**Desafio**

Você terá o desafio de escrever um programa que leia um valor inteiro N (1 < N < 1000). Este N é a quantidade de linhas de saída que serão apresentadas na execução do programa.

**Entrada**

O arquivo de entrada contém um número inteiro positivo N.

**Saída**

Imprima a saída conforme o exemplo fornecido.

| Exemplo de Entrada | Exemplo de Saída |
| --- | --- |
| 5 | 1 1 1 2 4 8 3 9 27 4 16 64 5 25 125 |

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

// 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 linhas = parseInt(gets());

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

let limit = parseInt(gets());

for (let i = 1; i <= limit; i++) {

let X =( );

let Y = ( );

print( );

}