

Lista-parte-01

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

Sumário

1	Aprovado ou Reprovado (+)	2
2	Composição Inteira (+)	3
3	Conta de Água (+)	4
4	Conversão de temperatura (+)	5
5	Divisível por 3 e 5 (+)	6
6	Conversão de Nota em Conceito (+)	7
7	Série de pares (+)	8
8	Turma de Introdução à Programação (+)	9
9	Ultrapassagem populacional (+)	11
10	Dados (++)	12
11	Fatorial (++)	13
12	Maior segmento crescente de uma sequência (++)	14
13	Média de pares e ímpares (++)	15
14	Número primo (++)	16
15	Ordena 3 números (++)	17
16	Raízes de equações de grau 2 (++)	18
17	Soma dos 3 menores (++)	19
18	Transcrição de datas (++)	20
19	Classificação do Aço (+++)	21
20	Grãos de milho no tabuleiro de xadrez (+++)	22
21	José (+++)	23
22	Hipotenusas inteiras (+++)	24

23 Lucro de Mercadorias (+++)	25
24 Mínimo múltiplo comum (+++)	27
25 N ao cubo (+++)	28
26 Número palíndromo (+++)	29
27 Número perfeito (+++)	30
28 Ordem (+++)	31
29 Ordena 4 números (+++)	32
30 Triângulo de Pascal (++)	33
31 Professor de Fibonacci (+++)	34
32 Sequência intercalada (+++)	35
33 Calcular Dígito Verificador do Título de Eleitor (+++)	36
34 Transforma decimal em fração (+++)	39
35 Procura por número amigo (++++)	40
36 Série de Taylor para a função e^x (++++)	41
37 Decomposição em fatores primos (+++++)	42

1 Aprovado ou Reprovado (+)



(+)

Fazer um algoritmo que calcule a média aritmética das três notas de um aluno e mostre, além do valor da média, uma mensagem de "APROVADO", caso a média seja igual ou superior a seis, ou a mensagem "REPROVADO", caso contrário.

Entrada

A entrada conterá uma linha com as três notas do aluno, separadas entre si por um caractere de espaço.

Saída

A saída deve conter duas linhas. A primeira linha deve conter uma frase com o seguinte formato: $MEDIA = x$, onde x é o valor da média entre as três notas do aluno, contendo duas casas decimais. A segunda linha contém uma das duas mensagens: APROVADO ou REPROVADO. Após o valor da média e após a mensagem, o programa deve imprimir o caractere de quebra de linha: ‘\n’.

Exemplo

Entrada
7.4 6.2 3.7
Saída
MEDIA = 5.77
REPROVADO

Entrada
5.1 9.9 7.2
Saída
MEDIA = 7.40
APROVADO

2 Composição Inteira (+)



(+)

Escreva um algoritmo em Linguagem C que leia três números inteiros separados (n_1 , n_2 , n_3) e calcule o número inteiro correspondente à concatenação dos três números lidos, de modo que n_1 seja a centena, n_2 a dezena e n_3 a unidade. O programa deve apresentar o número calculado e também o seu quadrado. Caso n_1 , n_2 ou n_3 tenham mais que 1 dígito, o programa deve apresentar a mensagem: "DIGITO INVALIDO" e encerrar a execução. O valor de saída não deve ter zeros à esquerda.

Entrada

O programa deve ler 3 números inteiros.

Saída

O programa deve imprimir uma linha contendo o número resultado da composição dos três números inteiros e seu quadrado separados por vírgula e um espaço.

Exemplo

Entrada
1
2
3
Saída
123, 15129

Entrada
10
0
3
Saída
DIGITO INVALIDO

3 Conta de Água (+)



Desenvolver um programa para calcular a conta de água para uma empresa de saneamento. O custo da água varia dependendo se o consumidor é residencial, comercial ou industrial. A regra para calcular a conta é:

- Residencial: R\$ 5,00 de taxa mais R\$ 0,05 por metros cúbicos gastos;
- Comercial: R\$ 500,00 para os primeiros 80 metros cúbicos gastos mais R\$ 0,25 por metros cúbicos excedentes;
- Industrial: R\$ 800,00 para os primeiros 100 metros cúbicos gastos mais R\$ 0,04 por metros cúbicos excedentes;

O programa deverá ler a conta do cliente, o consumo de água por metros cúbicos e o tipo de consumidor (residencial, comercial e industrial). Como resultado, o programa deve imprimir a conta do cliente e o valor em Reais a ser pago pelo mesmo.

