

Universidade Federal de Goiás
Instituto de Informática
Introdução à Programação - 2018-1
Lista L1b

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

Sumário

1	Aprovado ou Reprovado	3
2	Conta de Água	4
3	Divisível por 3 e 5	6
4	Locadora de charretes	7
5	Conversão de Nota em Conceito	8
6	Reajuste salarial	9
7	Valor de y dado x	10
8	Ano bissexto (++)	11
9	Descrição de inteiros	12
10	Ordena 3 números	13
11	Raízes de equações de grau 2	14
12	Raízes de equações de grau 2	16
13	Soma dos 3 menores	18
14	Transcrição de datas	19
15	Classificação do Aço	20
16	Cálculo do imposto de renda	21
17	Número palíndromo (+++)	23
18	Ordem (+++)	24
19	Ordena 4 números (+++)	25
20	Regra de cotas (+++)	26
21	Triângulo ou trapézio? (+++)	27
22	Várias Ordenações (+++)	29

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'`.

Observação

Os valores floats podem sofrer variações de uma máquina para outra. Quando se usa `printf("%.2f", x)`; O `printf` arredonda o float x para duas casas decimais. Esse arredondamento pode ter pequenas diferenças de uma máquina para outra. por exemplo, se $x = 0.34561$, uma máquina pode imprimir 0.34 enquanto outra imprime $x = 0.35$. Isso pode fazer com que seu programa seja considerado errado pelo Sharif. Para evitar isso, vamos usar um "truncamento do número x " usando a função `truncf()`: `printf("%.2f", truncf(x*100.0)*100.0)`; A expressão acima multiplica o valor de x por 100.00 antes de trunca-lo (podar sustas casas decimais) e depois divide o número por 100.0 novamente, fazendo com que o número seja impresso truncado em duas casas decimais. No exemplo: $0.34561 * 100.00 = 34.561$. `truncf(34.51) = 34`; dividido por 100: 0.34. Para truncar em uma casa decimal, multiplique e depois divida por 10.0. Se for com três casas decimais, multiplique e depois divida por 1000.0, e assim por diante.

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 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 gastos;
- Industrial: R\$ 800,00 para os primeiros 100 metros cúbicos gastos mais R\$ 0,04 por metros cúbicos gastos;

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.

Observação

Os valores floats podem sofrer variações de uma máquina para outra. Quando se usa `printf("%.2f", x)`; O `printf` arredonda o float x para duas casas decimais. Esse arredondamento pode ter pequenas diferenças de uma máquina para outra. por exemplo, se $x = 0.34561$, uma máquina pode imprimir 0.34 enquanto outra imprime $x = 0.35$. Isso pode fazer com que seu programa seja considerado errado pelo Sharif. Para evitar isso, vamos usar um "truncamento do número x " usando a função `truncf()`: `printf("%.2f", truncf(x*100.0)*100.0)`; A expressão acima multiplica o valor de x por 100.00 antes de trunca-lo (podar sustas casas decimais) e depois divide o número por 100.0 novamente, fazendo com que o número seja impresso truncado em duas casas decimais. No exemplo: $0.34561 * 100.00 = 34.561$. `truncf(34.51) = 34`; dividido por 100: 0.34. Para truncar em uma casa decimal, multiplique e depois divida por 10.0. Se for com três casas decimais, multiplique e depois divida por 1000.0, e assim por diante.

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

3 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

4 Locadora de charretes



(+)

Uma locadora de charretes cobra R\$ 10,00 de taxa para cada 3 horas de uso de uma charrete e R\$5,00 para cada 1 hora abaixo dessas 3 horas. Fazer um programa que leia a quantidade de horas que a charrete foi usada e que calcule e escreva quanto o cliente tem de pagar.

Entrada

O programa deve ler uma única linha na entrada, contendo o número de horas que o locatário utilizou a charrete.

Saída

O programa deve imprimir uma linha contendo a frase: O VALOR A PAGAR E = X, onde X é o valor do aluguel e deve conter no máximo 2 casas decimais. Após o valor do aluguel o programa deve imprimir um caractere de quebra de linha: ‘\n’.

