Universidade Federal de Goiás
Instituto de Informática
Introdução à Programação
Curso de Engenharia de Software - 2024-1
Lista 2

Prof. Dr. Jacson Rodrigues Barbosa
Prof. Dr. Thierson Couto Rosa

Pontuação

Sumário

1 Turma de Introdução à Programação (+)

 (+)

A disciplina de Introdução à Programação possui oito provas, cinco listas de exercícios e uma nota de trabalho final. Para que um aluno seja aprovado por nota na disciplina, ele deve obter uma nota final maior ou igual a seis. A nota final é computada pela seguinte fórmula:

$$NF = 0.7 \cdot MP + 0.15 \cdot ML + 0.15 \cdot NT \tag{1}$$

onde MP é a média aritmética das notas de prova, ML é a média aritmética das notas das cinco listas e NT é a nota do trabalho final.

Para ser aprovado na disciplina o aluno deve ter presença igual a ou superior a 75% da carga horária da disciplina que no caso de Introdução à Programação é 128 horas.

Escreva um programa para ler as notas de cada aluno de uma turma, computar a nota final do aluno e imprimir a nota final e uma indicação da situação final do aluno. Essa indicação pode ser uma das seguintes alternativas:

- Aprovado - se o aluno teve $NF \geq 6$ e presença superior à quantidade de horas mínima.
- Reprovado por nota - se o aluno teve a presença minima, mas sua nota NF não é suficiente para ser aprovado.
- Reprovado por frequência insuficiente - o aluno obteve nota NF superior ou igual a seis mas sua presença às aulas não foi suficiente para ser aprovado.
- Reprovado por frequência e por nota - o aluno não alcançou o valor mínimo de NF e também não tem frequência mínima para aprovação.

Entrada

A entrada contém várias linhas, cada uma contendo os dados de um aluno separados entre si por um espaço. O primeiro valor em uma linha corresponde à matrícula do aluno (um valor inteiro sem sinal). Os próximos oito valores seguintes correspondem às notas das oito provas. Os seguintes cinco valores correspondem às notas obtidas nas listas de exercícios. O penúltimo valor corresponde a nota do trabalho final e o último valor em uma linha corresponde à presença do aluno. A última linha da entrada contém todos os valores iguais a -1 essa linha serve apenas para indicar o fim da entrada e não deve ser processada.

Saída

O programa deve gerar uma linha para cada aluno contendo a seguinte frase: “Matricula: m, Nota Final: n, Situação Final: s”. O valor de m corresponde à matricula de um aluno, o valor de n corresponde ao valor da nota final (NF) do aluno e s é uma das seguintes frases correspondendo à situação final do aluno:

- APROVADO
- REPROVADO POR FREQUENCIA
- REPROVADO POR NOTA
- REPROVADO POR NOTA E POR FREQUENCIA

Exemplo

Entrada
4448901 2.3 3.2 4.3 5.0 6.5 7.2 7.3 8.4 9.2 8.3 9.5 7.6 10.0 9.0 118
4448902 4.5 8.2 4.4 7.0 7.5 9.2 8.3 9.5 10.0 9.2 8.3 9.5 7.6 10.0 80
4448903 4.1 3.2 4.4 5.0 6.5 6.2 5.5 6.5 8.4 7.2 6.3 9.5 5.6 4.0 110
4448903 4.1 3.2 4.4 5.0 6.5 6.2 5.5 6.5 8.4 7.2 6.3 9.5 5.6 4.0 80
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
Saída
Matricula: 4448901, Nota Final: 6.56, Situacao Final: APROVADO
Matricula: 4448902, Nota Final: 7.97, Situacao Final: REPROVADO POR FREQUENCIA
Matricula: 4448903, Nota Final: 5.33, Situacao Final: REPROVADO POR NOTA
Matricula: 4448903, Nota Final: 5.33, Situacao Final: REPROVADO POR NOTA E POR FREQUENCIA

2 Ultrapassagem populacional (+)

 (+)

Supondo que a população de um país A seja de a habitantes com uma taxa anual de crescimento de 3% e que a população de um país B seja de b habitantes, com uma taxa anual de crescimento de 1,5%, fazer um algoritmo que calcule e escreva o número de anos necessários para que a população do país A ultrapasse ou iguale a população do país B, mantidas essas taxas de crescimento.

Entrada

O programa deverá ler duas linhas de entrada, cada uma contendo um número inteiro positivo representando a população de um país. O valor na primeira linha corresponde ao número de habitantes do país A e será sempre menor que o valor na segunda linha, o qual corresponde ao número de habitantes do país B.

Saída

A saída deve conter, numa linha com a frase ANOS = x, onde x é um valor em anos e deve ser seguido por um caractere de quebra de linha: '\n'.

