Universidade Federal de Goiás
Instituto de Informática
Introdução à Programação
Lista - L1- c

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

Instruções para a Resolução dos Problemas

Os problemas devem ser submetidos ao sistema Sharif da sua turma. A pontuação de cada problema é definida de acordo com o grau de dificuldade do problema, conforme a tabela abaixo:

Grau de Dificuldade	**Pontos**
+	1
++	2
+++	3
++++	4
+++++	5

A lista Lista L1 completa vale 100 pontos (que correspondem a 10 em termos de nota da lista). No Sharif a Lista L1 aparece segmentada em três listas. Este texto corresponde à lista L1- c. Para obter os cem pontos o aluno deve conseguir resolver um número de exercícios de graus de dificuldade +, ++ e +++ que somados formem 90 pontos. Pontos excedentes obtidos com exercícios desses graus de dificuldade serão descartados. Os 10 pontos restantes devem ser obtidos resolvendo-se problemas com graus de dificuldade ++++ ou +++++.

Sumário

1 Cálculo da raiz quadrada

 (+++)

Os Babilônios utilizavam um algoritmo para aproximar uma raiz quadrada de um número qualquer, da seguinte maneira:

Dado um número n, para calcular $r = \sqrt{n}$ assume-se uma aproximação inicial $r_0 = 1$ e calcula-se r_k para $k = 1, \ldots, \infty$ até que $r_k^2 \approx n$. O algoritmo deve realizar a aproximação enquanto $|n - r_k^2| > e$. O método babilônico é dado pela seguinte equação:

$$r_k = \frac{r_{k-1} + \frac{n}{r_{k-1}}}{2} \tag{1}$$

Entrada

O programa deve ler um número **double** n, cuja raiz quadrada deseja-se obter, e o erro e que deverá ser considerado pelo algoritmo.

Saída

A saída deve apresentar cada iteração do algoritmo, sendo cada linha composta pelo valor aproximado da raiz quadrada de n com 9 casas decimais, seguido do erro, também com 9 casas decimais.

Exemplo

Entrada
`2` `0.00001`
Saída

`r:  1.500000000, err:  0.250000000`
`r:  1.416666667, err:  0.006944444`
`r:  1.414215686, err:  0.000006007`

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".

Desenvolver um programa que leia o conteúdo do carbono (CC), a dureza Rockwell (DR) e a resistência à tração (RT) e fornece 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 Companhia de Teatro

 (+++)

Uma companhia de teatro deseja dar uma série de espetáculos. A direção calcula que o ingresso sendo vendido ao valor comum de mercado ($ValorIngresso$), serão vendidos 120 ingressos e que as despesas fixas serão de R$ 200,00 mais R$ 0,05 por cada ingresso. Diminuindo-se R$ 0,50 o preço dos ingressos, espera-se que as vendas aumentem em 25 ingressos. Aumentando-se R$ 0,50 o preço dos ingressos, espera-se que as vendas diminuam 30 ingressos. Para resolver este problema, a companhia de teatro deseja que você faça um programa que escreva uma lista de valores de lucros esperados em função do preço do ingresso, fazendo-se variar esse preço de A a B de R$ 1,00 em R$ 1,00. O programa deve apresentar na tela um resumo contendo o preço do ingresso informado, o lucro máximo calculado e a quantidade de ingressos vendidos para a obtenção desse lucro.

Entrada

O programa deve ler três números reais: $ValorIngresso$, correspondente ao valor de mercado dos ingressos, $ValorInicial$ e $ValorFinal$ correspondentes ao intervalo de valores que se deseja testar. Caso o $ValorInicial$ informado seja maior ou igual ao $ValorFinal$, o programa deve encerrar após apresentar a mensagem: "INTERVALO INVALIDO."