Observação

Os valores floats podem sofrer variações de uma máquina para outra. Quando se usa `printf("%.2f", x)`; O `printf` arredonda o float `x` para duas casas decimais. Esse arredondamento pode ter pequenas diferenças de uma máquina para outra. por exemplo, se `x= 0.34561`, uma máquina pode imprimir 0.34 enquanto outra imprime `x=0.35`. Isso pode fazer com que seu programa seja considerado errado pelo Sharif. Para evitar isso, vamos usar um "truncamento do número `x`" usando a função `truncf()`: `printf("%.2f", truncf(x*100.0)*100.0)`; A expressão acima multiplica o valor de `x` por 100.00 antes de trunca-lo (podar sustas casas decimais) e depois divide o número por 100.0 novamente, fazendo com que o número seja impresso truncado em duas casas decimais. No exemplo: `0.34561*100.00= 34.561`. `truncf(34.51) = 34`; dividido por 100: 0.34. Para truncar em uma casa decimal, multiplique e depois divida por 10.0. Se for com três casas decimais, multiplique e depois divida por 1000.0, e assim por diante.

Exemplo

Entrada
29
Saída
O VALOR A PAGAR E = 100.00

5 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'`.

Observação

Os valores floats podem sofrer variações de uma máquina para outra. Quando se usa `printf("%.2f", x)`; O `printf` arredonda o float x para duas casas decimais. Esse arredondamento pode ter pequenas diferenças de uma máquina para outra. por exemplo, se $x=0.34561$, uma máquina pode imprimir 0.34 enquanto outra imprime $x=0.35$. Isso pode fazer com que seu programa seja considerado errado pelo Sharif. Para evitar isso, vamos usar um "truncamento do número x " usando a função `truncf()`: `printf("%.2f", truncf(x*100.0)*100.0)`; A expressão acima multiplica o valor de x por 100.00 antes de trunca-lo (podar sustas casas decimais) e depois divide o número por 100.0 novamente, fazendo com que o número seja impresso truncado em duas casas decimais. No exemplo: $0.34561*100.00= 34.561$. `truncf(34.51) = 34`; dividido por 100: 0.34. Para truncar em uma casa decimal, multiplique e depois divida por 10.0. Se for com três casas decimais, multiplique e depois divida por 1000.0, e assim por diante.

Exemplo

Entrada
3.4
Saída
NOTA = 3.4 CONCEITO = D

Entrada
6.0
Saída
NOTA = 6.0 CONCEITO = C

6 Reajuste salarial



(+)

Fazer um algoritmo que calcule e imprima o salário reajustado de um funcionário de acordo com as seguintes regras:

- Salário de até R\$ 300,00, reajuste de 50%;
- Salário maior que R\$300,00 reajuste de 30%;

Entrada

A entrada conterá uma linha com o salário do funcionário.

Saída

A saída deve conter, numa linha com a frase: SALARIO COM REAJUSTE = x , onde x é um valor real com duas casas decimais e corresponde ao valor do salário reajustado. Logo em seguida ao valor de x , o programa devem imprimir o caractere de quebra de linha: '\n'.

Observação

Os valores floats podem sofrer variações de uma máquina para outra. Quando se usa `printf("%.2f", x)`; O `printf` arredonda o float x para duas casas decimais. Esse arredondamento pode ter pequenas diferenças de uma máquina para outra. por exemplo, se $x=0.34561$, uma máquina pode imprimir 0.34 enquanto outra imprime $x=0.35$. Isso pode fazer com que seu programa seja considerado errado pelo Sharif. Para evitar isso, vamos usar um "truncamento do número x " usando a função `truncf()`: `printf("%.2f", truncf(x*100.0)*100.0)`; A expressão acima multiplica o valor de x por 100.00 antes de trunca-lo (podar sustas casas decimais) e depois divide o número por 100.0 novamente, fazendo com que o número seja impresso truncado em duas casas decimais. No exemplo: $0.34561*100.00=34.561$. `truncf(34.51)=34`; dividido por 100: 0.34. Para truncar em uma casa decimal, multiplique e depois divida por 10.0. Se for com três casas decimais, multiplique e depois divida por 1000.0, e assim por diante.

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 por apenas uma linha de entrada.