Exemplo

A seguir são mostrados dois casos distintos de entrada, somente para efeito de ilustração, porém, esse problema contém apenas um caso de teste na entrada, formado pelas duas linhas de entrada descritas acima.

Entrada
90000000 200000000
Saída
ANOS = 55

3 Fatorial (++)

 (++)

Dado um número inteiro n, calcule seu fatorial $n!$. O fatorial de um número é dado pela equação: $n! = n(n-1)(n-2)\ldots 1$. Por definição, $0! = 1$.

Entrada

O programa deve ler um número inteiro n.

Saída

O programa deve apresentar uma linha com a mensagem: "n! = f", onde n é o número lido e f o seu fatorial.

Observações

O fatorial de um número é resultado de uma operação de produtório que pode levar a valores incrivelmente grandes. Lembre-se de usar tipos de dados apropriados ao problema proposto.

Exemplo

Entrada
`2`
Saída
`2! = 2`

Entrada
`4`
Saída
`4! = 24`

4 Gerador de tabuada (++)

 (++)

Escreva um programa em linguagem Go que leia um número qualquer n de 0 a 9 e imprima na tela a tabuada de soma, subtração, multiplicação e divisão desse número para o K valores, iniciando em i em incrementos de s.

Observação

Utilize apenas o tipo `double` para armazenar valores reais.

Entrada

O programa deve ler quatro números quaisquer n, i, K e s.

Saída

O programa deve apresentar, em sequência, a tabuada de soma, subtração, multiplicação e divisão, com o texto: "Tabuada de soma:", "Tabuada de subtracao:", "Tabuada de multiplicacao:"e "Tabuada de divisao:"antes de cada tabuada. Cada linha da tabuada segue o formado: "n op B = R", onde n é o número lido, B é o segundo termo da tabuada, op é o operador da tabuada e R o resultado da operação. Os números devem ser apresentados com 2 casas decimais.

Exemplo

Entrada
3 1 2 0.1
Saída
Tabuada de soma: 3.00 + 1.00 = 4.00 3.00 + 1.10 = 4.10 Tabuada de subtracao: 3.00 - 1.00 = 2.00 3.00 - 1.10 = 1.90 Tabuada de multiplicacao: 3.00 x 1.00 = 3.00 3.00 x 1.10 = 3.30 Tabuada de divisao: 3.00 / 1.00 = 3.00 3.00 / 1.10 = 2.73

5 Maior segmento crescente de uma sequência (++)

(++) (POLI 89) Dados n e uma seqüência de n números inteiros, determinar o comprimento máximo de um segmento crescente na sequência. O comprimento do segmento é dado pelo número de elementos do segmento menos um.

Entrada

O programa deve ler um número inteiro n maior que zero e uma sequência de n números inteiros. Os números podem aparecer em qualquer ordem na sequência.

Saída

O programa deve apresentar a mensagem "O comprimento do segmento crescente maximo e: k", onde k é o tamanho do maior segmento crescente encontrado. A mensagem deve ser seguida por um caractere de quebra de linha.

Exemplo

Entrada
9 5 10 3 2 4 7 9 8 5
Saída
O comprimento do segmento crescente maximo e: 3

Entrada
5 10 8 7 5 2
Saída
O comprimento do segmento crescente maximo e: 0

Entrada
7 1 2 0 40 34 35 16
Saída
O comprimento do segmento crescente maximo e: 1

6 Maior segmento igual de uma sequência (++)

(++) (POLI 87) Dados n e uma sequência de n números inteiros, determinar a maior subsequência de números iguais consecutivos na sequência lida e imprimir o número de elementos iguais nessa subsequência.

Entrada

O programa deve ler um número inteiro maior que zero n e uma sequência de n números inteiros que podem aparecer em qualquer ordem.

Saída

O programa deve apresentar a mensagem "A maior subsequência de elementos iguais possui k elementos.". A frase deve terminar com um caractere de quebra de linha.

Exemplo

Entrada
`2` `1 1`
Saída
`A maior subsequência de elementos iguais possui 2 elementos.`

Entrada
`7` `0 0 2 5 5 5 6`
Saída
`A maior subsequência de elementos iguais possui 3 elementos.`

7 Média de pares e ímpares (++)

 (++)

Faça um programa que leia uma sequência de números inteiros diferentes de zero e que apresente a média nos números pares e a média nos números ímpares que formam a sequência.

Entrada

O programa deve ler uma sequência de números inteiros diferentes de zero. A entrada termina quando o valor zero for lido. O zero não deve ser processado e apenas indica fim da entrada de dados.

Saída

O programa deve apresentar duas linhas, a primeira contendo a mensagem: "MEDIA PAR = *mp*"e a segunda com a mensagem: "MEDIA IMPAR = *mi*", onde *mp* e *mi* são os valores das médias dos números pares e ímpares respectivamente, escritos com até duas casas decimais.

