

II MARATONA DE PROGRAMAÇÃO INTERIF 2019

ETAPA FINAL

19/10/2019 – Campus Birigui

Caderno de Problemas

Informações Gerais

Este caderno contém oito problemas, as páginas estão numeradas de 1 a 15, não contando esta página de rosto. Verifique se o caderno está completo.

A) Sobre a entrada

- 1) A entrada de seu programa deve ser lida da entrada padrão.
- 2) A entrada é composta de um ou mais casos de teste, depende do problema.
- 3) Quando uma linha da entrada contém vários valores, estes são separados por um único espaço em branco; a entrada não contém nenhum outro espaço em branco.
- 4) Cada linha, incluindo a última, contém o caractere final-de-linha.
- 5) O final da entrada pode coincidir com o final do arquivo ou com uma entrada determinada

B) Sobre a saída

- 1) A saída de seu programa deve ser escrita na saída padrão.
- 2) Espaços em branco só devem ser colocados quando solicitado.
- 3) Cada linha, incluindo a última, deve conter o caractere final-de-linha.

C) Regras

- 1) Só é permitida a comunicação entre os membros de um mesmo grupo.
- 2) Não é permitido o acesso à internet.
- 3) Não é permitido o uso de qualquer aparelho de comunicação.
- 4) Não é permitido o uso de meios de armazenamento externos (*pendrives*, cartões de memória, hd externo, etc).
- 5) É permitida a consulta de qualquer material impresso.
- 6) Não é permitida a comunicação com o técnico (coach) do time.

Problema A**Figurinhas - Parte 2***Por Marcio Kassouf Crocomo (IFSP – campus Piracicaba)**Arquivo: figurinha.[c/cpp/java/py]****Timelimit: 1***

Joãozinho quer completar seu álbum de figurinhas e, para isso, quebrou seu cofrinho no qual guardava suas economias. Joãozinho quer comprar o máximo de pacotes de figurinhas que seu dinheiro permitir, mas também quer ficar com o máximo de moedas para começar a guardar dinheiro em um novo cofre. Assim, a estratégia adotada por Joãozinho para juntar o dinheiro para pagar o total da conta foi a de sempre priorizar gastar as moedas de maior valor. Por exemplo, Joãozinho só irá usar moedas de 10 centavos para pagar a conta se todas as suas moedas de 25 centavos se esgotarem. Seu Alcides, o dono da loja de figurinhas, apesar de perceber a estratégia adotada por Joãozinho, não dá troco.

Entrada

Na primeira linha, o valor de um pacote de figurinhas em reais (iniciado com R\$ e usando ponto como separador de casa decimal), que deverá estar entre R\$0.01 e R\$50.00. Na linha seguinte, a quantidade de moedas que Joãozinho possui de 25, 10, 5 e 1 centavos, nesta ordem, separadas por espaços. A quantidade máxima de moedas de um tipo que Joãozinho poderá possuir é 100.

Saída

Duas linhas, sendo a primeira o número de pacotes de figurinhas que Joãozinho irá comprar, e a segunda, a quantidade de moedas restantes.

Exemplo de entrada	Exemplo de saída
R\$3.71 10 10 10 10	1 15
R\$1.99 9 19 9 1	2 11

