

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

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

## Instruções para a Resolução dos Problemas

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

Grau de dificuldade	Pontos
+	1
++	2
+++	3
++++	4
+++++	5

A lista vale 100 pontos (que correspondem a 10 em termos de nota da lista). Para obter os cem pontos o aluno deve conseguir resolver um número de exercícios de graus de dificuldade +, ++ e +++ que somados formem 90 pontos. Pontos excedentes obtidos com exercícios desses graus de dificuldade serão descartados. Os 10 pontos restantes devem ser obtidos resolvendo-se problemas com graus de dificuldade ++++ ou +++++.

# Sumário

<b>1</b>	<b>Aprovado ou Reprovado</b>	<b>5</b>
<b>2</b>	<b>Arrecadação de Jogos</b>	<b>6</b>
<b>3</b>	<b>Composição Inteira</b>	<b>7</b>
<b>4</b>	<b>Consumo de energia</b>	<b>8</b>
<b>5</b>	<b>Conta de Água</b>	<b>9</b>
<b>6</b>	<b>Conversão de temperatura</b>	<b>10</b>
<b>7</b>	<b>Conversões para o Sistema Métrico</b>	<b>11</b>
<b>8</b>	<b>Custo da Lata de Cerveja</b>	<b>12</b>
<b>9</b>	<b>Cálculo do Delta na Equação de Báskara</b>	<b>13</b>
<b>10</b>	<b>Cálculo do Determinante de uma Matriz Quadrada de Duas Dimensões</b>	<b>14</b>
<b>11</b>	<b>Divisível por 3 e 5</b>	<b>15</b>
<b>12</b>	<b>Locadora de charretes</b>	<b>16</b>
<b>13</b>	<b>Conversão de Nota em Conceito</b>	<b>17</b>
<b>14</b>	<b>Volume da Pirâmide de Base Hexagonal</b>	<b>18</b>
<b>15</b>	<b>Quadrado de pares</b>	<b>19</b>
<b>16</b>	<b>Reajuste salarial</b>	<b>20</b>
<b>17</b>	<b>Série de pares</b>	<b>21</b>
<b>18</b>	<b>Soma de progressão aritmética</b>	<b>22</b>
<b>19</b>	<b>Somatório simples</b>	<b>23</b>
<b>20</b>	<b>Tempo em segundos</b>	<b>24</b>
<b>21</b>	<b>Turma de Introdução à Programação</b>	<b>25</b>
<b>22</b>	<b>Ultrapassagem populacional</b>	<b>27</b>
<b>23</b>	<b>Valor de <math>y</math> dado <math>x</math></b>	<b>28</b>
<b>24</b>	<b>Arredondamento</b>	<b>29</b>
<b>25</b>	<b>Cálculo da Área de um Triângulo</b>	<b>30</b>
<b>26</b>	<b>Custo Final de um Carro</b>	<b>31</b>
<b>27</b>	<b>Decolagem</b>	<b>32</b>

<b>28 Distância entre dois pontos</b>	<b>34</b>
<b>29 Fatorial</b>	<b>35</b>
<b>30 Gerador de tabuada</b>	<b>36</b>
<b>31 José</b>	<b>37</b>
<b>32 Maior segmento crescente de uma sequência</b>	<b>38</b>
<b>33 Maior segmento igual de uma sequência</b>	<b>39</b>
<b>34 Número de finais</b>	<b>40</b>
<b>35 Número primo</b>	<b>41</b>
<b>36 Número primo</b>	<b>42</b>
<b>37 Ordena 3 números</b>	<b>43</b>
<b>38 Quatro Algarismos</b>	<b>44</b>
<b>39 Raízes de equações de grau 2</b>	<b>45</b>
<b>40 Salário</b>	<b>47</b>
<b>41 Sequência ordenada</b>	<b>48</b>
<b>42 Soma dos 3 menores</b>	<b>49</b>
<b>43 Transcrição de datas</b>	<b>50</b>
<b>44 Cálculo da raiz quadrada</b>	<b>51</b>
<b>45 Classificação do Aço</b>	<b>52</b>
<b>46 Companhia de Teatro</b>	<b>53</b>
<b>47 Conversão de decimal para binário</b>	<b>55</b>
<b>48 Grãos de milho no tabuleiro de xadrez</b>	<b>57</b>
<b>49 Cálculo do imposto de renda</b>	<b>58</b>
<b>50 Índices da matriz inferior</b>	<b>60</b>
<b>51 Hipotenusas inteiras</b>	<b>61</b>
<b>52 Lucro de Mercadorias</b>	<b>62</b>
<b>53 N ao cubo</b>	<b>64</b>
<b>54 Número Invertido</b>	<b>65</b>
<b>55 Número perfeito</b>	<b>66</b>

