**Desafio**

Como um bom desenvolvedor, você precisa fazer um programa que leia dois inteiros: R e V (devem ser lidos tantos valores para V quantos necessários, até que seja digitado um valor maior do que R para ele). Conte quantos números inteiros devem ser somados em sequência (considerando o R nesta soma) para que a soma ultrapasse a V o mínimo possível. Escreva o valor final da contagem.  
  
A entrada pode conter, por exemplo, os valores 21 21 15 30. Neste caso, é então assumido o valor 21 para R enquanto os valores 21 e 15 devem ser desconsiderados pois são menores ou iguais a R. Como o valor 30 está dentro da especificação (maior do que R) ele será válido e então deve-se processar os cálculos para apresentar na saída o valor 2, pois é a quantidade de valores somados para se produzir um valor maior do que 30 (21 + 22).

**Entrada**

A entrada contém somente valores inteiros, um por linha, podendo ser positivos ou negativos. O primeiro valor da entrada será o valor de R. A próxima linha da entrada irá conter V. Se V não atender a especificação do problema, ele deverá ser lido novamente, tantas vezes quantas forem necessárias.

**Saída**

Imprima uma linha com um número inteiro que representa a quantidade de números inteiros que devem ser somadas, de acordo com a especificação acima.

| Exemplo de Entrada | Exemplo de Saída |
| --- | --- |
| 3 1 20 | 5 |

R = 3

~~V =1~~ (<R) , V = 20 (>R)

R + (R+1)+ (R+2)+... > V

3+4 = 7 < 20 cont=2

3+4+5 = 12 < 20 cont=3

3+4+5+6 = 18 < 20 cont=4

3+4+5+6+7 = 25 > 20 cont=5

cont = 5

Todas as entradas e saída dos algoritmos são utilizados o STDIN e STDOUT de cada linguagem, abaixo tem algumas dicas de como utilizar cada STDIN e STDOUT de cada linguagem.

**JavaScript**

Em JavaScript as funções de STDIN e STDOUT respectivamente são**gets** e **console.log**, a função gets é implementada internamente para auxiliar a entrada dos dados.

**Exemplo:**

let line = gets(); // Retorna a próxima linha de entrada

console.log(line); // Imprime o dado

**Java**

Em Java existe várias formas de implementar o STDIN e STDOUT recomendamos utilizar **BufferedReader** para o STDIN e o **System.out.println** para o STDOUT.

**Exemplo:**

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

StringTokenizer st = new StringTokenizer(br.readLine()); // Lê a linha de entrada

int a = Integer.parseInt(st.nextToken());

System.out.println(a); // Imprime o dado

**Python**

Em Python existe várias formas de implementar o STDIN e STDOUT recomendamos utilizar **sys.stdin.readline** para o STDIN e o **print** para o STDOUT.

**Exemplo:**

import sys

a = int(sys.stdin.readline()) // Lê a linha de entrada

print(a); // Imprime o dado

27

// a função gets é implementada dentro do sistema para ler as entradas(inputs) dos dados e a função print para imprimir a saída (output) de dados.

// Abaixo segue um exemplo de código que você pode ou não utilizar

*let* R = parseInt(gets());

*let* V = 0;

// Complete os espaços em branco com uma possível solução para o problema

while(true){

V = parseInt(gets())

if( ){

break

}

}

*let* cont = 1;

*let* it = R + 1;

while(true){

cont++

it++

if( ){

print( )

break;

}

}

**TESTES**

**EXECUTAR TESTES**

###### Teste #1



###### Teste #2



###### Teste #3



###### Teste #4