Problema B

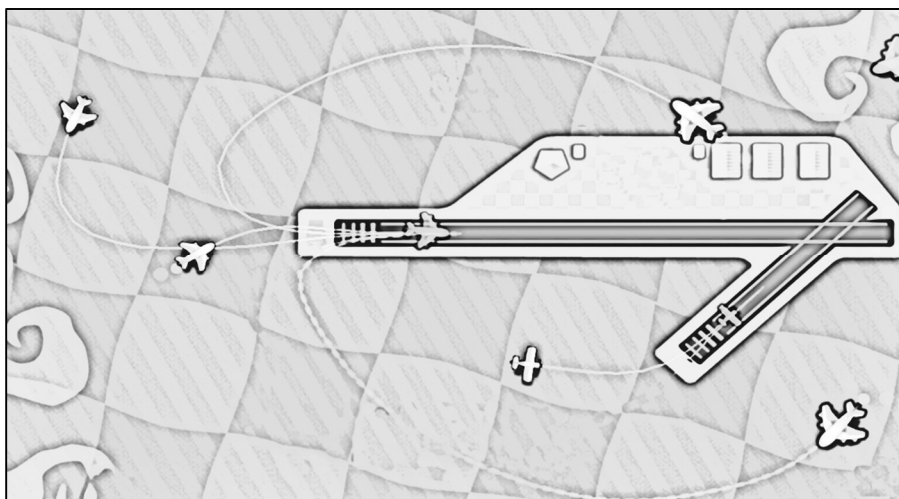
Controlador de Tráfego Aéreo

Criado por Jones Mendonça de Souza (IFSP – campus Barretos)

Arquivo: fly.[c|cpp|java|py]

Timelimit: 1

O aeroporto de Congonhas recebe todos os dias uma média de 600 pousos e decolagens, ou cerca de 36 por hora. No último ano, foram exatamente 223.989 movimentos aéreos. Para organizar todo o fluxo de aviões que chegam a Congonhas e saem de lá, a torre de controle funciona o tempo inteiro com nível máximo de atenção. Para descartar qualquer possibilidade de erro humano o chefe do controle de tráfego aéreo de Congonhas contratou a sua equipe da Maratona de Programação InterIF para desenvolver um programa que organize automaticamente o fluxo de aviões no campo de pouso. Para isso, basta seguir o seguinte protocolo, os aviões que veem do lado Oeste da pista têm maior prioridade de serem colocados na fila, pois são aqueles que estão mais próximo do localizador (início da pista). Já os aviões que estão se aproximando pelo lado Norte e Sul, devem ser inseridos na fila 1 por vez, ou seja, insere-se 1 avião do lado Norte e em seguida 1 avião do lado Sul. Por último, insere-se o próximo avião que esteja se aproximando ao lado leste da pista.



Entrada

A entrada é composta por um número inteiro P , representando o ponto cardeal do avião que entrou no campo da pista ($-4 \leq P \leq -1$), onde -4 representa o lado leste, -3 o lado norte, -2 lado sul e -1 lado oeste). Em seguida é realizada a entrada dos respectivos aviões, compostos de um identificador começando com a letra “A” seguida de um número inteiro I ($1 \leq I \leq 1000$). A qualquer momento é permitido trocar o ponto cardeal, e inserir novas aeronaves, repetidamente até que o controlador finalize a sessão com o dígito 0.

Saída

A saída é composta de uma linha contendo as aeronaves enfileiradas pela ordem do protocolo estabelecido pelo aeroporto. O resultado de seu programa deve ser escrito na saída padrão.

Exemplos de Entradas	Exemplos de Saídas
-4 A1 A26 A38 A23 -1 A80 A40 -2 A2 A16 A108 -3 A20 A44 0	A80 A20 A2 A1 A40 A44 A16 A26 A108 A38 A23
-4 A12 A33 -3 A8 A33 -2 A77 A102 A866 -3 A21 A15 A9 -1 A2 0	A2 A8 A77 A12 A33 A102 A33 A21 A866 A15 A9

Problema C

Dança das Cadeiras

Por Daniel Corrêa Lobato (IFSP – campus Piracicaba)

Arquivo: votos.[c/cpp/java/py]

Timelimit: 1

Alguns países, Brasil incluído, adotam um sistema eleitoral proporcional, onde são usados o quociente eleitoral e o quociente partidário. Esse modelo mistura dois métodos de alocação de cadeiras em órgãos colegiados, chamados de método de d'Hondt e o método de Jefferson, que prestigiam a força da representatividade do voto tanto ao candidato como ao partido. No Brasil, esse modelo é aplicado em todas as eleições para os cargos legislativos, menos para o cargo de senador, que utiliza o modelo de eleição majoritária: quem tem mais votos, é eleito.