Exemplo

Entrada	Saída
1 5 8 7 6 3 0	MEDIA PAR: 7.00 MEDIA IMPAR: 4.00

8 Número de finais (++)

 (++)

Em um campeonato de futebol os times são nomeados como Time1, Time2, ..., TimeN. A organização do campeonato deseja saber quais são as finais possíveis dado a quantidade *N* de times. Para resolver esse problema, você foi contratado para fazer um programa de computador que, dada a quantidade *N* de times, imprima todas as configurações possíveis de finais.

Entrada

O programa deve ler um número *N*, inteiro e positivo, referente à quantidade de times do campeonato.

Saída

O programa deve apresentar na tela a sequência de finais com cada linha no formato: Final *k*: Time*i* X Time*j*, onde *k* é um contador de finais, *i* e *j* são as denominações de cada time. Caso o número de times informado for menor que 2, então o programa deve imprimir a mensagem: "Campeonato invalido!".

Exemplo

Entrada
`3`
Saída

`Final 1:  Time1 X Time2`
`Final 2:  Time1 X Time3`
`Final 3:  Time2 X Time3`

Entrada
`1`
Saída
`Campeonato invalido!`

9 Número primo (++)

 (++)

Faça um programa que leia um número *N* e informa se o número é primo ou não.

Entrada

O programa deverá ler um número inteiro *N* positivo.

Saída

O programa deverá apresentar a mensagem "PRIMO"caso *N* seja primo e "NAO PRIMO"caso contrario. Caso o valor de *N* não seja um número inteiro positivo, o programa deve apresentar a mensagem "Numero invalido!".

Exemplo

Entrada
7
Saída
PRIMO

Entrada
9
Saída
NAO PRIMO

10 Salário (++)

 (++)

Escreva um programa que leia várias linhas contendo cada uma a matrícula de um funcionário (um valor inteiro), seu número de horas trabalhadas, o valor que recebe por hora e calcule o salário desse funcionário. A seguir, mostre o número e o salário do funcionário, com duas casas decimais.

Entrada

A entrada é formada por varias linhas, cada uma contendo três valores decimais separados um do outro por um espaço em branco. O último valor na linha é seguido pelo caractere de quebra de linha (\n). A última linha contém uma matrícula igual a zero e não deve ser processada. Ela serve apenas para indicar ao programa o término da entrada de dados.

Saída

A saída deve conter para cada linha de entrada a matrícula, um espaço em branco e o salário calculado com duas casas decimais

Observação

- Para ler uma linha com os três valores, utilize a função **Scanf**: Scanf("%d %lf %lf", &A, &B, &C); e, em seguida, a função bufio.NewReader(os.Stdin).ReadByte(). Essa função é usada para consumir o caractere de quebra de linha na entrada.

Exemplo

Entrada
2015001 16000 3.23 2015002 16010 3.0 2015003 16009 2.99 2015004 15080 3.13 0 0 0.0
Saída
2015001 51680.00 2015002 48030.00 2015003 47866.91 2015004 47200.40

11 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:

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

Entrada

O programa deve ler um número **float** 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, erro:  0.250000000`
`r:  1.416666667, erro:  0.006944444`
`r:  1.414215686, erro:  0.000006007`

12 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"

Observação

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. Utilize apenas o tipo `float` para representar valores reais.

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

13 Grãos de milho no tabuleiro de xadrez (+++)

 (+++)

(Adaptado de FARRER, 1999) Faça um algoritmo em linguagem Go 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.

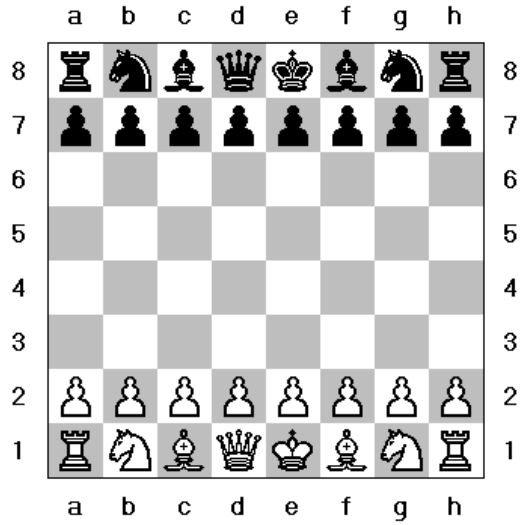

Figura 1: Tabuleiro de xadrez.

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

14 Índices da matriz inferior (+++)

 (+++)

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

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \tag{3}$$

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 ífen.

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)

15 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.

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

16 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

17 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 (double).
- O preço de venda da mercadora (double).
- 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%

18 Mínimo múltiplo comum (+++)

 (+++)

Faça um programa que calcule o Mínimo Múltiplo Comum (MMC) de 3 números inteiros. A Figura 18 apresenta um exemplo de cálculo de MMC.

