

Universidade Federal de Goiás
Instituto de Informática
Introdução à Programação 2017-2
Lista de Exercícios L2

Prof. Msc. Elias Batista Ferreira
Prof. Dr. Gustavo Teodoro Laureano
Profa. Dra. Luciana Berretta
Prof. Dr. Thierson Rosa Couto

Sumário

1	Lê número (+)	2
2	Classificação do Aço	3
3	Quermesse	4
4	Número perfeito	5
5	Interceptos em x e em y de uma Reta	6
6	Número Invertido	7
7	Raízes de equações de grau 2	8
8	José	10
9	Triângulo ou trapézio? (+++)	11
10	Valor em Notas e Moedas	13

1 Lê número (+)



(+)

Escreva um programa que leia n números e apresente sua média. Você deve implementar uma função, denominada `le_numero`, que leia um número via terminal e retorne seu valor. A leitura dos dados deve ser realizada via a função `le_numero`. A função deve seguir o protótipo:

```
1 double le_numero();
```

Entrada

O programa deve ler um número n e em seguida n números reais.

Saída

O programa deve apresentar a média dos números lidos com 2 casas decimais.

Exemplo

Entrada	Saída
3 2 2 2	2.00

2 Classificação do Aço



(+)

Um certo aço é classificado de acordo com o resultado de três testes abaixo, que devem determinar se o mesmo satisfaz às especificações:

1. Conteúdo de Carbono abaixo de 7.
2. Dureza Rockwell maior do que 50.
3. Resistência à tração maior do que 80.000 psi.

Ao aço é atribuído o grau “10” se passar por todos os testes; grau “9” se passar somente nos testes 1 e 2; grau “8” se passar no teste 1 apenas; grau “7” caso o aço não se enquadre nos graus, “10”, “9”, e “8”.

Escreva uma função que receba como parâmetros o valor de conteúdo de carbono, o valor da dureza e o valor de resistência à tração de um aço e retorne o grau do aço. Desenvolver um programa que leia o conteúdo do carbono (CC), a dureza Rockwell (DR) e a resistência à tração (RT), chame a função especificada acima e imprima a classificação do aço.

Entrada

A entrada é formada por três linhas. A primeira, contém um valor inteiro correspondendo ao conteúdo do carbono (CC). A segunda linha contém um valor inteiro correspondendo à dureza Rockwell (DR). A terceira linha, contém um valor inteiro correspondendo à resistência à tração (RT).

Saída

O programa deve imprimir uma linha, contento a frase ACO DE GRAU = x , onde x é um dos graus possíveis de classificação do aço (7, 8, 9, ou 10). Após o valor do grau do aço, o program deve imprimir o caractere de quebra de linha ‘\n’.

Exemplo

Entrada
3
57
96783
Saída
ACO DE GRAU = 10

Entrada
2
61
80000
Saída
ACO DE GRAU = 9

Entrada
4
39
77000
Saída
ACO DE GRAU = 8

Entrada
7
32
65234
Saída
ACO DE GRAU = 7



(++)

3 Quermesse

Os alunos do último ano 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, Teófilo, dono de uma loja de informática, decidiu doar um computador 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 Teófilo 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, Teófilo 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. Teófilo ficou tão encantado com a coincidência que decidiu que o sorteio não seria necessário: esta pessoa seria o ganhador do computador.

Tarefa

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. **Você deve escrever uma função do tipo `int` que retorne o índice da pessoa sorteada.**

Entrada

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de teste contém um número inteiro positivo N , $N \leq 200$, 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 final da entrada é indicado quando $N = 0$. Para cada conjunto de teste da entrada haverá um único ganhador.

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas. A primeira linha identifica o conjunto de teste, no formato "Teste n ", onde n é numerado a partir de 1. A segunda linha deve conter o número do ingresso do ganhador, conforme determinado pelo seu programa. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo

Entrada:
4
4 5 3 1
10
9 8 7 6 1 4 3 2 12 10
0

Saída:
Teste 1
3
Teste 2
10

4 Número perfeito



(++)

Dado um número n inteiro e positivo, dizemos que n é perfeito se n for igual à soma de seus divisores positivos diferentes de n . Construa um programa que leia um número inteiro n , apresente a soma dos divisores de n e verifique se o número informado é perfeito ou não.