Entrada

O programa deverá ler uma linha na entrada contendo: a conta do cliente (um número inteiro), o consumo de água por metros cúbicos (float) e o tipo do consumidor (um caractere: 'C' - COMERCIAL, 'I' - INDUSTRIAL ou 'R' - RESIDENCIAL). Há um espaço entre os valores na linha de entrada

Saída

O programa deve imprimir duas linhas, contendo o seguinte:

- $CONTA = u$, onde u é o código inteiro identificador da conta;
- $VALOR DA CONTA = v$, onde v é o valor da conta com duas casas decimais, a ser pago pelo consumidor;

Após o valor impresso em cada linha, o programa deve imprimir o caractere de quebra de linha; '\n'. 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
39393939 230 C
Saída
CONTA = 39393939
VALOR DA CONTA = 537.50

Entrada
888 3752 I
Saída
CONTA = 888
VALOR DA CONTA = 946.08

4 Conversão de temperatura (+)



(+)

Escreva um programa que imprima uma tabela de conversão de graus Fahrenheit para graus Celsius. Dado um valor de temperatura F medida na escala Fahrenheit, seu valor equivalente C na escala Celsius é dado pela seguinte equação:

$$C = \frac{5(F - 32)}{9}$$

Entrada

A entrada conterá várias linhas. A primeira delas contém o número n de temperaturas em Fahrenheit a serem convertidas para Celsius. Cada uma das n linhas seguintes contém um valor real (**double**) com a medida de uma temperatura em graus Fahrenheit.

Saída

O programa deve imprimir n linhas cada uma no seguinte formato x FAHRENHEIT EQUIVALE A y CELSIUS, onde x corresponde a um valor de temperatura em Fahrenheit e y corresponde ao valor equivalente em graus Celsius. Logo após a palavra CELSIUS em cada linha de saída deve ser impresso o caractere de quebra de linha. Os valores de x e y devem ser impressos com duas casas decimais.

Exemplo

Entrada
3
8
60
-20
Saída
8.00 FAHRENHEIT EQUIVALE A -13.33 CELSIUS
60.00 FAHRENHEIT EQUIVALE A 15.56 CELSIUS
-20.00 FAHRENHEIT EQUIVALE A -28.89 CELSIUS

5 Divisível por 3 e 5 (+)



(+)

Desenvolver um programa que leia um número inteiro e verifique se o número é divisível por três e também é divisível por cinco.

Entrada

O programa deve ler uma linha contendo um número inteiro na entrada.

Saída

O programa deve imprimir uma linha contendo a frase: O NUMERO E DIVISIVEL, se ele for divisível tanto por três quanto por cinco, ou a frase O NUMERO NAO E DIVISIVEL, em caso contrário. Após imprimir uma das frases, o programa deve imprimir um caractere de quebra de linha: ‘\n’.

Exemplo

Entrada
75
Saída
O NUMERO E DIVISIVEL

6 Conversão de Nota em Conceito (+)



(+)

Em um curso de mestrado as avaliações dos alunos no histórico escolar aparecem em forma de conceito. O regulamento do mestrado indica que um professor pode avaliar seus alunos com notas convencionais de zero a dez, mas precisa repassar à secretaria do curso a avaliação em termos de conceito. Nesse caso, a seguinte tabela de conversão deve ser usada pelo professor:

Intervalo da Nota	Conceito
[9,0 a 10]	A
[7,5,9,0)	B
[6, 7,5)	C
[0 ,6,0)	D

Escreva um programa para ler um nota e converte-la no conceito correspondente.

Entrada

A entrada consiste de uma linha com um valor real entre 0 e 10 e com uma casa decimal.

Saída

O programa deve imprimir a seguinte frase: `NOTA = x CONCEITO = y`, onde x é o valor da nota lido na entrada, impresso com uma casa decimal y é o conceito correspondente. Após a frase, o programa deve imprimir o caractere de quebra de linha: `'\n'`.