Para o sistema eleitoral proporcional funcionar, precisamos de três indicadores – o quociente eleitoral, o quociente partidário, e a cláusula de barreira.

O quociente eleitoral (Q_e) é a razão entre o número de votos válidos (V_v) e o número de cadeiras a serem preenchidas (C) desprezada a fração se igual ou inferior a meio, ou adicionada de um, se superior (3,4 vira 3; 4,7 vira 5). Já o quociente partidário (Q_p) é a razão entre o total de votos que o partido recebeu e o quociente eleitoral, desprezando a parte fracionária.

Cada partido terá direito à parte inteira do quociente partidário, atribuindo-as para os candidatos mais votados do partido. Se a soma das cadeiras alocadas aos partidos for menor que o total de cadeiras disponíveis, as cadeiras restantes são divididas de acordo com o sistema de distribuição de sobras.

O terceiro indicador é a cláusula de barreira, que é aplicada após o cálculo do quociente partidário, do qual serão removidas as cadeiras dos candidatos que não tenham recebido pelo menos 10% do quociente eleitoral. As cadeiras removidas serão redistribuídas, também de acordo com o sistema de distribuição de sobras. Por exemplo, considerando $Q_e = 100$, em que determinado partido, ou coligação, dispute o pleito com apenas 2 candidatos (A e B), recebendo A 300 votos e B apenas 1, nesse caso, o candidato B não fará jus a nenhuma cadeira, apesar de o partido ter direito a 3 cadeiras, conforme o quociente partidário ($Q_p = 301/100 \approx 3$). Isso acontece porque o candidato B não tem pelo menos 10% de Q_e , ou seja, B precisaria de pelo menos 10 votos direcionados a ele para ter direito à cadeira conquistada pelo partido. O mecanismo da cláusula de barreira foi introduzido para evitar que partidos pequenos tivessem candidatos “puxadores de votos”.

Considere o exemplo a seguir, onde 9 cadeiras estão em disputa e há 6050 votos válidos (excluindo brancos e nulos). Os votos se distribuíram da seguinte forma.

Partido	Votos
A	1900
B	1350
C	550
D	2250
Total de votos	6050

O quociente eleitoral, Q_e , é calculado como se segue:

$$Q_e = \frac{V_v}{C} = \frac{6500}{9} \approx 672,22 \Rightarrow 672$$

O número de cadeiras obtido por cada partido pode ser calculado então.

Partido	Votos	Qe	Qp	Cadeiras
A	1900	672	$\frac{1900}{672} \approx 2,8273$	2
B	1350	672	$\frac{1350}{672} \approx 2,0089$	2
C	550	672	$\frac{550}{672} \approx 0,8184$	0
D	2250	672	$\frac{2250}{672} \approx 3,3482$	3
Total de votos	6050		Total de cadeiras	7
			Sobras	2

A distribuição das sobras é feita de forma iterativa, a partir do cálculo da razão entre o total de votos recebidos por cada partido e o número de cadeiras recebidas mais 1, atribuindo-se uma sobra ao partido que apresentar a maior média. Continuando o exemplo, temos

Partido	Votos	Cadeiras	Média	Rodada 1 de 2
A	1900	2	$1900/(2+1) = 633$	+1
B	1350	2	$1350/(2+1) = 450$	0
C	550	0	$550/(0+1) = 550$	0
D	2250	3	$2250/(3+1) = 562$	0

Assim, a primeira vaga de sobra vai para o partido A. Como ainda há uma vaga sobrando, repete-se o processo.

Partido	Votos	Cadeiras	Média	Rodada 2 de 2
A	1900	3	$1900/(3+1) = 475$	0
B	1350	2	$1350/(2+1) = 450$	0
C	550	0	$550/(0+1) = 550$	0
D	2250	3	$2250/(3+1) = 562$	+1

Agora, a vaga que sobrava foi atribuída ao partido D, que acaba o pleito com direito a 4 cadeiras, como pode ser visto no quadro final

Partido	Votos	Cadeiras
A	1900	3
B	1350	2
C	550	0
D	2250	4
Total de votos	6050	9

A partir deste ponto, pega-se a lista de votos de cada partido, e respeitando a cláusula de barreira, faz-se a atribuição das cadeiras aos candidatos em função do número de votos recebidos, mas isso fica para depois.

