

Exercícios usados nas avaliações

(Para ser rápido apenas copiei do moodle para esse arquivo, então a formatação pode não estar organizada, mas dá para entender o enunciado das questões)

Escreva uma função **calculaMinimo2()** que receba como parâmetro uma lista de números inteiros e retorne soma dos dois menores valores na lista. Gere a lista de forma automática, conforme apresentado nos exemplos em aula.

Exemplo de execução:

Informe o tamanho da lista: 5

Exibindo a lista: 4,3,6,1,2

Soma dos dois menores: 3

Observação: A resposta com o código completo da questão pode ser informada abaixo ou você pode anexar o arquivo no formato .java.

Escreva uma função **somaQuadrados()** que receba como parâmetro uma lista de números e retorne a soma dos quadrados dos números na lista.

Exemplo de execução:

Informe o tamanho da lista: 3

Exibindo a lista: 2, 3, 4

Soma dos quadrados: 29

Note no exemplo que 29 corresponde a seguinte soma: $4+9+16$.

Observação: A resposta com o código completo da questão pode ser informada abaixo ou você pode anexar o arquivo no formato .java.

Parte do treinamento de um novo garçom é carregar uma grande bandeja com várias latas de bebidas e copos e entregá-las todas numa mesa do restaurante. Durante o treinamento é comum que os garçons deixem cair as bandejas, quebrando todos os copos.

A SBC — Sociedade Brasileira de Copos — analisou estatísticas do treinamento de diversos garçons e descobriu que os garçons em treinamento deixam cair apenas bandejas que têm mais latas de bebidas que copos. Por exemplo, se uma bandeja tiver 10 latas e 4 copos, certamente o garçom em treinamento a deixará cair, quebrando os 4 copos. Já se a bandeja tiver 5 latas e 6 copos, ele conseguirá entregá-la sem deixar cair.

Escreva um programa que, dado o número de latas e copos em cada bandeja que o garçom tentou entregar, imprime o total de copos que ele quebrou.

Entrada: A primeira linha da entrada contém um inteiro N ($1 \leq N \leq 100$) representando o número de bandejas que o garçom tentou entregar. As N linhas seguintes representam as N bandejas. Cada linha contém dois inteiros L e C , indicando o número de latas e o número de copos naquela bandeja, respectivamente ($0 \leq L, C \leq 100$).

Saída: Seu programa deve imprimir uma única linha, contendo um único inteiro, indicando o número total de copos que o garçom quebrou. O formato mostrado nos exemplos de execução abaixo deve ser seguido rigorosamente.

Exemplos de entrada	Exemplos de saída
3 10 5 6 8 3 3	5

Os alunos do último ano do curso técnico em Informática resolveram organizar uma quermesse para arrecadar fundos para a festa de formatura. A festa prometia ser um sucesso, pois o pai de um dos formandos, Joaquim, dono de uma loja de informática, decidiu doar um tablet para ser sorteado entre os que comparecessem. Os alunos prepararam barracas de quentão, pipoca, doces, ensaiaram a quadrilha e colocaram à venda ingressos numerados sequencialmente a partir de 1. O número do ingresso serviria para o sorteio do computador. Ficou acertado que Joaquim decidiria o método de sorteio; em princípio o sorteio seria, claro, computadorizado.

O local escolhido para a festa foi o ginásio da escola. A entrada dos participantes foi pela porta principal, que possui uma roleta, onde passa uma pessoa por vez. Na entrada, um funcionário inseriu, em uma lista no computador da escola, o número do ingresso, na ordem de chegada dos participantes. Depois da entrada de todos os participantes, Joaquim começou a trabalhar no computador para preparar o sorteio. Verificando a lista de presentes, notou uma característica notável: havia apenas um caso, em toda a lista, em que o participante que possuía o ingresso numerado com i , havia sido a i -ésima pessoa a entrar no ginásio. Joaquim ficou tão encantado com a coincidência que decidiu que o sorteio não seria necessário: esta pessoa seria o ganhador do tablet.

Sua tarefa será: conhecendo a lista de participantes, por ordem de chegada, sua tarefa é determinar o número do ingresso premiado, sabendo que o ganhador é o único participante que tem o número do ingresso igual à sua posição de entrada na festa.