Exemplo

Entrada
3.4
Saída
NOTA = 3.4 CONCEITO = D

Entrada
6.0
Saída
NOTA = 6.0 CONCEITO = C

7 Série de pares (+)



(+)

Escreva um programa para ler uma linha com dois números inteiros x e y . O programa deve verificar se x é um número par. Se for, o programa deve imprimir uma sequência de y números pares, iniciando com x . Se x não for par, o programa deve imprimir uma linha com a mensagem: O PRIMEIRO NUMERO NAO E PAR.

Entrada

A entrada conterá uma linha com dois números inteiros separados entre si por um caractere de espaço. Após o segundo número na entrada há um caractere de quebra de linha ($\backslash n$).

Saída

Se o primeiro número for par, o programa deve imprimir uma linha contendo a sequência de números pares, com um espaço entre cada número par. Após o último número da serie, o programa deve imprimir um espaço seguido de um caractere de quebra de linha ($\backslash n$). Se o primeiro número não for par, o programa deve imprimir a mensagem O PRIMEIRO NUMERO NAO E PAR e logo em seguida, o caractere de quebra de linha.

Exemplo

Entrada
20 10
Saída
20 22 24 26 28 30 32 34 36 38

Entrada
3 20
Saída
O PRIMEIRO NUMERO NAO E PAR

8 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:

$$NF = 0.7 \cdot MP + 0.15 \cdot ML + 0.15 \cdot NT \quad (1)$$

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 mínima, 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 matrícula 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																	
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												

9 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 $\text{ANOS} = x$, 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
900000000
2000000000
Saída
ANOS = 55

10 Dados (++)



(++)

Um jogador gostaria de saber quais as suas chances de ganhar em uma partida de dados. Para isso ele acha importante prever de quantas maneiras um par de dados somados pode gerar um número inteiro n , desde que os números sorteados em cada dado sejam diferentes. Ele te contratou para criar um programa que calcule e apresente todas as possibilidades de sorteio. Ele ganhará milhões em apostas no jogo e dividirá os lucros com você se seu programa estiver correto.

Entrada

O programa deverá ler um número inteiro n .

Saída

O programa deve apresentar todas as combinações possíveis das faces do par de dados cuja soma seja igual a n . Cada combinação deve ser apresentada em uma linha seguindo o formato: "D1: x , D2: y ", onde x e y correspondem ao valor das faces dos dados D1 e D2. Na última linha, o programa deve apresentar a quantidade de combinações possíveis com o seguinte texto: "Ha k possibilidades", onde k é a quantidade de possibilidades. Caso não seja possível gerar o valor de n , o programa deve apresentar a mensagem: "Combinacao impossivel".

Exemplo

Entrada	Saída
5	D1: 1, D2: 4 D1: 2, D2: 3 D1: 3, D2: 2 D1: 4, D2: 1 Ha 4 possibilidades
Entrada	Saída
2	Combinacao impossivel

11 Fatorial (++)



(++)

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

12 Maior segmento crescente de uma sequência (++)



(++) (POLI 89) Dados n e uma sequência de n números inteiros, determinar o comprimento de um segmento crescente de comprimento máximo.

Entrada

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

Saída

O programa deve apresentar a mensagem "O comprimento do segmento crescente maximo e: k n", onde k é o tamanho do maior segmento crescente encontrado.

Exemplo

Entrada
9 5 10 3 2 4 7 9 8 5
Saída
O comprimento do segmento crescente maximo e: 4
Entrada
5 10 8 7 5 2
Saída
O comprimento do segmento crescente maximo e: 1

13 Média de pares e ímpares (++)



(++)

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

Entrada

O programa deve ler uma sequência de números inteiros diferentes de zero.

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.

Exemplo

Entrada	Saída
1 5 8 7 6 3 0	MEDIA PAR: 7.000000 MEDIA IMPAR: 4.000000

14 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 seja um número inteiro negativo, o programa deve apresentar a mensagem "Numero invalido!".

Exemplo

Entrada
7
Saída
PRIMO

Entrada
9
Saída
NAO PRIMO

15 Ordena 3 números (++)



(++)

Escreva um algoritmo que leia 3 números reais em qualquer ordem e os apresente de forma ordenada na tela.