O seu objetivo é construir um programa que, sabendo o total de votos recebido por cada partido e o número de cadeiras em disputa, indique quantas cadeiras cabem a cada partido, considerando os critérios discutidos aqui.

Entrada

Os dados de entrada são compostos por uma linha contendo o número inteiro n de cadeiras em disputa ($0 < n \leq 500$), e um número inteiro p de partidos que estão na disputa pelas n cadeiras ($0 < p \leq 20$). Em seguida, há um total de p linhas, cada uma contendo número de votos v_i recebido por cada um dos p partidos ($1 \leq v_i \leq 30000$). A soma do total de votos é igual ou maior ao número de cadeiras em disputa. Todo o partido recebe, pelo menos, um voto.

Saída

A saída deve conter p linhas, cada uma contendo um número inteiro que corresponde ao total de cadeiras alocada para cada partido.

Exemplos de Entradas	Exemplos de Saídas
9 4 1900 1350 550 2250	3 2 0 4

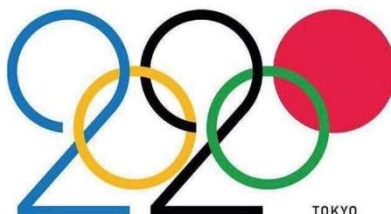
Problema D

Placar Olímpico

Criado por Tiago Alexandre Dócusse (IFSP – campus Barretos)

Arquivo: olimpиаda.[c|cpp|java|py]

Timelimit: 1



A olimpíada de Tóquio 2020 está chegando, e você decidiu visitar o Japão para conhecer a cidade antes dos jogos. Em sua visita, encontrou-se com o comitê organizador dos jogos e, após ficar sabendo do seu incrível talento com programação, o comitê decidiu te contratar para escrever o algoritmo que fará a classificação dos países nas olimpíadas de 2020. Você está preparado para a tarefa?

Entrada

O programa deve receber como entrada o nome do país, seguido da quantidade de medalhas de ouro, prata e bronze conseguidas por este país. É possível ter entradas de uma quantidade qualquer de países. Para indicar término da entrada de países, é necessário informar a palavra “FIM”.

Saída

A saída do programa deve exibir a classificação dos países, do melhor para o pior, de acordo com a quantidade de medalhas: o país mais bem colocado é o que possui a maior quantidade de medalhas de ouro; em caso de empate, o país mais bem colocado é o que possui a maior quantidade de medalhas de prata; em caso de empate, o país mais bem colocado é o que possui a maior quantidade de medalhas de bronze. Em caso de países empatados (mesma quantidade de medalhas de ouro, prata e bronze), exibi-los em ordem alfabética. Não se esqueça de exibir a colocação do país no quadro de medalhas (países empatados devem ser exibidos com a mesma colocação).

Exemplos de Entradas	Exemplos de Saídas
Brasil 1 2 0 EUA 5 3 1 Noruega 3 1 1 FIM	1: EUA 2: Noruega 3: Brasil
Brasil 2 0 1 China 5 3 1 Espanha 2 1 0 Portugal 2 0 0 Chile 1 0 2 Argentina 2 0 1 FIM	1: China 2: Espanha 3: Argentina 3: Brasil 5: Portugal 6: Chile

Problema E

Candlestick

Por Cássio Agnaldo Onodera (IFSP - campus Birigui)

Arquivo: `candle.[c|cpp|java|py]`

Timelimit: 1

Os investidores que negociam ações na bolsa de valores (*traders*) costumam fazer uma leitura de gráficos para decidirem se devem comprar ou vender um determinado ativo.

Existem vários tipos de gráficos que demonstram as variações dos preços das ações, mas o mais utilizado é o gráfico de velas, mais conhecido com *candlestick*.

