**1912 – HELP SEU MADRUGA**

1. import java.util.Scanner; // import Scanner class

2.

3. public class HelpSeuMadruga { // HelpSeuMadruga class declaration

4.     public static void main(String[] args) { // main method declaration

5.         Scanner scanner = new Scanner(System.in); // instantiate Scanner object

6.         while (scanner.hasNextInt()) { // loop through input

7.             int numberStrip = scanner.nextInt(); // read number of strips

8.             int desiredHeight = scanner.nextInt(); // read desired height

9.             if (numberStrip + desiredHeight == 0) { // if it is the last test case

10.                 break; // break loop

11.             } // end of if

12.

13.             int[] stripHeights = new int[numberStrip]; // instantiate strip heights array

14.             int totalHeight = 0; // instantiate total height

15.             for (int i = 0; i < numberStrip; i++) {

16.                 stripHeights[i] = scanner.nextInt(); // read strip height

17.                 totalHeight += stripHeights[i]; // add strip height to total height

18.             } // end of loop through strip heights

19.

20.             if (totalHeight < desiredHeight) { // if it is impossible

21.                 System.out.println("-.-"); // print result

22.                 continue; // continue loop

23.             } else if (totalHeight == desiredHeight) { // if it is not necessary to cut

24.                 System.out.println(":D"); // print result

25.                 continue; // continue loop

26.             } // end of if

27.

28.             boolean isPossible = false; // instantiate is possible

29.             double left = 0.0; // instantiate left

30.             double right = 100001.0; // instantiate right

31.             double currentHeight = (left + right) / 2.0; // instantiate current height

32.             double error = 0.0001; // instantiate error

33.

34.             while (right - left > 0.000001) { // loop through binary search

35.                 double currentTotal = 0.0; // instantiate current total

36.                 for (int i = 0; i < numberStrip; i++) { // loop through strip heights

37.                     if (stripHeights[i] > currentHeight) { // if it is necessary to cut

38.                         currentTotal += (stripHeights[i] - currentHeight); // add cut height to current total

39.                     } // end of if

40.                 } // end of loop through strip heights

41.

42.                 if (Math.abs(currentTotal - desiredHeight) < error) { // if it is possible

43.                     System.out.printf("%.4f\n", currentHeight); // print result

44.                     isPossible = true; // change is possible

45.                     break; // break loop

46.                 } // end of if

47.

48.                 if (currentTotal < desiredHeight) { // if it is possible

49.                     right = currentHeight; // change right

50.                     currentHeight = (right + left) / 2.0; // change current height

51.                 } // end of if

52.

53.                 if (currentTotal > desiredHeight) { // if it is not possible

54.                     left = currentHeight; // change left

55.                     currentHeight = (right + left) / 2.0; // change current height

56.                 } // end of if

57.             } // end of loop through binary search

58.

59.             if (isPossible) { // if it is possible

60.                 continue; // continue loop

61.             } // end of if

62.             System.out.printf("%.4f\n", currentHeight); // print result

63.         }

64.

65.         scanner.close(); // close Scanner object

66.     } // end of main method

67. } // end of HelpSeuMadruga class

68.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

EXPLICAÇÃO DO CÓDIGO

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

O código em questão é uma implementação em Java para resolver um problema que envolve a determinação da altura de um corte em tiras retangulares de papel, de modo a obter uma área total específica. Vamos analisar o código passo a passo:

1. Importação de Bibliotecas:

import java.util.Scanner;

O código começa importando a classe Scanner para facilitar a leitura de entrada do usuário.

2. Declaração da Classe:

public class HelpSeuMadruga

A classe principal é declarada como HelpSeuMadruga.

3. Método main:

public static void main(String[] args) {

Início do método principal main.

4. Leitura da Entrada:  
Scanner scanner = new Scanner(System.in);

Instancia um objeto Scanner para ler a entrada do usuário.

5. Loop Principal:

while (scanner.hasNextInt()) {

Inicia um loop que continua enquanto houver próximo número inteiro na entrada.

6. Leitura de N e A:

int numberStrip = scanner.nextInt();

int desiredHeight = scanner.nextInt();

Lê dois inteiros que representam a quantidade de tiras (numberStrip) e a área desejada (desiredHeight).

7. Verificação de Fim:

if (numberStrip + desiredHeight == 0) {

break;

}

Se a soma de numberStrip e desiredHeigh for zero, encerra o loop.

8. Leitura dos Comprimentos das Tiras:

int[] stripHeights = new int[numberStrip];

int totalHeight = 0;

for (int i = 0; i < numberStrip; i++) {

stripHeights[i] = scanner.nextInt();

totalHeight += stripHeights[i];

}

Lê os comprimentos das tiras e calcula a altura total.

9. Verificações e Impressões:

// ... (código omitido para economia de espaço)

10. Loop de Busca Binária:

while (right - left > 0.000001) {

// ... (código omitido para economia de espaço)

}

Implementa um loop de busca binária para encontrar a altura do corte necessário.

11. Saída Final:

if (isPossible) {

continue;

}

System.out.printf("%.4f\n", currentHeight);

Se for possível encontrar uma altura de corte, continua o loop; caso contrário, imprime a altura calculada.

12. Fechamento do Scanner:

scanner.close();

Fecha o objeto `Scanner` após a conclusão do programa.

O código utiliza uma abordagem de busca binária para encontrar a altura de corte desejada de forma eficiente. Ele processa vários casos de teste e imprime a altura do corte ou indica se não é possível alcançar a área desejada.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**EXECUÇÃO DO ALGORITMO**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Vamos analisar os exemplos dados para entender como o código funciona:

### Exemplo de Entrada 1:

```java

5 3

5 3 6 2 3

```

\*\*Explicação passo a passo:\*\*

1. `numberStrip = 5`, `desiredHeight = 3`.

2. Leitura dos comprimentos das tiras: `stripHeights = {5, 3, 6, 2, 3}`.

3. Cálculo da altura total: `totalHeight = 5 + 3 + 6 + 2 + 3 = 19`.

Agora, o código entra em um loop de busca binária para encontrar a altura do corte. Ele realiza iterações ajustando a altura do corte e verificando se a área total cortada é igual a `desiredHeight`. No final, imprime a altura do corte encontrada: `4.0000`.

### Exemplo de Entrada 2:

```java

4 14

2 5 2 5

```

\*\*Explicação passo a passo:\*\*

1. `numberStrip = 4`, `desiredHeight = 14`.

2. Leitura dos comprimentos das tiras: `stripHeights = {2, 5, 2, 5}`.

3. Cálculo da altura total: `totalHeight = 2 + 5 + 2 + 5 = 14`.

Neste caso, a altura total já é igual à `desiredHeight`, então a saída é `:D` (nenhum corte necessário).

### Exemplo de Fim de Entrada:

```java

0 0

```

O código encerra porque tanto `numberStrip` quanto `desiredHeight` são iguais a zero, indicando o fim da entrada.

Esses exemplos demonstram como o código lida com diferentes situações, utilizando a busca binária para encontrar a altura do corte desejada ou indicar se não é possível alcançar a área desejada.