Entrada
755.00
Saída
SALARIO COM REAJUSTE = 981.50

Entrada
265.32
Saída
SALARIO COM REAJUSTE = 397.98

7 Valor de y dado x



(+)

Desenvolver um algoritmo para ler um número x , calcular e imprimir o valor de y de acordo com as condições abaixo:

$$y = \begin{cases} x, & \text{se } x < 1; \\ 0, & \text{se } x = 1; \\ x^2, & \text{se } x > 1; \end{cases}$$

Entrada

O programa deve ler uma linha contendo um único número inteiro correspondendo ao valor de x .

Saída

O programa deve imprimir $Y = y$, onde y é o valor computado de y dado x . Após o valor de y , o programa deve imprimir um caractere de quebra de linha: ‘\n’.

Exemplo

Entrada
3
Saída
Y = 9

8 Ano bissexto (++)



(++)

Escreva um algoritmo que leia um ano e imprima na tela se o ano lido é bissexto ou não. Um ano é bissexto se ele for múltiplo de 4, exceto quando ele for múltiplo de 100. Os anos múltiplos de 100 somente são bissextos quando são múltiplos de 400. Podem ser anos bissextos somente os anos maiores que 1582.

Entrada

O programa deve ler um número inteiro.

Saída

O programa deve imprimir a mensagem: "ANO BISSEXTO" caso o ano seja bissexto ou "ANO NAO BISSEXTO" caso contrário.

Exemplo

Entrada	Saída
1872	ANO BISSEXTO
2000	ANO BISSEXTO
1901	ANO NAO BISSEXTO

9 Descrição de inteiros



(++)

Escreva um algoritmo em Linguagem C que leia um número inteiro (maior que zero e de no máximo 4 dígitos), imprima na tela quantas unidades de milhar, centenas, dezenas e unidades formam o número e a qual ordem o número pertence.

Entrada

O programa deve ler um número inteiro.

Saída

O algoritmo deve imprimir a mensagem “Numero invalido!” se o número não está dentro do intervalo estabelecido e, caso seja válido, realizar a decomposição do número e decidir se as palavras devem ser impressas no singular ou no plural. Caso a quantidade de uma ordem seja 0, o algoritmo não deve imprimir a quantidade da ordem. O algoritmo deve reproduzir fielmente os exemplos abaixo.

Exemplo

Entrada
1257
Saída
(quarta ordem) 1257 = 1 unidade de milhar + 2 centenas + 5 dezenas + 7 unidades = 1000 + 200 + 50 + 7

Entrada
725
Saída
(terceira ordem) 725 = 7 centenas + 2 dezenas + 5 unidades = 700 + 20 + 5

Entrada
203
Saída
(terceira ordem) 203 = 2 centenas + 3 unidades = 200 + 3

Entrada
12
Saída
(segunda ordem) 12 = 1 dezena + 2 unidade

10 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.

Observação

Os valores floats podem sofrer variações de uma máquina para outra. Quando se usa `printf("%.2f", x)`; O `printf` arredonda o float `x` para duas casas decimais. Esse arredondamento pode ter pequenas diferenças de uma máquina para outra. por exemplo, se `x= 0.34561`, uma máquina pode imprimir 0.34 enquanto outra imprime `x=0.35`. Isso pode fazer com que seu programa seja considerado errado pelo Sharif. Para evitar isso, vamos usar um "truncamento do número `x`" usando a função `truncf()`: `printf("%.2f", truncf(x*100.0)*100.0)`; A expressão acima multiplica o valor de `x` por 100.00 antes de trunca-lo (podar sustas casas decimais) e depois divide o número por 100.0 novamente, fazendo com que o número seja impresso truncado em duas casas decimais. No exemplo: `0.34561*100.00= 34.561`. `truncf(34.51) = 34`; dividido por 100: 0.34. Para truncar em uma casa decimal, multiplique e depois divida por 10.0. Se for com três casas decimais, multiplique e depois divida por 1000.0, e assim por diante.