<b>56</b>	<b>Ordem</b>	<b>67</b>
<b>57</b>	<b>Ordena 4 números</b>	<b>68</b>
<b>58</b>	<b>Sistemas de Equação Linear</b>	<b>69</b>
<b>59</b>	<b>Triângulo ou trapézio?</b>	<b>70</b>
<b>60</b>	<b>Valor em Notas e Moedas</b>	<b>71</b>
<b>61</b>	<b>Várias Ordenações</b>	<b>72</b>
<b>62</b>	<b>Transforma decimal em fração</b>	<b>73</b>
<b>63</b>	<b>Procura por número amigo</b>	<b>74</b>
<b>64</b>	<b>Série de Taylor para a função cosseno</b>	<b>75</b>
<b>65</b>	<b>Série de Taylor para a função <math>e^x</math></b>	<b>76</b>
<b>66</b>	<b>Série de Taylor para a função seno</b>	<b>77</b>
<b>67</b>	<b>Decomposição em fatores primos</b>	<b>78</b>

# 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 Arrecadação de Jogos



(+)

Escrever um algoritmo que lê o público total de futebol, as percentagens de pessoas nas seguintes categorias: *popular*, *geral*, *arquibancada* e *cadeiras* e forneça a renda total do jogo. Sabe-se que o valor dos ingressos por categoria são dados pela tabela abaixo:

Categoria	Valor ingresso
Popular	R\$ 1,00
Geral	R\$ 5,00
Arquibancada	R\$10,00
Cadeiras	R\$ 20,00

### Entrada

A entrada contém uma linha inicial com um valor inteiro informando o número de casos de testes que ocorrem nas linhas seguintes. Cada caso de teste seguinte é formado por uma linha contendo os seguintes valores, separados entre si por um espaço:

- O número de pessoas que compraram ingresso para o jogo correspondente ao caso de teste.
- A percentagem de pessoas que compraram ingresso na categoria *Popular*.
- A percentagem de pessoas que compraram ingresso na categoria *Geral*.
- A percentagem de pessoas que compraram ingresso na categoria *Arquibancada*.
- A percentagem de pessoas que compraram ingresso na categoria *Cadeiras*.

### Saída

O programa deve gerar uma linha para cada caso de teste na entrada, contendo a frase: A RENDA DO JOGO N.  $x$  E =  $y$ , onde  $x$  corresponde a ordem do caso de teste na entrada e  $y$  é um valor real com duas casas decimais que corresponde ao valor da renda total do jogo  $x$ .

### Exemplo

Entrada
3
55000 20.2 50.4 30.2 10.2
49732 15.2 53.4 20.24 11.16
67890 30.0 42.20 23.8 4.0
Saída
A RENDA DO JOGO N. 1 E = 428010.00
A RENDA DO JOGO N. 2 E = 352003.09
A RENDA DO JOGO N. 3 E = 379505.09

### 3 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.

#### 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

## 4 Consumo de energia



(+)

Sabendo-se que 100 kW de energia custam 70% do salário mínimo, escreva um algoritmo em Linguagem C que leia o valor do salário mínimo e a quantidade de kW gasta por uma residência. Calcule e imprima:

- o valor em reais de cada kW;
- o valor em reais a ser pago pelo consumo da residência;
- o novo valor a ser pago pela residência com um desconto de 10%.

### Entrada

O programa deve ler o valor do salário mínimo e a quantidade de kW gasta por uma residência. Ambos os valores são reais.

### Saída

O programa deve imprimir três linhas contendo o texto:

Custo por kW: R\$ x.xx

Custo do consumo: R\$ x.xx

Custo com desconto: R\$ x.xx

### Exemplo

Entrada
81
3.54
Saída
Custo por kW: R\$ 56.70
Custo do consumo: R\$ 200.72
Custo com desconto: R\$ 180.65



## 5 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.

### Exemplo

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

<b>Entrada</b>
39393939 230 C
<b>Saída</b>
CONTA = 39393939
VALOR DA CONTA = 537.50

<b>Entrada</b>
888 3752 I
<b>Saída</b>
CONTA = 888
VALOR DA CONTA = 946.08

## 6 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 (float) 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

## 7 Conversões para o Sistema Métrico



(+)

Muitos países estão passando a utilizar o sistema métrico. Faça um programa para executar as seguintes conversões:

- Ler uma temperatura em Fahrenheit e imprimir o equivalente em Celsius ( $C = (5F - 160)/9$ ).
- Ler uma quantidade de chuva dada em polegadas e imprimir o equivalente em milímetros (1 polegada = 25.4 mm).