Entrada

O programa deve ler 3 valores reais.

Saída

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

Exemplo

Entrada
3.0 1 3.1
Saída
1.00, 3.00, 3.10

16 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.

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 = -5.00
X2 = -1.00

Entrada
2 12 18
Saída
RAIZ UNICA
X1 = -3.00

Entrada
15 17 89
Saída
RAIZES IMAGINARIAS

17 Soma dos 3 menores (++)



(++)

Fazer um programa para quatro valores inteiros e imprimir a soma dos três menores.

Entrada

O programa deve ler quatros valores inteiros na entrada. Cada valor ocupa uma linha na entrada.

Saída

O programa deve imprimir uma linha contendo o valor da soma dos três menores números. Após o valor da soma, o programa deve imprimir um caractere de quebra de linha: ‘\n’.

Exemplo

Entrada
9
4
2
12
Saída
15

18 Transcrição de datas (++)



(++)

Faça um algoritmo que leia uma data no formato ddmmaaaa usando um único número inteiro. Escreva a mesma data no formato dia/mês/ano, <dia> de <mês por extenso> de <ano>. O programa deve verificar se o número informado representa uma data válida. Caso não seja, imprimir na tela a mensagem "Data invalida!". Considere que o ano em questão nunca é bissexto, ou seja, fevereiro tem somente 28 dias.

Entrada

Um número inteiro positivo com 8 dígitos.

Saída

O programa deve apresentar a transcrição da data no formado "dd de mês por extenso de aa".

Exemplo

Entrada
30022001
Saída
Data invalida!
Entrada
12092017
Saída
12 de setembro de 2017

19 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

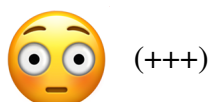
Entrada
3
57
96783
Saída
ACO DE GRAU = 10

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

20 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.

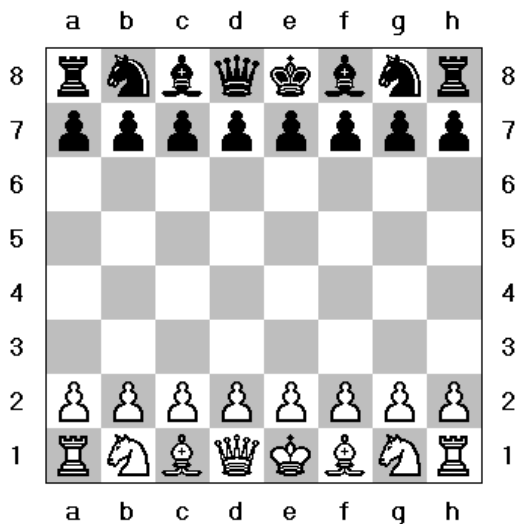


Figura 1: Tabuleiro de xadrez.

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

21 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

22 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 $(c_1, c_2), (c_3, c_4), \dots (c_k, c_{k+1})$, 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

23 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 contém 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%			

24 Mínimo múltiplo comum (+++)



(+++)

Faça um programa que calcule o Mínimo Múltiplo Comum (MMC) de 3 números inteiros. A Figura 24 apresenta um exemplo de cálculo de MMC.

$$\begin{array}{r|l} 4; 6; 8 & 2 \\ 2; 3; 4 & 2 \\ 1; 3; 2 & 2 \\ 1; 3; 1 & 3 \\ 1; 1; 1 & \hline \end{array} \quad mmc(4; 6; 8) = 2^3 \times 3 = 24$$

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

Entrada

O programa deve ler 3 números inteiros diferentes de zero.

Saída

A saída é composta por várias linhas. O programa deve replicar a saída do procedimento da Figura 24 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

25 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: " $k * k * k = x_1 + x_2 + \dots + x_k$ ", onde $k = 1, 2, \dots, m$ e x_i é a sequência de números ímpares consecutivos.

Exemplo

Entrada
4
Saída
$1 * 1 * 1 = 1$
$2 * 2 * 2 = 3 + 5$
$3 * 3 * 3 = 7 + 9 + 11$
$4 * 4 * 4 = 13 + 15 + 17 + 19$

26 Número palíndromo (+++)



(+++)

Faça um programa que leia um número e verifique se ele é palíndromo. Um número é palíndromo quando representa a mesma quantidade lido da esquerda para a direita e da direita para a esquerda. Neste exercício o usuário irá informar números inteiros de no máximo 5 dígitos.

