

Desafios Inicias Java - GFT Start Woman

1 / 4 - Multiplicação Simples

Você receberá dois valores inteiros. Faça a leitura e em seguida calcule o produto entre estes dois valores. Atribua esta operação à variável PROD, mostrando esta de acordo com a mensagem de saída esperada (exemplo abaixo).

Entrada

A entrada contém 2 valores inteiros.

Saída

Exiba a variável PROD conforme exemplo abaixo, tendo obrigatoriamente um espaço em branco antes e depois da igualdade.

Exemplos de Entrada

3
9

Exemplos de Saída

PROD = 27

-30
10

PROD = -300

0
9

PROD = 0

Código

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int a, b, PROD;

        a = sc.nextInt();
        b = sc.nextInt();

        PROD = a * b; //implemente o código que representa a
multiplicação.

        System.out.println("PROD = " + PROD);

        sc.close();
    }
}

---
```

2 / 4 - Coxinha de Bueno

Em 2015 um novo record foi alcançado na competição de Coxinhas de Bueno de Andrada, onde Mônica mandou pra dentro 43 coxinhas em apenas 10 minutos, passando se antecessor que conseguiu comer, no mesmo tempo, 38 coxinhas em 2014.

O restaurante especializado em coxinhas do pequeno distrito de Bueno de Andrada, interior de São Paulo, organiza essa competição todos os anos, mas nunca conseguiram entrar para o livro dos recordes, o Guinness Book. Para isso, o restaurante precisa preencher informações sobre a competição, como o número de coxinhas consumidas pelos competidores durante o evento.

Porém, como já foi informado, a especialidade deles é coxinha, não matemática, então será que você pode ajudá-los? Com base no número total de coxinhas consumidas e o número de participantes na competição, o dono do restaurante precisa que você desenvolva um programa para saber a quantidade média de coxinha que os participantes da competição conseguem devorar.

Ah, lembre que, em troca da sua ajuda, você poderá comer quantas coxinhas conseguir.

Entrada

A entrada consiste de uma única linha que contém dois inteiros H e P ($1 \leq H, P \leq 1000$) indicando respectivamente o número total de coxinhas consumidas e o número total de participantes na competição.

Saída

Seu programa deve produzir uma única linha com um número racional representando o número médio de coxinhas consumidas pelos participantes. O resultado deve ser escrito como um número racional com exatamente dois dígitos após o ponto decimal, arredondado se necessário.

Exemplos de Entrada	Exemplos de Saída
10 90	0.11
840 11	76.36
1 50	0.02

código original

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

```
import java.util.Scanner;
```

```
public class Main {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String input = scan.nextLine();

        String[] inputSplit = input.split(" ");
        //TODO: Complete os espaços em branco com uma possível solução para o desafio
        double media = Double.parseDouble(
            )/Double.parseDouble();

        System.out.printf("%.2f",    );
    }
}
```

solução encontrada sem inputSplit

```
// Abaixo segue um exemplo de código que você pode ou não utilizar
import java.util.*;
import java.lang.Math;
```

```
public class CoxinhaBueno {
    private static Scanner input = new Scanner(System.in);

    public static void main(String[] args) {
        double h, p, media;

        h = input.nextDouble();
        p = input.nextDouble();

        media = h / p;

        System.out.printf("%.2f", media);

        scan.close();
    }
}
```

solução encontrada com inputSplit

```
// Abaixo segue um exemplo de código que você pode ou não utilizar
import java.util.Scanner;
```

```
public class Main {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String input = scan.nextLine();

        String[] inputSplit = input.split(" ");

        double H = Double.parseDouble(inputSplit[0]);
        double P = Double.parseDouble(inputSplit[1]);

        double media = (double) H / P;

        System.out.printf("%.2f", media);

