Imprima o número máximo de voos possíveis.

|  |  |
| --- | --- |
| **Entrada de Teste**  2 2  6 2  1 1  4 5  6 3 3 1  1 2 2 1  6 5  6 3 2 5 4 2  1 2 2 1 2 2 | **Saída de Teste**  1  3  4 |

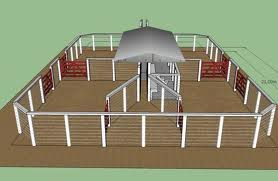
Nota: No último exemplo, uma possível resposta seria as viagens (numerando a partir do 1): 2,3,5,6

**Problema D: Segura Pião**

*TL: 2 segundos (~50 casos)*

*ML: 128 MB*

Você é o boi Zebu. Você fica preso em um curral. Este curral é separado em várias partes, e podemos ter uma porteira que conecta duas partes, e ela só se abre depois de um tempo determinado (automaticamente). Você está na parte identificada pelo número **1**.



Supondo que você não gasta tempo indo de uma parte à outra, quando será possível chegar até a parte **N**?

**Entrada**

A primeira linha de cada caso de teste contém os números **N** (1 <= **N** <= 10^4) e **M** (1 <= **M** <= 10^5), indicando o número de partes do curral e o número de porteiras respectivamente. Nas próximas **M** linhas haverá três números inteiros **a**, **b** (1 <= **a,b** <= N) e **c** (1 <= **c** <= 10^9), indicando que as partes **a** e **b** estão conectadas por uma porteira que só se abre no tempo **c**. Uma porteira pode magicamente conectar a mesma parte, e podemos ter mais de uma porteira que conecte duas partes.

**Saída**

Imprima quando será possível chegar até a parte **N**.

|  |  |
| --- | --- |
| **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: Carlota Fã da Capricho**

*TL: 1 segundo (~100 casos)*

*ML: 128 MB*

Carlota gosta muito de ler revistas da Capricho. Ele não tem tempo para bobagens, mas hoje está com algum tempo para ler sua revista favorita. Ele possui várias revistas da Capricho em mãos. Elas estão ordenadas em uma certa sequência e ele gasta um certo tempo na leitura de cada revista.

Carlota decide que irá começar a ler uma das revistas e continuará a lê-las na sequência dada até que o final da sequência seja alcançada ou o seu tempo livre acabe. Ou seja, se ele começar a ler a revista 5, então a próxima a ser lida deve ser a 6, depois a 7 e assim por diante.

Você consegue dizer quantas revistas da Capricho Carlota consegue ler no máximo caso considerarmos todos os inícios da leitura? Ele não começa a ler a próxima revista se ele não tiver tempo suficiente para terminá-la.

**Entrada**

A primeira linha de cada caso de teste contém um número **N** (1 <= **N** <= 10^4) e um número **T** (1 <= **T** <= 10^9), indicando o número de revistas e o tempo livre. Na linha seguinte, há **N** valores (1 <= valor <= 10^4) que correspondem ao tempo gasto na leitura de cada revista.

**Saída**

Imprima o número máximo de revistas que Carlota consegue ler.

|  |  |
| --- | --- |
| **Entrada de Teste**  4 5  3 1 2 1  3 3  2 2 3  1 3 5  10 15 10 9 1 1 5 10 5 3 7 2 | **Saída de Teste**  3  1  0  3 |

**Problema F: Fossa e os Primos**

*TL: 1 segundo (~50 casos)*

*ML: 128 MB*

Fossa possui um intervalo de números inteiros, representado por [**A**,**B**], isto é, temos os números **A**, **A+1**, **A+2**, **…**, **B**. Ele quer escolher um **L** qualquer, e criar todos os intervalos válidos da forma **[A+K, A+K+L-1]**, onde **A+K+L-1** **<= B**. Como Fossa é louco, ele quer que nestes intervalos, exista pelo menos **K** números primos nele. Você consegue encontrar o mínimo valor de **L** caso ele exista? (Veja o exemplo de teste explicado em detalhes para entender melhor).

**Entrada**

A linha de cada caso de teste contém os números **A**, **B** e **K** (1 <= **A**,**B**,**K** <= 5\*10^4, **A** <= **B**).

**Saída**

Imprima o valor **L** (1 <= **L** <= **B-A+1**) ou **-1** se ele não existir.

|  |  |
| --- | --- |
| **Entrada de Teste**  6 13 1  1 4 3 | **Saída de Teste**  4  -1 |

A resposta no primeiro caso de teste é L = 4. Por quê? Porque sim Zequinha: o novo intervalo será [6, 13-4+1] = [6, 10]. Assim, os intervalos criados serão: [6, 6+4], [7, 7+4], [8, 8+4], [9, 9+4], [10, 10+4] = [6, 10], [7, 11], [8, 12], [9, 13], [10, 14]. Todos eles possuem pelo menos um número primo! Tente fazer para L = 1, 2 ou 3 e verá que os intervalos criados não atenderão à vontade de Fossa.