Entrada

Um número inteiro.

Saída

Se o número fornecido exceder 5 dígitos o programa deve imprimir a mensagem "NUMERO INVALIDO". Caso o número seja válido o programa deve imprimir a mensagem "PALINDROMO", caso o número seja palíndromo, ou "NAO PALINDROMO" caso contrário.

Exemplo

Entrada	Saída
131	PALINDROMO
560262	NUMERO INVALIDO
15001	NAO PALINDROMO
12321	PALINDROMO

27 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.

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)

28 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. Assim, 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

29 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 contendo 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

30 Triângulo de Pascal (++)



(++)

Faça um programa que calcule e apresente uma faixa de linhas do Triângulo de Pascal. Cada linha do Triângulo de Pascal é dado pela seguinte equação:

$$\text{Linha } n : \binom{n}{0} \quad \binom{n}{1} \quad \binom{n}{2} \quad \dots \quad \binom{n}{n} \quad (2)$$

sendo que

$$\binom{n}{p} = C_{n,p} = \frac{n!}{p!(n-p)!} \quad (3)$$

Entrada

O programa deverá ler 2 números inteiros, sendo o primeiro correspondendo à linha inicial e o segundo à linha final do Triângulo de Pascal.

Saída

O programa deverá imprimir as linhas do Triângulo de Pascal com os números separados por vírgula.

Exemplo

Entrada	Saída
3 4	1, 3, 3, 1 1, 4, 6, 4, 1

Entrada	Saída
0 5	1 1, 1 1, 2, 1 1, 3, 3, 1 1, 4, 6, 4, 1 1, 5, 10, 10, 5, 1

31 Professor de Fibonacci (+++)



(+++)

Um professor de sequências de números inteiros pretende automatizar o ensino da sequência de Fibonacci. Para isso ele precisa de um software que lê uma sequência qualquer e verifique se a sequência dada é de Fibonacci. Uma sequência de Fibonacci é construída de forma que, dados dois números iniciais, o próximo número sempre é a soma dos dois números anteriores. Por exemplo, para a sequência que inicia-se com $F_1 = 1$ e $F_2 = 2$, a sequência fica:

$$\begin{array}{cccccccccc} F_1 & F_2 & F_3 & F_4 & F_5 & F_6 & F_7 & F_8 & \dots \\ 1 & 2 & 3 & 5 & 8 & 13 & 21 & 34 & \dots \end{array} \quad (4)$$

Matematicamente podemos escrever:

$$F_n = F_{n-1} + F_{n-2} \quad (5)$$

O software deve ler número por número e verificar se a sequência segue a estrutura de construção da sequência de Fibonacci.

Entrada

Inicialmente o programa deve ler 3 números inteiros n , F_1 e F_2 . O primeiro é a quantidade de números da sequência, o segundo e o terceiro são os números iniciais da sequência de Fibonacci. Em seguida deve-se ler n números inteiros. O valor de n sempre será maior que 2.

Saída

Para cada entrada dos n números, o programa deve imprimir uma linha com a mensagem "OK" caso o número dado coincida com a série de Fibonacci. Caso não coincida, o programa deve escrever a mensagem "Nao e Fibonacci" e encerrar. Caso todos os números dados formem a série de Fibonacci, o programa deve imprimir a mensagem "A serie informada e de Fibonacci".

Exemplo

Entrada	Saída
4 1 1 1 1 2 3	OK OK OK OK A serie informada e de Fibonacci

Entrada	Saída
5 0 1 0 1 2 3 8	OK OK Nao e Fibonacci

Entrada	Saída
7 2 3 2 4 6 10 16 26 42	OK Nao e Fibonacci

32 Sequência intercalada (+++)



(+++)

Faça um programa que leia uma sequência de números inteiros e verifique se esta sequência tem um padrão intercalado de ordem crescente e decrescente.