Exemplo

Entrada
3.0
1
3.1
Saída
1.00, 3.00, 3.10

11 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:

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

- Os valores floats podem sofrer variações de uma máquina para outra. Quando se usa `printf("%.2f ", x)`; O `printf` arredonda o float x para duas casas decimais. Esse arredondamento pode ter pequenas diferenças de uma máquina para outra. por exemplo, se $x = 0.34561$, uma máquina pode imprimir 0.34 enquanto outra imprime $x = 0.35$. Isso pode fazer com que seu programa seja considerado errado pelo Sharif. Para evitar isso, vamos usar um "truncamento do número x " usando a função `truncf()`: `printf("%.2f", truncf(x*100.0)*100.0)`; A expressão acima multiplica o valor de x por 100.00 antes de trunca-lo (podar sustas casas decimais) e depois divide o número por 100.0 novamente, fazendo com que o número seja impresso truncado em duas casas decimais. No exemplo: $0.34561 * 100.00 = 34.561$. `truncf(34.51) = 34`; dividido por 100: 0.34. Para truncar em uma casa decimal, multiplique e depois divida por 10.0. Se for com três casas decimais, multiplique e depois divida por 1000.0, e assim por diante.

Exemplo

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

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

Entrada
2 12 18
Saída
RAIZ UNICA
X1 = -3.00

Entrada
15 17 89
Saída
RAIZES IMAGINARIAS

12 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 = -1.00
X2 = -5.00

Entrada
2 12 18
Saída
RAIZ UNICA
X1 = -3.00

Entrada
15 17 89
Saída
RAIZES IMAGINARIAS

13 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

14 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 formato "dd de mês por extenso de aa".

Exemplo

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

15 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

16 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 R300,00 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 normal 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.

Observação

Os valores floats podem sofrer variações de uma máquina para outra. Quando se usa `printf("%.2f", x)`; O `printf` arredonda o float x para duas casas decimais. Esse arredondamento pode ter pequenas diferenças de uma máquina para outra. por exemplo, se $x = 0.34561$, uma máquina pode imprimir 0.34 enquanto outra imprime $x = 0.35$. Isso pode fazer com que seu programa seja considerado errado pelo Sharif. Para evitar isso, vamos usar um "truncamento do número x " usando a função `truncf()`: `printf("%.2f", truncf(x*100.0)*100.0)`; A expressão acima multiplica o valor de x por 100.00 antes de trunca-lo (podar sustas casas decimais) e depois divide o número por 100.0 novamente, fazendo com que o número seja

impresso truncado em duas casas decimais. No exemplo: $0.34561 \times 100.00 = 34.561$. $\text{truncf}(34.51) = 34$; dividido por 100: 0.34. Para truncar em uma casa decimal, multiplique e depois divida por 10.0. Se for com três casas decimais, multiplique e depois divida por 1000.0, e assim por diante.

Exemplo

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

Entrada	Saída
99123 510.0 3 1531.97 8.5	MATRICULA = 99123 IMPOSTO BRUTO = 0.00 IMPOSTO LIQUIDO = -900.00 RESULTADO = -1030.22 IMPOSTO A RECEBER

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

17 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

18 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	Saída
1 5 3 A B C	1 3 5

Entrada	Saída
6 4 2 C A B	6 2 4

19 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.

Observação

Os valores floats podem sofrer variações de uma máquina para outra. Quando se usa `printf("%.2f", x)`; O `printf` arredonda o float `x` para duas casas decimais. Esse arredondamento pode ter pequenas diferenças de uma máquina para outra. por exemplo, se `x = 0.34561`, uma máquina pode imprimir 0.34 enquanto outra imprime `x = 0.35`. Isso pode fazer com que seu programa seja considerado errado pelo Sharif. Para evitar isso, vamos usar um "truncamento do número `x`" usando a função `truncf()`: `printf("%.2f", truncf(x*100.0)*100.0)`; A expressão acima multiplica o valor de `x` por 100.00 antes de trunca-lo (podar sustas casas decimais) e depois divide o número por 100.0 novamente, fazendo com que o número seja impresso truncado em duas casas decimais. No exemplo: `0.34561*100.00 = 34.561`. `truncf(34.51) = 34`; dividido por 100: 0.34. Para truncar em uma casa decimal, multiplique e depois divida por 10.0. Se for com três casas decimais, multiplique e depois divida por 1000.0, e assim por diante.