Saída

O programa deve apresentar na tela uma linha para cada valor testado com o seguinte formato: "V: xxx.xx, N: xxx, L: xxx.xx", onde V é o valor do ingresso, N é a quantidade de ingressos vendidos e L o lucro obtido. Ao final, o programa deve apresentar um resumo contendo três linhas com o seguinte formato:
"Melhor valor final: xxx.xx"
"Lucro: xxx.xx"
"Numero de ingressos: xx"

Observações

Todos os valores reais devem ser apresentados com 2 casas decimais. Caso o intervalo de valores indicados não produza lucro positivo, os valores que devem aparecer no resumo devem assumir o valor zero.

Exemplo

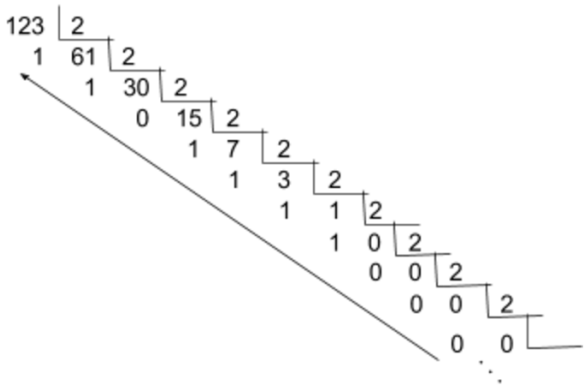
Entrada
5 2 8
Saída
V: 2.00, N: 270, L: 326.50 V: 3.00, N: 220, L: 449.00 V: 4.00, N: 170, L: 471.50 V: 5.00, N: 120, L: 394.00 V: 6.00, N: 60, L: 157.00 V: 7.00, N: 0, L: -200.00 V: 8.00, N: -60, L: -677.00 Melhor valor final: 4.00 Lucro: 471.50 Numero de ingressos: 170

Entrada
0.5 0 2
Saída
V: 0.00, N: 145, L: -207.25 V: 1.00, N: 90, L: -114.50 V: 2.00, N: 30, L: -141.50 Melhor valor final: 0.00 Lucro: 0.00 Numero de ingressos: 0

4 Conversão de decimal para binário

 (+++)

Escreva um algoritmo em Linguagem C que leia um número $0 \leq n \leq 255$ na base decimal e apresente sua representação em binário. Caso o número informado não esteja no intervalo especificado, o programa deve finalizar imprimindo a mensagem “Numero invalido!” na tela. A transformação de um número na base decimal para binária é obtida pela sequência de divisões por 2. O número 123, por exemplo, tem sua representação binária 01111011 porque:



Não é permitido o uso de outras bibliotecas além da stdio.h.

Entrada

O programa deve ler um número inteiro qualquer.

Saída

Caso o número lido esteja fora do intervalo especificado, o programa deve imprimir a mensagem "Numero invalido!"e encerrar. Caso o número lido seja válido, o programa deve apresentar a representação binária de n na tela.

Observações

Neste problema, todos os números binários deverão conter 8 bits. O número zero (em decimal), por exemplo, tem sua representação binária 00000000. O número 1 = 00000001, o 2 = 000000010 e assim por diante.

Exemplo

Entrada
0
Saída
00000000

Entrada
123
Saída
01111011

Entrada
128
Saída
10000000

5 Grãos de milho no tabuleiro de xadrez

 (+++)

(Adaptado de FARRER, 1999) Faça um algoritmo em linguagem C que calcule e escreva o número de grãos de milho que se pode colocar em um tabuleiro de xadrez, colocando n no primeiro quadro e nos quadros seguintes o dobro de n, caso o quadro seja escuro, e a mesma quantidade de n, caso o quadro seja branco. Percorra o tabuleiro sempre da esquerda para a direita e de baixo para cima. A Figura 1 apresenta um tabuleiro de xadrez típico.