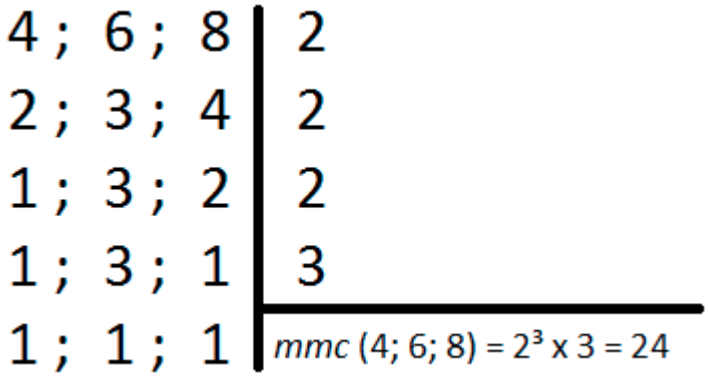

Figura 2: Exemplo de cálculo do MMC para os números 4, 6 e 8

Entrada

O programa deve ler 3 números inteirios diferentes de zero.

Saída

A saída é composta por várias linhas. O programa deve replicar a saída do procedimento da Figura 18 com expresso nos exemplos de entrada e saída. Cada linha deve ser impressa com o seguinte código: "%d %d %d :%d", onde o número 5 indica a quantidade mínima de espaços ou dígitos do número a ser apresentado.

Exemplo

Entrada	Saída
10 5 6	10 5 6 :2 5 5 3 :3 5 5 1 :5 MMC: 30

Entrada	Saída
3 5 7	3 5 7 :3 1 5 7 :5 1 1 7 :7 MMC: 105

19 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`

20 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)`

21 Sequência ordenada (+++)

 (+++)

Tia Zélia é uma professora muito dedicada. Ela está explicando as relações de ordem entre números para seus alunos. Na aula de hoje ela explicou a relação "menor que" representada pelo operador $<$. Ela explicou também sobre as propriedades dessa relação entre os números, incluindo a propriedade transitiva, isto é, se $x < y$ e $y < z$, então $x < z$. Tia Zélia quer fazer um treinamento com seus alunos. Ela quer propor o seguinte exercício aos seus queridos pupilos: apresentar a eles várias sequências de números e pedir que eles analisem cada sequência e que indiquem se ela está na ordem crescente. Para isso, tia Zélia quer sua ajuda. Ela vai editar um arquivo com várias seqüências de números; para cada sequência haverá duas linhas no arquivo: uma com um número inteiro indicando o tamanho da sequência e a linha imediatamente seguinte contém a sequência de valores reais propriamente dita, formada por números separados por um espaço, a menos do último valor que vem seguido diretamente por um caractere de quebra de linha. Ela quer gerar seqüências de números sem ter o trabalho de verificar se formam sequência ou não. Ela quer que um programa de computador faça isso para ela. Tia Zélia ficou sabendo que você é aluno de Introdução à Programação do INF-UFG e que os alunos dessa disciplina são exímios programadores! Portanto ela pede a você que faça um programa que resolva esse problema para ela.

Entrada

Para cada sequência numérica há na entrada duas linhas: uma com, apenas um valor inteiro, indica o número de valores reais que deve ocorrer na próxima linha. A linha seguinte contém tantos valores quanto indicado na linha anterior. Entre dois valores há apenas um espaço e após o último valor há um caractere de quebra de linha. A última linha da entrada contém um tamanho de sequência igual a zero e serve apenas para indicar término do processamento. Não há uma linha com sequência de valores após a ocorrência de uma linha com valor zero.

Saída

Para cada seqüência da entrada o seu programa deve emitir uma das seguintes respostas: ORDENADA, se a sequência estiver em ordem crescente de valores ou DESORDENADA, em caso contrário. Após cada palavra impressa deve haver apenas um caractere de quebra de linha.

Exemplo

Entrada
10 2.98 16.42 18.0 23.67 31.99 38.50 42.30 61.782000.00 2000.10 5 4.51 4.32 4.90 56.70 150.80 6 0.00 2.56 4.00 80.4 100.98 100.97 0
Saída
ORDENADA DESORDENADA DESORDENADA

22 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`

23 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)`

24 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:

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

, 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.

Observação

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 *math.pi* do pacote math. 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

25 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:

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

, 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.

Observação

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

26 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:

$$\sin(x) = \sum_{n=0}^{N} \frac{(-1)^n x^{2n+1}}{(2n+1)!} = \frac{x^1}{0!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \ldots + \frac{(-1)^N x^{2N+1}}{(2N+1)!} \tag{6}$$

, 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 *math.pi* do pacote math. 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

27 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 3 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 3: 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`. Em ambas as situações o programa deve terminar a impressão com um caractere de quebra de linha.

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