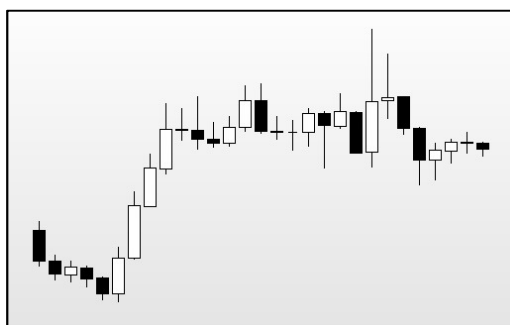
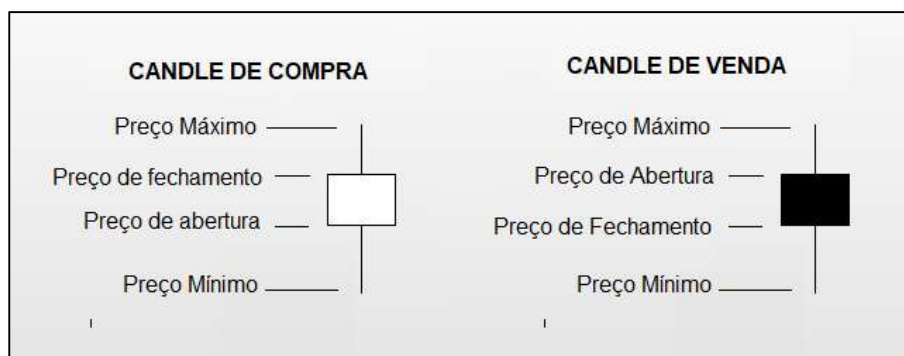


Gráfico de Candlestick

No gráfico acima temos “velas” (*candles*) pretas e “velas” brancas, que podem ser substituídos por vermelhas e verdes respectivamente. Um *candle* preto, indica que o preço caiu (venda), ou seja, na abertura do *candle* o preço de um ativo era negociado por um valor e no fechamento do *candle*, este ativo era negociado por um valor menor. Já um *candle* branco, indica que o preço subiu (compra), ou seja, na abertura era negociado por um valor e no fechamento era negociado por um valor maior.

Cada *candle* representa a variação de tempo em um período de tempo, que pode ser de 1 minuto, 3 minutos, 5 minutos, 10 minutos ou mesmo de 1 dia ou 1 semana. O *trader* determina em qual tempo gráfico ele deseja ver os *candles*.

Além do preço de abertura e fechamento, os *candles* indicam o preço máximo e o preço mínimo.



Candles de compra e de venda

As linhas abaixo ou acima dos preços de abertura e fechamento são chamadas de pavio. Os pavios, indicam até onde o preço chegou.

Atualmente um número enorme de pessoas estão iniciando esta atividade mas não entendem o significado destes gráficos. Seu papel será ajudar estas pessoas. Você deve desenvolver um programa que ao receber os dados de variação do preço de um ativo, desenhe o *candle* referente à esta variação.

O desenho deverá ser feito utilizando-se apenas textos. Nos *candles* de compra, você deve utilizar o caracter “C” e nos *candles* de venda, você deve utilizar o caracter “V”. Quando o preço de fechamento for o mesmo que o preço da abertura, você deve utilizar o caracter “X”.

Entrada

A entrada é composta por quatro números inteiros A, F, C e B ($0 \leq A, F, C, B \leq 1000000$) indicando os preços de abertura, fechamento, máximo e mínimo respectivamente.

Saída

O programa deverá desenhar utilizando os caracteres “C”, “V” ou “X” os *candles* que representam a variação de preço.