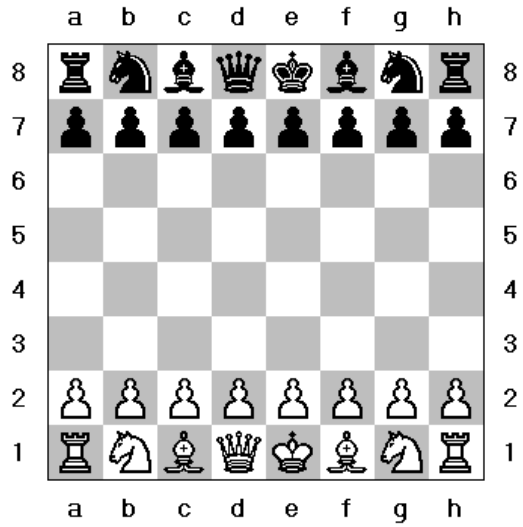


Figura 1: Tabuleiro de xadrez.

Entrada

O programa deve ler uma linha contendo um número inteiro n.

Saída

O programa deve apresentar uma linha contendo a quantidade de grãos que podem ser colocados no tabuleiro.

Exemplo

Entrada
6
Saída
570

Entrada
11
Saída
1045

6 Cálculo do imposto de renda

 (+++)

Desenvolver um algoritmo que determine o imposto de renda cobrado de um funcionário pelo governo. Seu programa deverá ler a matrícula de um funcionário, o valor do salário mínimo, o número de dependentes, o salário do funcionário e a taxa de imposto normal que já foi paga pelo funcionário. O imposto bruto é:

- 20% do salário do funcionário, se o funcionário ganha mais de 12 salários mínimos;
- 8% do salário do funcionário, se o funcionário ganha mais de cinco salários mínimos;
- Zero % do salário do funcionário, se ele ganha cinco salários mínimos ou menos.

Determine o imposto líquido a ser pago pelo funcionário subtraindo R$300,00$ no imposto bruto, para cada dependente do funcionário. O programa calculará e imprimirá o imposto a ser pago ou devolvido, que é a diferença entre o imposto líquido e o imposto normal **descontado** do salário do funcionário. Se a diferença for negativa, o programa deve emitir a mensagem de "imposto a receber". Se a diferença for um valor positivo o programa deve emitir a mensagem, "imposto a pagar", e, se for igual a zero, deve emitir a mensagem "imposto quitado".

Entrada

O programa deve ler uma linha contendo cinco valores na entrada, separados entre si por um espaço: a matrícula (um número inteiro), o valor do salário mínimo (float), o número de dependentes (inteiro), o salário do funcionário (float) e a taxa de imposto (float), nesta ordem.

Saída

O programa deve imprimir quatro linhas, contendo o seguinte:

- MATRICULA = u, onde u é o valor da matrícula do funcionário;
- IMPOSTO BRUTO = v, onde v é o valor do imposto bruto;
- IMPOSTO LIQUIDO = x, onde x é o valor do imposto líquido;
- RESULTADO = w, onde w é o valor da diferença entre o imposto bruto e o imposto líquido;
- A mensagem IMPOSTO A RECEBER, se o valor de w for negativo ou a mensagem IMPOSTO QUITADO, se w for igual a zero, ou a mensagem IMPOSTO A PAGAR, caso w for maior que zero.

Os valores de v,x e w devem conter duas casas decimais.

Exemplo

Abaixo são mostrados dois exemplos de entrada e saída, mas há apenas um caso de entrada (uma linha) para esse programa.

Entrada
`99123 510.0 3 1531.97 8.5`
Saída

`MATRICULA = 99123`
`IMPOSTO BRUTO = 0.00`
`IMPOSTO LIQUIDO = -900.00`
`RESULTADO = -1030.22`
`IMPOSTO A RECEBER`

Entrada
56789 630.00 2 4567.01 56.7
Saída
MATRICULA = 56789 IMPOSTO BRUTO = 365.36 IMPOSTO LIQUIDO = -234.64 RESULTADO = -2824.13 IMPOSTO A RECEBER

7 Índices da matriz inferior

 (+++)

Faça um algoritmo em linguagem C que apresente os pares de índices inferiores à diagonal principal de uma matriz $m \times n$. A diagonal principal corresponde aos elementos $a_{i,i}$.