Escreva uma função `somaDivisores` que receba como parâmetro um número inteiro e retorne a soma dos divisores desse número excluindo o próprio número como divisor de si mesmo. Seu programa deve chamar a função `somaDivisores` para resolver o problema.

Entrada

O programa deve ler um número inteiro n .

Saída

O programa deve apresentar uma linha contendo o texto: " $n = d_1 + d_2 + d_3 + \dots + d_k = x$ (MENSAGEM)", onde n é o número lido, d_i são os divisores de n em ordem crescente, x é a soma dos divisores e MENSAGEM é a mensagem "NUMERO PERFEITO" ou "NUMERO NAO E PERFEITO".

Observações

Suponha que o usuário sempre fornecerá um número maior que 1.

Exemplo

Entrada
6
Saída
6 = 1 + 2 + 3 = 6 (NUMERO PERFEITO)

Entrada
12
Saída
12 = 1 + 2 + 3 + 4 + 6 = 16 (NUMERO NAO E PERFEITO)

5 Interceptos em x e em y de uma Reta



(++)

Dada a equação de uma reta: $y = ax + bx$, o *intercepto* em x corresponde ao ponto em que a reta toca o eixo x . O intercepto em y é o ponto em que a reta cruza o eixo y . Escreva um programa que leia uma quantidade n de coeficientes de equações de reta e imprima para cada um os interceptos em x e em y da reta.

Escreva duas funções `intercetoEmX` e `interceptoEmY`. Ambas possuem dois parâmetros de entrada do tipo `float` que correspondem aos coeficientes a e b da equação de uma reta e ambas possuem dois parâmetros de saída do tipo `float` correspondentes às coordenadas (x,y) do ponto intercepto correspondente a cada uma das funções. Seu programa deve usar essas funções para computar os interceptos em x e em y . Considere que a é sempre diferente de zero.

Entrada

A primeira linha da entrada contém um inteiro $n > 0$ que corresponde ao número de casos de teste. Em seguida há n linhas, cada uma correspondente a um caso de teste. Cada linha contém dois valores do tipo `float`, correspondentes aos coeficientes de uma reta.

Saída

Para cada caso de teste o programa deve imprimir duas linhas: “Intercepto em X: $(x,0)$ ”, e “Intercepto em Y: $(0.00,y)$ ”, onde $(x,0.00)$ corresponde ao intercepto em x e $(0,y)$ corresponde ao intercepto em y . Os valores de x e de y devem ser impressos com duas casas decimais.

Exemplo

Entrada	
2	
-9 4	
-9 0	
Saída	
Intercepto em x:	(0.44, 0.00)
Intercepto em y:	(0.00, 4.00)
Intercepto em x:	(0.00, 0.00)
Intercepto em y:	(0.00, 0.00)

6 Número Invertido



(++)

Escreva um programa para ler um número de três dígitos e imprimir o número invertido. Seu programa deve ter uma função: `separaDigitos` que possui quatro parâmetros. O primeiro, é um parâmetro de entrada e corresponde a um número inteiro (com três dígitos), os demais três parâmetros são de saída, e correspondem, respectivamente ao primeiro, ao segundo e ao terceiro dígitos do número correspondente ao primeiro parâmetro. A função recebe um valor inteiro no primeiro parâmetro e devolve os dígitos que o formam nos três parâmetros seguintes.

Entrada

A entrada contém apenas um número com três dígitos. Esse número é diferente de zero e não é múltiplo de 10 ou 100.

Saída

A saída deve conter apenas uma linha com o número correspondente ao valor da entrada, com seus dígitos invertidos. Logo após o número, deve ser impresso o caractere de quebra de linha: `'\n'`.

Exemplos

Entrada
123
Saída
321
Entrada
987
Saída
789

7 Raízes de equações de grau 2



(++)

Desenvolver um programa que leia os coeficientes (a , b e c) de uma equação de segundo grau e calcule as raízes da equação. O programa deve mostrar a classificação das raízes, e, quando possível, o valor das raízes calculadas.