Exemplos de Entradas	Exemplos de Saídas
102 99 104 95	V V VVVVV VVVVV VVVVV VVVVV V V V V
35 37 38 33	C CCCCC CCCCC CCCCC C C

Problema F

Bordando em Ponto Cruz

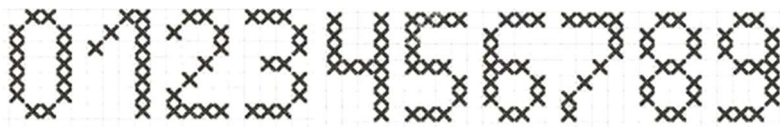
Por: Jones Mendonça de Souza (IFSP-Barretos)

Arquivo: pontocruz.[c|cpp|java|py]

Timelimit: 1

Taismara é uma das melhores bordadeiras da cidade de Jaborandi, ela utiliza a técnica de Ponto em Cruz para confeccionar seus bordados. Um dos pedidos mais requisitados é a confecção de tolhas de banho para bebês recém-nascidos. Como Jaborandi é uma cidade pequena a maioria dos bebês nascem de parto normal sem data pré-definida, pegando as mães de surpresa e também a bordadeira. A tradição na cidade de Jaborandi diz que os bebês terão saúde para a vida inteira se tomarem o primeiro banho enxugando com uma toalha gravada com a data de nascimento. Sendo assim, os familiares procuram por Taismara para que ela borde a data de nascimento nas tolhas no mesmo dia. Diante deste contexto, a maior dificuldade de Taismara é de montar a arte da data de nascimento para transpor no material étamine. Como você é um bom programador e está na final da II Maratona de Programação InterIF possivelmente ajudará Taismara com este problema. Para isso, desenvolva um programa que leia uma data de nascimento e mostre a arte resultante para o bordado seguindo os padrões da Figura I.

Figura I – Ilustração dos caracteres utilizados para o bordado



(A) Números de 0 a 9



(B) Caractere ponto (.)

Entrada

A entrada é composta de uma string N contendo a data de nascimento no seguinte formato: DD.MM.YYYY

Saída

Seu programa deverá exibir cada caractere de entrada conforme o modelo apresentado na Figura I, utilizando-se o caractere “X” maiúsculo para a impressão de pontos em cruz, e “ ” (vazio) para aqueles pontos que não forem bordados. Cada dígito possui 7 linhas e 4 colunas, exceto o caractere ponto Figura I (B), que possui 7 linhas e 1 coluna. Entre cada caractere deverá ser impresso dois espaços em branco, cuide para que seu programa imprima o final de linha.

Exemplos de Entradas	Exemplos de Saídas
07.06.2018	<pre> XX XXXX XX XXX XX XX XX XX X X X X X X X X X X X X X X X X X X X X X X X X X X X X X XXX X X X X XX X X X X X X X X X X X X X X X X X X X X X X X X X X X XX X X XX XX X XXXX XX X XX </pre>
23.12.2019	<pre> XX XXX XX XX XX XX XX XX X X X X X X X X X X X X X X X X X X X X X X X XX X X X X X XXX X X X X X X X X X X X X X X X X X X XXXX XXX X X XXXX X XXXX XX X XXX </pre>

Problema G

Bruxa do 171

Por André Rodrigues da Cruz (IFSP – campus São Paulo)

Arquivo: bruxa.[c|cpp|java|py]

Timelimit: 1

Rapunzel, João e Maria formaram este ano a equipe Historinhas, do campus Sitiolândia, para disputar o II InterIF. Depois de muito estudo e dedicação eles conseguiram se classificar para a etapa final em Birigui no dia 19 de Outubro de 2019. Porém, infelizmente, a malvada Bruxa do 171 lançou um feitiço sobre a educação do país que fez minar os recursos financeiros de muitas escolas.



A instituição da equipe Historinhas foi fortemente afetada pela maldição da bruxa e corre o risco de não participar da tão esperada final. Pois, ela dispõe de pouco dinheiro para comprar combustível e não se sabe se é suficiente para ir e voltar de Birigui.

Rapunzel, João e Maria foram pedir a ajuda do sábio Mago Dijkstra, que vive no mais alto dos vértices nas Montanhas Encantadas. Ele observou nos emaranhados de arestas que existe um caminho, que é exatamente o mais curto.