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \tag{2}$$

Entrada

O programa deve ler as dimensões m e n da matriz, onde m é o número de linhas e n o número de colunas.

Saída

O programa deve apresentar em cada linha os pares de índices de uma mesma linha. Os pares devem ser apresentados entre parênteses e separados por um hífen.

Exemplo

Entrada
3 3
Saída
(2,1) (3,1)-(3,2)

Entrada
6 3
Saída
(2,1) (3,1)-(3,2) (4,1)-(4,2)-(4,3) (5,1)-(5,2)-(5,3) (6,1)-(6,2)-(6,3)

Entrada
5 2
Saída
(2,1) (3,1)-(3,2) (4,1)-(4,2) (5,1)-(5,2)

8 Hipotenusas inteiras (+++)

 (+++)

(IME-USP) Dado um número inteiro positivo n, determinar todos os inteiros entre 1 e n que são comprimento da hipotenusa de um triângulo retângulo com catetos inteiros. Para cada valor de hipotenusa válido no intervalo de 1 a n, imprimir todos os pares de catetos que formam um triângulo retângulo distinto com aquele valor de hipotenusa.

Entrada

O programa deve ler um valor inteiro n maior que zero.

Saída

O programa deve apresentar uma linha com o texto: "hipotenusa = h, catetos c_1 e c_2", onde h é uma hipotenusa inteira, c_1 e c_2 são seus catetos inteiros, de modo que $c_1 \leq c_2$. No caso de haver mais de um par de catetos válidos para um mesmo valor de hipotenusa, por exemplo $(c1, c2), (c3, c4), \ldots (c_k, c_{k+1})$, imprima os pares de tal modo que o valor do primeiro cateto seja menor ou igual ao valor do segundo cateto de um mesmo par e que o valor do primeiro cateto de um par seja menor que o valor do primeiro cateto do par de subsequente. Por exemplo, para um valor de hipotenusa igual a 85, existem os seguintes pares de catetos: $(13, 84)$, $(40, 75)$, $(36, 77) e (51, 68)$. Nesse caso a saída deve ser a seguinte:

hipotenusa = 85, catetos 13 e 84
hipotenusa = 85, catetos 36 e 77
hipotenusa = 85, catetos 40 e 75
hipotenusa = 85, catetos 51 e 68

Exemplo

Entrada
5
Saída
hipotenusa = 5, catetos 3 e 4

Entrada
15
Saída
hipotenusa = 5, catetos 3 e 4 hipotenusa = 10, catetos 6 e 8 hipotenusa = 13, catetos 5 e 12 hipotenusa = 15, catetos 9 e 12

9 Lucro de Mercadorias

 (+++)

Um comerciante deseja fazer um levantamento do lucro das mercadorias que ele comercializa. Para isto, mandou digitar uma linha contendo para cada mercadoria, os seguintes dados:

- O código da mercadoria (unsigned long int).
- O preço de compra da mercadoria (float).
- O preço de venda da mercadora (float).
- O número de vendas da mercadoria (int).

Escreva um programa que leia uma quantidade indefinida de mercadorias e que faça o seguinte:

1. Determine a quantidade de Mercadorias que geraram lucro menor que 10%.
2. Determine a quantidade de Mercadorias que geraram lucro maior ou igual a 10% e menor ou igual a 20%.
3. Determine a quantidade de Mercadorias que geraram lucro maior que 20%.
4. Imprima o código da mercadoria que gerou maior lucro.
5. Imprima o código da mercadoria mais vendida.
6. Determine e escreva o valor total de compra e de venda de todas as mercadorias, assim como o lucro total.

Entrada

A entrada contém várias linhas, cada uma contendo quatro valores separados entre si por um espaço. O primeiro valor é um número inteiro que corresponde ao código de uma mercadoria, o segundo valor é o preço de compra de uma mercadoria, o terceiro, é o valor do preço de venda e o quarto, o número de unidades da mercadoria que foram vendidas.