### Entrada

O programa deve ler dois valores na entrada: um valor em Fahrenheit e outro valor em polegadas. Ambos os valores são do tipo float. Cada valor ocorre em uma linha diferente na entrada.

### Saída

O programa deve imprimir duas linhas. Aa primeira contém a frase: O VALOR EM CELSIUS = X, onde X é o valor de temperatura convertido de Fahrenheit para Celsius e deve ter duas casas decimais. A segunda linha deve conter a frase: A QUANTIDADE DE CHUVA E = Y, onde Y é o valor em milímetros correspondente ao valor em polegadas dado como entrada. Y é um valor real (float) e deve ter duas casas decimais. Logo após o valor de Y, o programa deve imprimir o caractere de quebra de linha ‘\n’.

### Exemplo

Entrada
53
120
Saída
O VALOR EM CELSIUS = 11.67
A QUANTIDADE DE CHUVA E = 3048.00

## 8 Custo da Lata de Cerveja



(+)

Um fabricante de latas deseja desenvolver um programa para calcular o custo de uma lata cilíndrica de alumínio, sabendo-se que o custo do alumínio por  $m^2$  é R\$ 100,00.

### Entrada

O programa deve ler dois valores na entrada: o raio e a altura da lata. Ambos os valores correspondem a valores em metros. Cada valor ocorre em uma linha diferente na entrada.

### Saída

O programa deve imprimir a frase: O VALOR DO CUSTO E = XXX.XX, onde XXX.XX é o valor do custo da lata. Logo após o valor do custo da lata o programa deve imprimir o caractere de quebra de linha '\n'.

### Observações

- O seu programa deve utilizar a constante  $\pi$  com o valor aproximado de 3.14159.
- O valor total da área de um cilindro é dada por  $A_t = 2A_c + A_l$ , onde  $A_c$  é a área do círculo, calculada como:  $A_c = \pi r^2$  e  $A_l$  é a área lateral do cilindro, computada por  $A_l = 2\pi r a$ , onde  $r$  é o raio e  $a$  a altura da lata em metros.

### Exemplos

<b>Entrada</b>
0.02
0.09
<b>Saída</b>
O VALOR DO CUSTO E = 1.38

## 9 Cálculo do Delta na Equação de Báskara



(+)

Fazer um programa para ler os valores dos coeficientes  $A$ ,  $B$  e  $C$  de uma equação quadrática e calcular e imprimir o valor do discriminante ( $\Delta$ ). O valor de  $\Delta$  é dado pela fórmula:  $\Delta = B^2 - 4AC$ .

### 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 do segundo grau. Cada valor ocorre em uma linha diferente na entrada.

### Saída

O programa deve imprimir uma linha contendo a frase: O VALOR DE DELTA E =  $X$ , onde  $X$  é o valor de delta computado pelo seu programa e deve conter no máximo 2 casas decimais. Após o valor de delta, o programa deve imprimir um caractere de quebra de linha: `"\n"`.

### Observação

Para imprimir uma expressão do tipo float com duas casas decimais, você deve usar a formatação `%.2f` na função de impressão `printf()`. Supondo que você usou a variável `delta` para armazenar o valor do discriminante, você poderia imprimir o conteúdo dessa variável com duas casas decimais usando a função de impressão do seguinte modo: `printf("%.2f\n", delta);`.

### Exemplo

Entrada
5
12
4
Saída
O VALOR DE DELTA E = 64.00

## 10 Cálculo do Determinante de uma Matriz Quadrada de Duas Dimensões



(+)

Fazer um programa tal que dados os quatro elementos de uma matriz  $2 \times 2$ , calcule e escreva o valor do determinante desta matriz.

### Entrada

O programa deve ler os quatro elementos  $a$ ,  $b$ ,  $c$  e  $d$  que formam uma matriz quadrada bidimensional. Há um valor por linha de entrada. Cada valor corresponde a um número real (float).