Seu programa deve criar uma função `raizesEq2Grau` que tenta computar as raízes de uma equação do segundo grau. A função deve retornar 2 se existir duas raízes reais distintas entre si, ou 1 se existir uma única raiz real, ou ainda, zero se as raízes são imaginárias. A função deve ter como parâmetros de entrada os coeficientes a , b , c de uma equação de segundo grau e deve ter dois parâmetros de saída, correspondendo às raízes da equação. No caso em que a função retorna 0 (raízes imaginárias) os parâmetros de saída não são utilizados pelo seu programa.

Entrada

O programa deve ler três valores reais na entrada. O primeiro valor corresponde ao valor do coeficiente a , o segundo, do coeficiente b e o terceiro, do coeficiente c , de uma equação de segundo grau. Os três valores ocorrem em uma única linha na entrada, separados entre si por um espaço.

Saída

O programa deve imprimir uma linha contendo uma das seguintes frases, conforme for o resultado do cálculo das raízes da equação: RAIZES DISTINTAS, ou RAIZ UNICA, ou RAIZES IMAGINARIAS. No primeiro caso o programa deve imprimir uma outra linha contendo a frase $X1 = x_1$, onde x_1 é o valor da menor raiz encontrada para a equação. Ainda no primeiro caso, o programa deve imprimir uma terceira linha com a frase $X2 = x_2$, onde x_2 corresponde ao valor da segunda raiz. No segundo caso, o programa deve imprimir uma frase $X1 = x_1$, onde x_1 é o valor da única raiz da equação. O terceiro caso não há o que imprimir pois as raízes são imaginárias.

Observações

Dada uma equação do segundo grau do tipo $ax^2 + bx + c$, Δ (delta) $= b^2 - 4ac$. Se $\Delta = 0$, a raiz da equação é ÚNICA. Se $\Delta < 0$. As raízes da equação são IMAGINÁRIAS. Se $\Delta > 0$, então há duas RAÍZES DISTINTAS para a equação. A fórmula geral para computar as raízes de uma equação do segundo grau é a fórmula de Báskara, dada por:

$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$

Exemplo

A seguir são mostrados três exemplos distintos de entrada, e suas correspondentes saídas, entretanto, existe apenas uma linha de entrada para esse problema.

Entrada
2 12 10
Saída
RAIZES DISTINTAS
X1 = -1.00
X2 = -5.00

Entrada
2 12 18
Saída
RAIZ UNICA X1 = -3.00

Entrada
15 17 89
Saída
RAIZES IMAGINARIAS

8 José



(+++)

João tem um irmão mais novo, José, que começou a ir à escola e já está tendo problemas com números. Para ajudá-lo a pegar o jeito com a escala numérica, sua professora escreve dois números de três dígitos e pede a José para comparar esses números. Mas em vez de interpretá-los com o dígito mais significativo à esquerda, ele deve interpretá-lo com o dígito mais significativo à direita. Ele tem que dizer à professora qual o maior dos dois números. Escreva um programa que irá verificar as respostas de José.

Escreva uma função `inverte` que receba como parâmetro um número inteiro (de três dígitos) e retorne o número invertido. Sugestão: aprenda a aproveitar código! Copie a função `SeparaDigitos` que você escreveu para o problema “Número Invertido” e cole no seu programa. Faça a função `inverte` chamar a função `SeparaDigitos` para te auxiliar a computar o número invertido. O seu programa deve chamar a função `inverte` quantas vezes for necessário para resolver o problema.

Entrada

A entrada conterá um inteiro T , o número de casos de testes, e, para cada caso de teste, uma única linha com dois números de três dígitos, A e B , os quais não serão iguais e não conterão zeros.

Saída

A saída deve conter, numa linha para cada caso de teste, com o maior dos números na entrada, comparados como descrito no enunciado da tarefa. O número deve ser escrito invertido, para mostrar a José como ele deve lê-lo.