Saída

O programa deve gerar seis linhas na saída. A primeira delas contém a frase: "Quantidade de mercadorias que geraram lucro menor que 10%: r", onde r é um número inteiro. A segunda linha contém a frase: "Quantidade de mercadorias que geraram lucro maior ou igual a 10% e menor ou igual a 20%: s", onde s é um número inteiro. A terceira linha contém a frase: "Quantidade de mercadorias que geraram lucro maior do que 20%: t", onde t é um número inteiro. A quarta linha contém a frase "Codigo da mercadoria que gerou maior lucro: u", onde u é um número inteiro. A quinta linha contem a frase: "Codigo da mercadoria mais vendida: v" onde v é um número inteiro. A sexta linha Contém a frase: 'Valor total de compras: x, valor total de vendas: y e percentual de lucro total: $z\%$", onde x, y e z são valores reais com duas casas decimais. Após os valores de r, s, t, u, v e o valor z o programa deve imprimir o caractere de quebra de linha.

Exemplo

Entrada
4448901 20.00 25.79 200 4448902 13.99 17.99 150 4448903 5.50 5.90 2000 4448904 33.50 37.90 100
Saída
Quantidade de mercadorias que geraram lucro menor que 10%: 1 Quantidade de mercadorias que geraram lucro maior ou igual a 10% e menor ou igual a 20%: 1 Quantidade de mercadorias que geraram lucro maior do que 20%: 2 Codigo da mercadoria que gerou maior lucro: 4448901 Codigo da mercadoria mais vendida: 4448903 Valor total de compras: 20448.50, valor total de vendas: 23446.50 e percentual de lucro total: 14.66%

10 N ao cubo

 (+++)

(IME-USP) Sabe-se que um número da forma n^3 é igual a soma de n ímpares consecutivos.

Exemplo: $1^3 = 1$, $2^3 = 3+5$, $3^3 = 7+9+11$ e $4^3 = 13+15+17+19$. Dado m, determine os ímpares consecutivos cuja soma é igual a n^3 para n assumindo valores de 1 a m.

Entrada

O programa deve ler um número inteiro maior que zero.

Saída

O programa deve apresentar m linhas com a seguinte mensagem: "$k * k * k = x_1 + x_2 + \ldots + x_k$", onde $k = 1, 2, \ldots, m$ e x_i é a sequência de números ímpares consecutos.

Exemplo

Entrada
`4`
Saída

`1*1*1 = 1`
`2*2*2 = 3+5`
`3*3*3 = 7+9+11`
`4*4*4 = 13+15+17+19`

11 Número Invertido

 (+++)

Escreva um programa para ler um número de três dígitos e imprimir o número invertido.

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

12 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, apresenta a soma dos divisores de n e verifica se o número informado é perfeito ou não.

Entrada

O programa deve ler um número inteiro n.

Saída

O programa deve apresentar uma linha contendo o texto: "n = d1 + d2 + d3 + ... + dk = 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)`

13 Ordem

 (+++)

Você receberá três valores inteiros e deve descobrir quais deles correspondem às variáveis a, b e c. Os números não serão dados em ordem exata, mas sabemos que o valor correspondente a a é menor do que o valor correspondente a b, e que o valor correspondente a b é menor do que o correspondente a c . Será informada a você a ordem em que os valores associados a cada variável devem ser impressos. Escreva um programa que imprima os valores na ordem requisitada.

Entrada

A entrada conterá duas linhas. A primeira, com três números inteiros positivos, separados entre si por um espaço. Todos os três números são inferiores ou iguais a 100. A segunda linha conterá três letras maiúsculas A, B e C (sem espaços entre elas) representando a ordem desejada de impressão dos valores das variáveis.

Saída

A saída deve conter, numa linha, os inteiros a, b e c na ordem desejada, separados por espaços simples. Após o último número da saída deve aparecer apenas o caractere de quebra de linha: '\n'.