Ajude a equipe Historinhas para que ela saiba a distância a percorrer de ida e volta.

Entrada

A entrada é composta por um caso de teste. Na primeira linha do caso de teste há, separados por um espaço, um inteiro N que indica o número de cidades do estado (incluindo Sitiolândia e Birigui) e um inteiro M que indica o número de ligações entre pares de cidades. Na segunda linha há, separados por um espaço, dois inteiros distintos S e B , que indicam respectivamente o índice da cidade de Sitiolândia e Birigui. Depois, seguem-se M linhas com os inteiros distintos U , V e D , separados por um espaço, indicando que existe uma ligação de mão dupla entre as cidades de índices U e V por uma distância D .

Saída

A saída do caso de teste deve apresentar um inteiro que representa a distância mínima percorrida de ida e volta da equipe Historinhas.

Exemplos de Entradas	Exemplos de Saídas
6 7 2 5 0 1 1 0 2 2 1 3 5 1 4 2 2 3 10 3 5 15 4 5 3	16
6 8 0 5 0 1 1 0 2 7 1 3 9 1 5 15 2 4 4 3 4 10 3 5 5 4 5 3	28
10 15 1 9 2 3 3 2 4 9 8 5 7 0 6 20 9 6 2 2 1 10 4 3 8 2 6 10 2 5 1 6 3 7 5 4 6 5 3 1 4 1 7 0 1 7 5 7 6	42

Problema H

Continuando a Investir

Por Daniel Corrêa Lobato (IFSP – campus Piracicaba)

Arquivo: medio.[c/cpp/java/py]

Timelimit: 1

No início do ano, você ajudou os investidores iniciantes a calcular, a partir de uma Nota de Corretagem, o preço final de cada ação operada, fazendo a apropriação adequada dos custos.

Agora, é a hora de começar a ganhar dinheiro com isso!

Um investidor normalmente faz “preço médio” de um papel. Isso quer dizer que ele compra um determinado papel em vários momentos, com preços diferentes. A ideia é tentar entrar no papel em preços diferentes, exatamente para tê-lo na carteira com um preço que seja a média do preço dele ao longo de determinado tempo. Depois você pode procurar as razões pelas quais isso é bom, e porque alguns investidores fazem errado, e acabam fazendo “preço mérdio”.

O preço médio é calculado como uma média ponderada dos preços de compra. Por exemplo, se forem compradas 100 ações de determinado papel a R\$10,00, e 300 ações do mesmo papel a R\$ 7,00, o preço médio é de $\frac{100 \times 10 + 300 \times 7}{100 + 300} = 7,75$.

Quando um papel é vendido, calcula-se o LAIR (lucro antes do imposto de renda) como sendo a diferença entre o preço de venda e o preço médio de compra. Sobre o lucro da operação, o investidor paga 15% de imposto de renda, mas legislação permite que sejam alienados (vendidos) até R\$20.000,00 por mês sem que seja necessário pagar imposto de renda sobre o lucro.

O seu trabalho é calcular o imposto de renda a ser pago por um investidor depois de um conjunto de operações feitas por ele no mês

Entrada

Os dados de entrada são compostos por uma linha contendo o número N ($1 \leq N \leq 15$) de operações de compra que foram feitas no papel, e um conjunto de N linhas contendo o número de ações compradas do papel (valor inteiro, estritamente positivo, até 10000), e o valor de cada papel (valor real, estritamente positivo, até 1000); a linha seguinte contém o preço de venda cada ação (valor real, estritamente positivo, até 10000), considerando que todas as ações foram vendidas. A operação sempre possui lucro maior ou igual a zero.

Saída

A saída deve conter duas linhas. A primeira contém o LAIR (com duas decimais), e a segunda contém o valor do imposto a ser pago, com duas casas decimais. Se não houver imposto a pagar, a saída deve ser “0.00”.

Exemplos de Entradas	Exemplos de Saídas
2 100 10 300 7 10	900.00 0.00
3 2000 8 1200 5 100 9 11	13400.00 2010.00