Entrada

Para cada sequência numérica há na entrada duas linhas: uma com apenas um valor inteiro, maior que 2 e que indica o número de valores que deve ocorrer na próxima linha. A linha seguinte contém a sequência, formada por tantos valores quanto indicado na linha anterior. Entre dois valores da sequência 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 sequência da entrada o seu programa deve emitir uma das seguintes respostas: "Intercalada", se a sequência estiver em ordem intercalada de valores ou "Nao intercalada", em caso contrário. Após cada palavra impressa deve haver apenas um caractere de quebra de linha.

Exemplo

Entrada	Saída
3	Intercalada
1 5 2	Intercalada
5	Nao intercalada
1 9 3 8 4	
4	
1 2 3 4	
0	

33 Calcular Dígito Verificador do Título de Eleitor (+++)



(+++)

O Número de Inscrição do Título Eleitoral tem o seu **Dígito Verificador (DV)**. Para cálculo do DV, observe-se que o número é desmembrado em Número Sequencial (NS), de até 8 dígitos, seguido de 2 dígitos para a Unidade da Federação (UF) e dos dois dígitos do próprio DV.

Assim, o Título Eleitoral número 43568709/06 tem NS=**435687**, UF=**09** e DV=**06**. Para calcular o 1º dígito do DV usa-se uma soma ponderada do número NS de acordo com a seguinte regra: o primeiro dígito do NS multiplica-se por 2, o segundo multiplica-se por 3 e assim por diante, até que o oitavo dígito é multiplicado por 9. Com a soma dessas multiplicações tem-se a base de cálculo (BC1d) para o 1º dígito do DV. O 1º dígito do DV corresponde ao **BC1d mod 11** do número sequencial (435687), sendo que **mod** é o operador de resto da divisão. O 2º dígito do DV já usa uma soma ponderada dos dígitos da UF seguidos do 1º dígito do DV. A base de cálculo BC2d é calculada pela soma das multiplicações do primeiro dígito do UF por 7, do segundo por 8 e do 1º dígito do DV por 9. O 2º dígito é então calculado pela operação **BC2d mod 11**. Desse modo, para o exemplo dado, temos DV=06. Portanto:

Para o primeiro dígito:

0	0	4	3	5	6	8	7
x	x	x	x	x	x	x	x
2	3	4	5	6	7	8	9

$0 + 0 + 16 + 15 + 30 + 42 + 64 + 63 = 230$
 $230 / 11 = 20$, com resto = 10.
(quando o resto for 10 então o dígito será 0)

Para o segundo dígito:

0	9	0
x	x	x
7	8	9

$0 + 72 + 0 = 72$
 $72 / 11 = 6$, com resto 6
(então o segundo dígito de DV será 6)

Observações

- Os dígitos de UF correspondem a: 01-SP, 02-MG, 03-RJ, 04-RS, 05-BA, 06-PR, 07-CE, 08-PE, 09-SC, 10-GO, 11-MA, 12-PB, 13-PA, 14-ES, 15-PI, 16-RN, 17-AL, 18-MT, 19-MS, 20-DF, 21-SE, 22-AM, 23-RO, 24-AC, 25-AP, 26-RR, 27-TO e 28-Exterior(ZZ).
- Quando a operação **BCx mod 11** tiver resultado igual a 0, o dígito resultante é também 0. Em **BCx** o **x** representa **1d** ou **2d**.

Entrada

Dois números inteiros que correspondem, respectivamente, ao **Número sequencial** do título de eleitor e **código da UF**.

Saída

Número do título completo e formatado: número sequencial, código da UF e DV. Por exemplo: 43568709/06
Se o código da UF for diferente de um dos códigos válidos, exibir a mensagem “CODIGO DA UF INVALIDO!”.

Exemplo

Entrada	Saída
435687 9	43568709/06

Entrada	Saída
435687 66	CODIGO DA UF INVALIDO!

Entrada	Saída
123456 10	12345610/07

Entrada	Saída
8745 17	874517/91

34 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

35 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 por 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)

36 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:

$$e^x = \sum_{n=0}^N \frac{x^n}{(n)!} = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \dots + \frac{x^N}{(N)!} \quad (6)$$

, 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 " $e^x = y \backslash n$ ", 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
$e^{1.00} = 2.718282$

37 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 de fatoração 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	120	2
18	2	60	2
9	3	30	2
3	3	15	3
1		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: $N = f_1 \times f_2 \times \dots \times f_k$.

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