Observações

Após o último número na primeira linha da entrada, está no buffer de entrada o caractere '\n'. Com isso ao tentar ler o primeiro caractere (A, B, ou C) na segunda linha de entrada com `scanf("%d", &x);` será lido o caractere '\n' na variável `x`, ao invés de uma das letras na entrada (A, B, ou C). Para evitar isso, você pode fazer com que a leitura do último número na primeira linha consuma o caractere '\n' da primeira linha, colocando esse caractere na especificação de formato do `scanf()`. Por exemplo, suponha que você declarou as seguintes variáveis na entrada: `int a, b,c;` para armazenar os três número da primeira linha e `char x, y, z;`, para armazenar as três letras que aparecem na segunda linha de entrada. A leitura dessas variáveis de entrada pode ser realizada assim: `scanf("%d %d %d\n", &a, &b, &c); scanf(%c%c%c", &x, &y, &z);` Repare o '\n' ao final da formatação do primeiro `scanf` e repare que não há espaços entre os "%c"na formatação do segundo `scanf`. O '\n'. ao final da formatação do primeiro `scanf()` faz com que o caractere de quebra de linha seja consumido no buffer. Asssim, no segundo `scanf()` será armazenada na variável `x` a primeira letra da segunda linha e não o '\n', resolvendo o problema da leitura.

Exemplo

Entrada
1 5 3 A B C
Saída
1 3 5

Entrada
6 4 2 C A B
Saída
6 2 4

14 Ordena 4 números

 (+++)

Escreva um algoritmo que leia 4 números reais em qualquer ordem e os apresente de forma ordenada na tela.

Entrada

O programa deve ler 4 valores reais.

Saída

O programa deve imprimir uma linha contento a lista ordenada de números separados por vírgula e espaço, cada número com 2 casas decimais.

Exemplo

Entrada

`3.0`
`1`
`3.1`
`8`

Saída
`1.00, 3.00, 3.10, 8.00`

15 Sistemas de Equação Linear

 (+++)

Dado um sistema de equações lineares do tipo:

$$ax + by = c$$
$$dx + ey = f$$

Escreva um programa para ler os valores dos coeficientes: a, b, c, d, e e f e calcular os valores de x e y.

Entrada

O programa deve ler os valores de a, b, c, d, e, f nesta ordem, um valor por linha. Os valores são números reais (float).

Saída

O programa deve imprimir uma linha contendo a frase: O VALOR DE X E = z, onde z é o valor da variável x, escrito com duas casas decimais. O programa deve imprimir uma segunda linha contendo a frase: O VALOR DE Y E = w, onde w corresponde ao valor da variável y escrito com duas casas decimais. Ao final da segunda linha o programa deve imprimir um caractere de quebra de linha: '\n'.

Exemplo

Entrada
7 8 12 3 5 9
Saída
O VALOR DE X E = -1.09 O VALOR DE Y E = 2.45

16 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

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 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`

17 Valor em Notas e Moedas

 (+++)

Escreva um algoritmo par 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.

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`

18 Várias Ordenações

 (+++) Escrever um programa que leia um conjunto de 4 valores: i, a, b e c, onde i é um valor inteiro e positivo e a, b e c são quaisquer valores reais. O programa deve imprimir os valores de a, b, c na ordem indicada pelo valor de i, conforme explicitado a seguir:

- Se i = 1 escrever os três valores a, b, c em ordem crescente.
- Se i = 2 escrever os três valores a, b, c em ordem decrescente.
- Se i = 3 escrever os três valores a, b, c de forma que o maior número entre a, b, c fique no meio dos outros dois números e e o menor fique por último.

Entrada

O programa deve ler uma linha com um número inteiro na entrada e outras três linhas, cada uma contendo um valor real (float)

Saída

O programa deve imprimir uma linha contendo os três números reais, na ordem indicada pela primeira linha da entrada. Os três números devem possuir duas casas decimais e devem estar separados entre si por um espaço. O último número a ser impresso deve ser seguido imediatamente por um caractere de quebra de linha.