Entrada: A primeira linha de um conjunto de teste contém um número inteiro positivo N que indica o número de participantes da festa. A linha seguinte contém a sequência, em ordem de entrada, dos N ingressos das pessoas que participaram da festa. O formato mostrado no exemplo de entrada abaixo deve ser seguido rigorosamente.

Saída: A saída deve conter o número do ingresso do ganhador, conforme determinado pelo seu programa.

Exemplos de entrada	Exemplos de saída
4 4 5 3 1	3
10 9 8 7 6 1 4 3 2 12 10	10

A corrida de lesmas é um esporte que cresceu muito nos últimos anos, fazendo com que várias pessoas dediquem suas vidas tentando capturar lesmas velozes, e treiná-las para faturar milhões em corridas pelo mundo. Porém, a tarefa de capturar lesmas velozes não é uma tarefa muito fácil, pois praticamente todas as lesmas são muito lentas. Cada lesma é classificada em um nível dependendo de sua velocidade:

Nível 1: Se a velocidade é menor que 10 cm/h .

Nível 2: Se a velocidade é maior ou igual a 10 cm/h e menor que 20 cm/h .

Nível 3: Se a velocidade é maior ou igual a 20 cm/h .

Sua tarefa é identificar qual nível de velocidade da lesma mais veloz de um grupo de lesmas.

É obrigatório usar uma função chamada **calcularNivel()**, que recebe um único parâmetro chamado **velocidades** que é uma **lista de inteiros**. O **retorno da função deve ser obrigatoriamente um inteiro** que indica o nível da lesma mais rápida contida na lista.

O formato mostrado nos exemplos de execução abaixo deve ser seguido rigorosamente.

Exemplos de execução:

Exemplo 1:

Quantidade de lesmas: 10

Lista de velocidades:

10

10

10

10

15

18

20

15

11

10

Nível da lesma mais veloz: 3

Exemplo 2:

Quantidade de lesmas: 5

Lista de velocidades:

1

5

2

9

5

Nível da lesma mais veloz: 1

Exemplo 3:

Quantidade de lesmas: 7

Lista de velocidades:

19

9

1

4

5

8

6

Nível da lesma mais veloz: 2

Faça um programa para ler duas listas de valores inteiros e gerar uma terceira lista que seja a união das duas listas informadas pelo usuário.

As listas devem ter pelo menos 1 elemento. Caso contrário, deve ser exibida a mensagem "Erro - A lista deve ter pelo menos 1 elemento."

Formato de entrada

Número de elementos da primeira lista

Valores da primeira lista (separados por uma quebra de linha)

Número de elementos da segunda lista

Valores da segunda lista (separados por uma quebra de linha)

Formato de saída

A lista contendo todos os valores da primeira lista e dos valores da segunda lista.

Exemplos de entrada	Exemplos de saída
1 1 3 1 2 3	1 1 2 3
3 3 2 1 4 1 2 3 4	3 2 1 1 2 3 4
0	Erro - A lista deve ter pelo menos 1 elemento.
2 4 5 0	Erro - A lista deve ter pelo menos 1 elemento.

Descrição do Problema

João e Maria estão querendo obter informações sobre os carros de sua cidade. Para isso eles pediram que você escrevesse um programa para ajudá-los. Eles vão digitar informações de diferentes carros e quando quiserem parar a leitura vão digitar a resposta 'n' ou 'N'.

Para cada carro serão lidos o ano e a velocidade.

O programa deve exibir a quantidade de carros, o carro mais novo e o mais rápido.

Entrada:

Um valor inteiro (ano) e um em ponto flutuante (velocidade)

Um caractere

Saída: Maior velocidade, maior ano e velocidade média.