Exemplo

Entrada
3
734 893
221 231
839 237
Saída
437
132
938

9 Triângulo ou trapézio? (+++)



(+++)

Leia três valores reais (A , B e C) e verifique se eles formam ou não um triângulo. Em caso positivo, calcule o perímetro do triângulo e imprima a mensagem:

Perimetro = XX.X

Caso os valores não formem um triângulo, calcule a área do trapézio que tem A e B como base e C como altura, mostrando a mensagem:

Area = XX.X

Escreva as seguintes funções para resolver o problema:

- Uma função `E_Triangulo` que receba como parâmetros três valores do tipo float correspondendo aos três lados de um possível triângulo e que retorne um número inteiro contendo um dos possíveis valores: 1 - se os três números formam um triângulo, ou 0 - se os três números não formam os lados de um triângulo.
- Uma função `Perimetro` que receba como parâmetros os valores de três lados de um triângulo e retorne um valor float correspondente ao perímetro do triângulo correspondente.
- Uma função `areaTrapezio` que receba como parâmetros três valores float correspondendo, respectivamente, à base inferior, à base superior e à altura de um trapézio e retorne um valor float correspondendo à área do trapézio.

Seu programa deve ler três valores float, verificar se formam um triângulo através do uso da função `E_Triangulo`. Se formarem triângulo, seu programa deve chamar a função `Perimetro` para computar o perímetro do triângulo formado pelos três valores e imprimir o perímetro. Se não formarem um triângulo, seu programa deve chamar a função `areaTrapezio` e deve imprimir a área do trapézio formado pelos valores.

Entrada

A entrada é formada por uma linha contendo três valores decimais separados um do outro por um espaço em branco.

Saída

A saída deve conter em uma única linha a frase apropriada. Observe nos exemplos acima que a saída deve conter apenas uma casa decimal. Os valores “X” que aparecem nos formatos são substituídos por dígitos que formam o valor de saída. Depois desses valores o programa deve imprimir o caractere de quebra de linha: ‘\n’.

Observações

Para que os três valores: A , B e C formem um triângulo as três condições abaixo devem ser satisfeitas:

- $|b - c| < a < b + c$;
- $|a - c| < b < a + c$;
- $|a - b| < c < a + b$;

A área de um trapézio é computada como $\text{Área} = \frac{(A+B)*C}{2}$.

Para imprimir um valor float com apenas uma casa decimal você deve usar a função `printf` com o código de formato “%.1f”.

Exemplo

Entrada
6.0 4.0 2.0
Saída
Area = 10.0

Entrada
6.0 4.0 2.1
Saída
Perimetro = 12.1

10 Valor em Notas e Moedas



(+++)

Escreva um algoritmo para ler um valor em reais e calcular qual o menor número possível de notas de \$R 100, \$R 50, \$R 10 e moedas de \$R 1 em que o valor lido pode ser decomposto. O programa deve escrever a quantidade de cada nota e moeda a ser utilizada.

Você deve escrever uma função `converteEmNotasMoedas` que possui 5 parâmetros. O primeiro parâmetro corresponde ao valor inteiro a ser convertido em notas e moedas, o segundo parâmetro corresponde ao número de notas de 100, o terceiro, corresponde ao número de notas de 50, o quarto, corresponde ao número de notas de dez e o quinto parâmetro corresponde ao número de moedas de 1 Real. Seu programa deve chamar essa função para resolver o problema proposto.

Entrada

O programa deve ler uma única linha na entrada, contendo um valor em Reais. Considere que somente um número inteiro seja fornecido como entrada.

Saída

O programa deve imprimir quatro frases, uma em cada linha: NOTAS DE 100 = X , NOTAS DE 50 = Y , NOTAS DE 10 = Z , MOEDAS DE 1 = W , onde X , Y , Z e W correspondem às quantidades de cada nota ou moeda necessárias para corresponder ao valor em Reais dado como entrada. Após cada quantidade, o programa deve imprimir um caractere de quebra de linha: ‘\n’.

Exemplo

Entrada
46395
Saída
NOTAS DE 100 = 463
NOTAS DE 50 = 1
NOTAS DE 10 = 4
MOEDAS DE 1 = 5