Exemplo

Entrada
3 80.0 36.9 -99.3
Saída
36.90 80.00 -99.30

Entrada
1 34.2 34.2 34.2
Saída
34.20 34.20 34.20

Entrada
2 12.3 12.3 12.3
Saída
12.30 12.30 12.30

Entrada
2 0.0 -0.4 89.0
Saída
89.00 0.00 -0.40

19 Transforma decimal em fração

 (+++)

Faça um programa que leia um número decimal e o converta para sua representação em fração simplificada.

Entrada

O programa deve ler um número real N.

Saída

O programa deve apresentar uma linha contendo a fração simplificada, correspondente ao número N informado. A fração deve ser apresentada no formato **num/den**, onde **num** e **den** são o numerador e o denominador respectivamente.

Exemplo

Entrada
`12.05`
Saída
`241/20`

20 Procura por número amigo

 (++++)

Números amigos são números onde cada um deles é a soma dos divisores do outro. Por exemplo, o par (220,284) são números amigos porque a soma dos divisores de 220 (1, 2, 4, 5, 10, 11, 20, 22, 44, 55 e 110) é igual a 284 e a soma dos divisores de 284 (1, 2, 4, 71 e 142) é igual a 220. Faça um programa que encontre os n primeiros números amigos do conjunto dos números naturais. O programa deve encontrar somente números amigos diferentes. Por exemplo, o par (220,284) tem o par de números amigos correspondente (284,220), no entanto, o par é formado pelos mesmos números. O programa deve apresentar somente o primeiro par (220,284), de modo que o primeiro número amigo sempre é menor que o segundo.

Entrada

O programa deve ser um número inteiro positivo n.

Saída

Os pares de números devem ser apresentados em linhas separadas, entre parênteses, separados pos vírgula e sem espaços entre si. Ex: "(x,y)".

Observações

A procura por números amigos pode demorar muito tempo. Limite seus testes para $n < 9$.

Exemplo

Entrada
`2`
Saída
`(220,284)` `(1184,1210)`

21 Série de Taylor para a função cosseno

 (++++)

Escreva um programa que dado um número real x e a quantidade de termos N, calcule o valor da função $\cos(x)$, a partir da série:

$$\cos(x) = \sum_{n=0}^{N} \frac{(-1)^n x^{2n}}{(2n)!} = \frac{x^0}{0!} - \frac{x^2}{2!} + \frac{x^4}{4!} - \ldots + \frac{(-1)^N x^{2N}}{(2N)!} \tag{3}$$

, onde x é o ângulo em radianos e N a quantidade de termos da série menos 1.

Entrada

O programa deve ler o valor de x e N.

Saída

O programa deve apresentar uma linha contendo o texto "cos(x) = y\n", onde x é o ângulo fornecido pelo usuário e y o seno do ângulo. x deve ser impresso com 2 casas decimais e y com 6 casas decimais.

Observações

Neste tipo de problema, a quantidade de termos pode gerar números muito grandes por conta da operação de fatorial e potenciação de x. Atente-se aos tipos de dados usados nas declarações das variáveis e não use valores de N maiores que 9. Lembre-se que um ângulo qualquer sempre pode ser representado por um valor entre 0 e 2π. Use a constante M_PI da biblioteca <math.h>. Como sugestão de desafio à solução do problema, tente escrever um algoritmo que use apenas um laço de repetição.

Exemplo

Entrada
2 9
Saída
cos(2.00) = -0.416147

Entrada
3.14 6
Saída
cos(3.14) = -0.999899

Entrada
1 4
Saída
cos(1.00) = 0.540303

22 Série de Taylor para a função e^x

 (++++)

Escreva um programa que dado um número real x e a quantidade de termos N, calcule o valor da função e^x, a partir da série:

$$e^x = \sum_{n=0}^{N} \frac{x^n}{(n)!} = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \ldots + \frac{x^N}{(N)!} \tag{4}$$

, onde x é o expoente da função e N a quantidade de termos da série menos 1.

Entrada

O programa deve ler o valor de x e N.