        scan.close();
    }
}
```

3 / 4 - Álbum da Copa

A Copa do Mundo de Futebol é um dos maiores eventos esportivos do Planeta Terra, e o álbum de figurinhas oficial é sempre um grande sucesso entre os amantes dessa competição. O álbum contém espaços numerados de 1 a N para colar as figurinhas; cada figurinha, também numerada de 1 a N, é uma pequena foto de um jogador de uma das seleções que jogará a Copa do Mundo. O objetivo é colar todas as figurinhas nos respectivos espaços no álbum, de modo a completar o álbum, ou seja, não deixar nenhum espaço sem a correspondente figurinha.

As figurinhas são vendidas em envelopes fechados, de forma que o comprador não sabe quais figurinhas está comprando, e pode ocorrer de comprar uma figurinha que ele já tenha colado no álbum.

Para ajudar os usuários, a empresa responsável pela venda do álbum e das figurinhas quer criar um aplicativo que permita gerenciar facilmente as figurinhas que faltam para completar o álbum e está solicitando a sua ajuda.

Dados o número total de espaços e figurinhas do álbum, e uma lista das figurinhas já compradas (que pode conter figurinhas repetidas), o seu desafio é determinar quantas figurinhas faltam para completar o álbum.

Entrada

A primeira linha contém um inteiro N ($1 \leq N \leq 100$) indicando o número total de figurinhas e espaços no álbum. A segunda linha contém um inteiro M ($1 \leq M \leq 300$) indicando o número de figurinhas já compradas. Cada uma das M linhas seguintes contém um número inteiro X ($1 \leq X \leq N$) indicando uma figurinha já comprada.

Saída

Seu programa deve produzir uma única linha contendo um inteiro representando o número de figurinhas que falta para completar o álbum.

Exemplos de Entrada	Exemplos de Saída
---------------------	-------------------

10	7
3	
5	
8	
3	
3	0
4	
2	
1	
3	
3	

##código original

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

```
import java.util.HashSet;
import java.util.Scanner;
import java.util.Set;

public class Main {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int numTotal = scan.nextInt();
        int numFigCompradas = scan.nextInt();
        Set<Integer> setFig = new HashSet<>();
        //TODO: Complete os espaços em branco com uma solução possível para o problema
        for (
            ) {

        }
    }
}
```

```

        System.out.println(numTotal - setFig.size());
    }
}

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

import java.util.HashSet;
import java.util.Scanner;
import java.util.Set;

public class CompletaAlbum {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);
        int numTotal = scan.nextInt();
        int numFigCompradas = scan.nextInt();
        Set<Integer> setFig = new HashSet<>(numFigCompradas);

        int count = 0;
        for (int i = 0; i < numFigCompradas; i++) {

            setFig.add(scan.nextInt());
            count++;
        }

        System.out.println(numTotal - setFig.size());

        scan.close();
    }
}

---
```

4 / 4 - Degustação de vinho

Degustação de vinho às escuras é a habilidade de identificar um vinho usando apenas seus sentidos do olfato e paladar.

Durante uma competição de degustação, uma garrafa de vinho é aberta e dividida em taças para que os cinco competidores possam provar. Eles podem cheiras, saborear e avaliar a bebida para conseguir identificar qual o tipo do vinho, sendo: (1) Cabernet; (2) Merlot; (3) Pinot Noir. No final, as respostas são verificadas para determinar o número de suposições corretas.

Dado o qual foi o tipo do vinho e as respostas fornecidas, determine o número de participantes que receberam a resposta correta.

Entrada

A primeira linha contém um inteiro T representando o tipo de vinho ($1 \leq T \leq 4$). A segunda linha contém cinco inteiros A, B, C, D e E , que indica a resposta dada por cada competidor ($1 \leq A, B, C, D, E \leq 4$).

Saída

A saída contém um inteiro representando o número de concorrentes que obtiveram a resposta correta.

Exemplos de Entrada	Exemplos de Saída
1	2
1 2 3 2 1	
3	0
4 1 1 2 1	

##codigo original

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

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        String T = scan.nextLine();
        String respostasCompetidores = scan.nextLine();

        String[] respostasCompetidoresSplit = respostasCompetidores.split(" ");
        //TODO: Complete o código com uma possível solução para o problema.
        int count = 0;
        for (                ) {
            if (                )
                count++;
        }

        System.out.println(                );
    }
}
```

##solução

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

```
import java.util.Scanner;

public class DegustacaoCorreta {
    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);
        String T = scan.nextLine();
        String respCompetidores = scan.nextLine();
        String[] respCompetidoresSplit = respCompetidores.split(" ");
        //TODO: Complete o código com uma possível solução para o problema.
        int count = 0;
        for (int i = 0; i < respCompetidoresSplit.length; i++) {
```

```
        if (respCompetidoresSplit[i].equals(T)) {  
            count++;  
        }  
    }  
  
    System.out.println(count);  
  
    scan.close();  
}  
}
```

Link vídeo de entrega do desafio de código inicial
<https://screencast-o-matic.com/watch/c3hqh1VrxVx>