Exemplo

Entrada
3.0
1
3.1
8
Saída
1.00, 3.00, 3.10, 8.00

20 Regra de cotas (+++)



(+++)

O modelo atual de seleção de alunos para o ingresso em universidades, SISU, considera o uso de cotas. No momento da inscrição, o candidato apresenta alguns dados que são usados para o enquadrar em uma das classes de cotistas:

- Cota L1: Candidatos que cursaram todo o ensino médio em escolas públicas, com renda familiar bruta per capita igual ou inferior a 1,5 salário mínimo
- Cota L2: Candidatos que cursaram todo o ensino médio em escolas públicas, com renda familiar bruta per capita igual ou inferior a 1,5 salário mínimo, autodeclarados pretos, pardos ou indígenas
- Cota L3: Candidatos que cursaram todo o ensino médio em escolas públicas, independente da renda
- Cota L4: Candidatos que cursaram todo o ensino médio em escolas públicas, independente da renda, autodeclarados pretos, pardos ou indígenas

Faça um programa que leia os dados do usuário e verifique se o aluno é cotista ou não. Caso seja, o programa deve verificar qual a classe de cota a qual o aluno pertence. Considere o valor do salário mínimo igual a R\$ 937,00.

Entrada

O programa deve ler 1 número real e 3 inteiros correspondentes às seguintes informações sobre o candidato: renda bruta da família, a quantidade de pessoas que compõe a família, o tipo de escola que cursou o ensino médio (1 - particular; 2 - pública) e a sua etnia (1 – preto; 2 – pardo; 3 – indígena; 4 – branco).

Saída

O programa deve imprimir a mensagem "ALUNO NAO COTISTA" caso o aluno não seja cotista e a mensagem "ALUNO COTISTA (Lx)" caso o aluno seja cotista. O valor de x deve ser a classe de cotas que o aluno pertence.

Exemplo

Entrada	Saída
5602 6 2 2	ALUNO COTISTA (L2)

Entrada	Saída
15000 3 2 1	ALUNO COTISTA (L4)

Entrada	Saída
2000 2 1 1	ALUNO NAO COTISTA

21 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:

- $|b - c| < a < b + c$;
- $|a - c| < b < a + c$;
- $|a - b| < c < a + b$;
- Os valores floats podem sofrer variações de uma máquina para outra. Quando se usa `printf("%.2f", x)`; O `printf` arredonda o float x para duas casas decimais. Esse arredondamento pode ter pequenas diferenças de uma máquina para outra. por exemplo, se $x = 0.34561$, uma máquina pode imprimir 0.34 enquanto outra imprime $x = 0.35$. Isso pode fazer com que seu programa seja considerado errado pelo Sharif. Para evitar isso, vamos usar um "truncamento do número x " usando a função `truncf()`: `printf("%.2f", truncf(x*100.0)*100.0)`; A expressão acima multiplica o valor de x por 100.00 antes de trunca-lo (podar sustas casas decimais) e depois divide o número por 100.0 novamente, fazendo com que o número seja impresso truncado em duas casas decimais. No exemplo: $0.34561 * 100.00 = 34.561$. `truncf(34.51) = 34`; dividido por 100: 0.34. Para truncar em uma casa decimal, multiplique e depois divida por 10.0. Se for com três casas decimais, multiplique e depois divida por 1000.0, e assim por diante.

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

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

Exemplo

Entrada
6.0 4.0 2.0
Saída
Area = 10.0

Entrada
6.0 4.0 2.1
Saída
Perimetro = 12.1

22 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 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.

Observação

Os valores floats podem sofrer variações de uma máquina para outra. Quando se usa `printf("%.2f", x)`; O `printf` arredonda o float x para duas casas decimais. Esse arredondamento pode ter pequenas diferenças de uma máquina para outra. por exemplo, se $x = 0.34561$, uma máquina pode imprimir 0.34 enquanto outra imprime $x = 0.35$. Isso pode fazer com que seu programa seja considerado errado pelo Sharif. Para evitar isso, vamos usar um "truncamento do número x " usando a função `truncf()`: `printf("%.2f", truncf(x*100.0)*100.0)`; A expressão acima multiplica o valor de x por 100.00 antes de trunca-lo (podar sustas casas decimais) e depois divide o número por 100.0 novamente, fazendo com que o número seja impresso truncado em duas casas decimais. No exemplo: $0.34561 * 100.00 = 34.561$. `truncf(34.51) = 34`; dividido por 100: 0.34. Para truncar em uma casa decimal, multiplique e depois divida por 10.0. Se for com três casas decimais, multiplique e depois divida por 1000.0, e assim por diante.

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