Saída

O programa deve apresentar uma linha contendo o texto "e^x = y\n", onde x é o expoente fornecido pelo usuário e y o valor da função. x deve ser impresso com 2 casas decimais e y com 6 casas decimais.

Observações

Neste tipo de problema, a quantidade de termos pode gerar números muito grandes por conta da operação de fatorial e potenciação de x. Atente-se aos tipos de dados usados nas declarações das variáveis e não use valores de N maiores que 9. Como sugestão de desafio à solução do problema, tente escrever um algoritmo que use apenas um laço de repetição.

Exemplo

Entrada
2 9
Saída
e^2.00 = 7.388713

Entrada
3.14 6
Saída
e^3.14 = 22.155058

Entrada
1 9
Saída
e^1.00 = 2.718282

23 Série de Taylor para a função seno

 (++++)

Escreva um programa que dado um número real x e a quantidade de termos N, calcule o valor da função $\sin(x)$, a partir da série:

$$\sin(x) = \sum_{n=0}^{N} \frac{(-1)^n x^{2n+1}}{(2n+1)!} = \frac{x^1}{0!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots + \frac{(-1)^N x^{2N+1}}{(2N+1)!} \tag{5}$$

, onde x é o ângulo em radianos e N a quantidade de termos da série menos 1.

Entrada

O programa deve ler o valor de x e N.

Saída

O programa deve apresentar uma linha contendo o texto "seno(x) = y\n", onde x é o ângulo fornecido pelo usuário e y o seno do ângulo. x deve ser impresso com 2 casas decimais e y com 6 casas decimais.

Observações

Neste tipo de problema, a quantidade de termos pode gerar números muito grandes por conta da operação de fatorial e potenciação de x. Atente-se aos tipos de dados usados nas declarações das variáveis e não use valores de N maiores que 9. Lembre-se que um ângulo qualquer sempre pode ser representado por um valor entre 0 e 2π. Use a constante M_PI da biblioteca <math.h>. Como sugestão de desafio à solução do problema, tente escrever um algoritmo que use apenas um laço de repetição.

Exemplo

Entrada
2 9
Saída
seno(2.00) = 0.909297

Entrada
3.14 6
Saída
seno(3.14) = 0.001614

Entrada
1 4
Saída
seno(1.00) = 0.841471

24 Decomposição em fatores primos

 (+++++)

Todo número natural maior que 1 pode ser escrito na forma de uma multiplicação em que todos os fatores são números primos. Por exemplo, o número 36 pode ser representado pela multiplicação $2 \times 2 \times 3 \times 3$. A essa representação multiplicativa dá-se o nome de Decomposição em Fatores Primos ou Fatoração, que é um produto de fatores primos. O processo de fatoração de N segue um método prático de divisões sucessivas pelo seu menor fator primo. A cada passo, deve-se encontrar o menor divisor primo do quociente da divisão anterior. A Figura 2 mostra dois exemplos d efatoração em números primos.

Faça um programa que leia um número inteiro maior que 1 e apresente sua fatoração em números primos. Uma vez executado, o programa deve sempre apresentar uma fatoração. Caso o número lido seja inválido, o programa deve lê-lo novamente.

36	2
18	2
9	3
3	3
1	

120	2
60	2
30	2
15	3
5	5
1	

Figura 2: Exemplo de fatoração dos números 36 e 120.

Entrada

O programa deve ler um número inteiro N.

Saída

O programa deve apresentar a mensagem "Fatoracao nao e possivel para o numero x!"sempre que o número lido não é válido. Caso o número lido seja válido, então o programa deve apresentar sua fatoração no seguinte formato: `N = f1 x f2 x ... x fk.`

Exemplo

Entrada
`554`
Saída
`554 = 2 x 277`

Entrada

`-1`
`0`
`120`

Saída

`Fatoracao nao e possivel para o numero -1!`
`Fatoracao nao e possivel para o numero 0!`
`120 = 2 x 2 x 2 x 3 x 5`

Entrada
2
Saída
2 = 2