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(); // Lê a 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

**Desafio**

Vamos agora, adicionar uma funcionalidade à classe UsuarioTelefone, que realizar chamadas para outros usuários. Cada chamada terá uma duração em minutos e o custo será deduzido do saldo do usuário, suponha o custo de $0.10 por minuto. Você pode criar um método fazer\_chamada que vai permitir que o usuário faça a chamada, ele vai receber o destinatario e duracao como parâmetros. Calcule o custo da chamada usando o método 'custo\_chamada' do objeto 'plano', além de adicionar o método deduzir\_saldo para deduzir o valor do saldo do plano e depois retorne uma mensagem adequada como mostra no exemplo a baixo.

**Entrada**

Número do usuário, número do telefone, saldo inicial, número do destinatário e a duração da chamada em minutos.

**Saída**

Mensagem indicando o sucesso da chamada ou saldo insuficiente para fazer a chamada.

**Exemplos**

A tabela abaixo apresenta exemplos com alguns dados de entrada e suas respectivas saídas esperadas. Certifique-se de testar seu programa com esses exemplos e com outros casos possíveis.

| **Entrada** | **Saída** |
| --- | --- |
| Rodrigo (00) 90000-0000 10.00 (33) 93333-3333 60 | Chamada para (33) 93333-3333 realizada com sucesso. Saldo: $4.00 |
| Yule (11) 91111-1111 30.00 (00) 90000-0000 240 | Chamada para (00) 90000-0000 realizada com sucesso. Saldo: $6.00 |
| Amelia (33) 93333-3333 10.00 (11) 91111-1111 120 | Saldo insuficiente para fazer a chamada. |

# Classe UsuarioTelefone e o encapsulamento dos atributos nome, numero e plano:

class UsuarioTelefone:

def \_\_init\_\_(self, nome, numero, plano):

self.nome = nome

self.numero = numero

self.plano = plano

# TODO: Crie um método fazer\_chamada para permitir que um usuário faça uma chamada telefônica:

# TODO: Calcule o custo da chamada usando o método 'custo\_chamada' do objeto 'plano':

# TODO: Verifique se o saldo do plano é suficiente para a chamada.

# TODO: Se o saldo for suficiente, deduz o custo da chamada do saldo do plano.

# TODO: E retorne uma mensagem de sucesso com o destinatário e o saldo restante após a chamada:

# Classe Pano, ela representa o plano de um usuário de telefone:

class Plano:

def \_\_init\_\_(self, saldo\_inicial):

self.saldo = saldo\_inicial

# TODO: Crie um método para verificar\_saldo e retorne o saldo atual:

# TODO: Crie um método custo\_chamada para calcular o custo de uma chamada supondo o custo de $0.10 por minuto:

# TODO: Crie um método deduzir\_saldo para deduz o valor do saldo do plano:

# Classe UsuarioPrePago, aqui vemos a herança onde UsuarioPrePago herda os atributos e métodos da classe UsuarioTelefone:

class UsuarioPrePago(UsuarioTelefone):

def \_\_init\_\_(self, nome, numero, saldo\_inicial):

super().\_\_init\_\_(nome, numero, Plano(saldo\_inicial))

# Recebendo as informações do usuário:

nome = input()

numero = input()

saldo\_inicial = float(input())

# Objeto de UsuarioPrePago com os dados fornecidos:

usuario\_pre\_pago = UsuarioPrePago(nome, numero, saldo\_inicial)

destinatario = input()

duracao = int(input())

# Chama o método fazer\_chamada do objeto usuario\_pre\_pago e imprime o resultado:

print(usuario\_pre\_pago.fazer\_chamada(destinatario, duracao))