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**

Agora, vamos Adicionar uma funcionalidade à classe UsuarioTelefone para que possa ser verificado o saldo disponível em seu plano. Para essa solução, você pode criar uma classe PlanoTelefone, o seu método de inicialização e encapsular os atributos, 'nome' e 'saldo' dentro da classe. Adicione também um método 'verificar\_saldo' para verificar o saldo do plano e uma  'mensagem\_personalizada' para gerar uma mensagem personalizada.

**Condições da verificação do saldo:**  
- Caso o saldo seja menor do que 10, retorne: "Seu saldo está baixo. Recarregue e use os serviços do seu plano."  
- Caso o saldo seja maior ou igual a 50, retorne: "Parabéns! Continue aproveitando seu plano sem preocupações."  
- Caso contrário, retorne: "Seu saldo está razoável. Aproveite o uso moderado do seu plano."

**Entrada**

Como entrada, será solicitado o nome, plano (Essencial, Prata, Premium) e saldo atual do cliente.

**Saída**

Mensagem personalizada de acordo o saldo do cliente.

**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** |
| --- | --- |
| João      Essencial       9 | Seu saldo está baixo. Recarregue e use os serviços do seu plano. |
| Debora Prata 11 | Seu saldo está razoável. Aproveite o uso moderado do seu plano. |
| Catarina Premium 50 | Parabéns! Continue aproveitando seu plano sem preocupações. |

# TODO: Crie a classe PlanoTelefone, seu método de inicialização e encapsule os atributos, 'nome' e 'saldo':

# TODO: Crie um método 'verificar\_saldo' para verificar o saldo do plano sem acessar diretamente o atributo:

# TODO: Crie um método 'mensagem\_personalizada' para gerar uma mensagem personalizada com base no saldo:

# Classe UsuarioTelefone:

class UsuarioTelefone:

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

self.nome = nome

self.plano = plano

# TODO: Crie um método para verificar o saldo do usuário e gerar uma mensagem personalizada:

# Recebendo as entradas do usuário (nome, plano, saldo):

nome\_usuario = input()

nome\_plano = input()

saldo\_inicial = float(input())

# Criação de objetos do plano de telefone e usuário de telefone com dados fornecidos:

plano\_usuario = PlanoTelefone(nome\_plano, saldo\_inicial)

usuario = UsuarioTelefone(nome\_usuario, plano\_usuario)

# Chamada do método para verificar\_saldo sem acessar diretamente os atributos do plano:

saldo\_usuario, mensagem\_usuario = usuario.verificar\_saldo()

print(mensagem\_usuario)