Exemplos de entrada	Exemplos de saída
s 1980 250.0 s 1999 172.0 s 2014 100.0 s 2000 200 n	250.00 2014 180.50
n	zero
s 1999 100.0 s 1972 120.0 s 1977 149.0 n	149.00 1999 123.00

Maria comprou vários tipos de doce de banana, de diversos fornecedores. Após provar e guardar os doces, Maria quis saber qual o doce mais barato e o valor total da compra. Faça um programa em Java que receba de início a quantidade total de doces comprados e em seguida, o peso (em gramas) e o valor (em R\$) de cada doce comprado por Maria. Calcule e exiba o **preço unitário** (R\$ por quilograma) de cada doce e no final, **informe o número do doce mais barato e o peso total e o gasto total realizado**.
Exemplo de execução: **(Utilize exatamente esse formato para exibição dos dados na tela, respeitando a nomenclatura e espaçamentos utilizados)**

```
Informe o total de doces comprados: 3
Doce 1:
Informe o Peso (g):1000
Informe o Preço (R$): 23.34
Preço unitário calculado = R$23.34/kg

Doce 2:
Informe o Peso (g): 500
Informe Preço (R$): 10.51
Preço unitário calculado = R$21.02/kg

Doce 3:
Informe o Peso (g): 1500
Informe o Preço (R$): 36.12
Preço unitário calculado = R$24.08/kg

Dados Finais da Compra:
Produto mais barato: Doce 2
Foram comprados 3000g de doce por R$69.97
```

Você conseguiu um estágio para trabalhar como programador no controle acadêmico do IFPB. Como primeira tarefa, Gleidson, o coordenador, solicitou que você desenvolva um programa simples. Gleidson pretende utilizar o programa para premiar o melhor aluno de cada turma do IFPB. Este programa tem como entrada **uma lista de matrículas e de médias finais dos alunos** de uma turma, e determina o aluno com **a maior média na turma**. Em caso de haver alunos empatados com a melhor média na turma, o programa deve produzir uma lista com todos esses alunos da turma que obtiveram a maior média, e não apenas um deles. Desenvolva este programa em Java.

Entrada: A primeira linha contém um número inteiro **N** que indica o total de alunos na turma. As **N** linhas seguintes contém, um par de números inteiros **C** e **M** ($0 \leq M \leq 100$), indicando respectivamente a matrícula e a média de um aluno. **O formato apresentado nos exemplos de entradas abaixo devem ser seguidos rigorosamente.**

Saída: A **saída** deve conter as matrículas dos alunos que obtiveram a maior média da turma, separados por um espaço em branco. As matrículas dos alunos devem aparecer na mesma ordem da entrada, e cada um deve ser

seguido de um espaço em branco. **O formato exibido nos exemplos de saída abaixo devem ser seguidos rigorosamente.**

Exemplo de Entrada	Exemplo de Saída
3 1 85 2 91 3 73	2
5 12300 81 12601 99 98012 76 10111 99 212 99	12601 10111 212

Após seus treinos para uma competição de natação, Benedito quer verificar seu desempenho em relação aos seus tempos de treino e ao tempo recorde regional. Para auxiliá-lo nesta tarefa, você deve fazer um programa em Java que solicite uma quantidade **N** de treinos realizados, o tempo do último recorde regional (em segundos), e para cada treino, solicite o **tempo** (em segundos) cronometrado. Para este conjunto de tempos, seu programa deve calcular o **maior** e o **menor** tempo, e a **média aritmética** dos tempos de treino. Ao final, seu programa deve exibir os resultados na tela e, caso o tempo médio seja menor que o tempo recorde informado, deve também mostrar uma mensagem indicando tal fato. Para calcular o tempo médio basta somar os tempos de cada treino e dividir pela quantidade de treinos. Observe os exemplos abaixo.

Exemplo de execução 1	Exemplo de execução 2
Informe a quantidade de Treinos: 5 Informe o Recorde Regional: 47 Digite o tempo 1: 45 Digite o tempo 2: 50 Digite o tempo 3: 44 Digite o tempo 4: 46 Digite o tempo 5: 48 Maior tempo: 50 segundos Menor tempo: 44 segundos Média: 46.6 segundos Você está abaixo do recorde!	Informe a quantidade de Treinos: 3 Informe o Recorde Regional: 47 Digite o tempo 1: 47 Digite o tempo 2: 48 Digite o tempo 3: 49 Maior tempo: 49 segundos Menor tempo: 47 segundos Média: 48 segundos