### Saída

O programa deve imprimir uma linha contendo a frase: O VALOR DO DETERMINANTE E =  $X$ , onde  $X$  é o valor do determinante computado pelo seu programa e deve conter no máximo 2 casas decimais. Após o valor do determinante, o programa deve imprimir um caractere de quebra de linha: `"\n"`.

### Observações

Dada uma matriz quadrada bidimensional  $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ , o determinante de  $M$ , denotado por  $\det(M)$  é definido como:  $\det(M) = ad - bc$ .

### Exemplo

Entrada
4
3
5
4
Saída
O VALOR DO DETERMINANTE E = 1.00

## 11 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

<b>Entrada</b>
75
<b>Saída</b>
O NUMERO E DIVISIVEL

## 12 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’.

### Exemplo

<b>Entrada</b>
29
<b>Saída</b>
O VALOR A PAGAR E = 100.00



## 13 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:  $\text{NOTA} = x$   $\text{CONCEITO} = y$ , 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

<b>Entrada</b>
3.4
<b>Saída</b>
NOTA = 3.4 CONCEITO = D

<b>Entrada</b>
6.0
<b>Saída</b>
NOTA = 6.0 CONCEITO = C

## 14 Volume da Pirâmide de Base Hexagonal



(+) O volume ( $V$ ) de uma pirâmide cuja base é um hexágono regular é computado pela

Equação 1:

$$v = \frac{1}{3} \cdot A_b \cdot h, \quad (1)$$

onde  $h$  é a altura da pirâmide e  $A_b$  é a área do hexágono que forma a base da pirâmide. A área do hexágono é computada pela Equação 2:

$$A_b = \frac{3 \cdot a^2 \cdot \sqrt{3}}{2}, \quad (2)$$

onde  $a$  é o comprimento de uma aresta do hexágono regular.

### Entrada

O programa deve ler uma linha com dois números float, separados entre si por um espaço. O primeiro número corresponde à altura da pirâmide e o segundo número corresponde a uma aresta do hexágono que forma a abase da pirâmide. Ambos são valores em metros.

### Saída

O programa deve emitir a frase: O VOLUME DA PIRAMIDE E =  $x$  METROS CUBICOS, onde  $x$  é o valor do volume da pirâmide em metros cúbicos e com duas casas decimais. Ao final da frase o programa deve imprimir o caractere de quebra de linha ( $\backslash n$ ).

### Exemplo

Entrada
12.0
8.0
Saída
O VOLUME DA PIRAMIDE E = 665.11 METROS CUBICOS

Entrada
0.45
0.23
Saída
O VOLUME DA PIRAMIDE E = 0.00 METROS CUBICOS

## 15 Quadrado de pares



(+)

1

Escreva um programa para ler um valor inteiro  $N$  e que gere o quadrado de cada um dos valores pares, de 1 até  $N$ , inclusive  $N$ , se for o caso.

### Entrada

A entrada conterà uma linha com um valor inteiro  $N$ ,  $5 < N < 2000$ .

### Saída

A saída deve conter, uma linha para cada quadrado computado. Em cada linha deve constar uma expressão do tipo  $x^2 = y$ , onde  $x$  é um número par e  $y$  é o seu valor elevado ao quadrado. Imediatamente após o valor de  $y$  deve aparecer o caractere de quebra de linha: ‘\n’.

### Exemplo

Entrada
6
Saída
$2^2 = 4$
$4^2 = 16$
$6^2 = 36$

---

<sup>1</sup>Fonte: Site do URI - <https://www.urionlinejudge.com.br/judge/pt/problems/view/1073>.

## 16 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’.

### 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.

<b>Entrada</b>
755.00
<b>Saída</b>
SALARIO COM REAJUSTE = 981.50

<b>Entrada</b>
265.32
<b>Saída</b>
SALARIO COM REAJUSTE = 397.98

## 17 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

## 18 Soma de progressão aritmética



(+)

Nerdilton adora Matemática, mas ele não tem a memória muito boa para memorizar fórmulas. Geralmente ele entende como a fórmula funciona e deriva o valor de saída da fórmula. Ele precisa responder a um conjunto de exercícios que pedem a soma dos  $n$  primeiros elementos de uma progressão aritmética, dados o primeiro elemento da progressão e a razão. Escreva um programa que seja capaz de encontrar a soma dos  $n$  primeiros termos de uma progressão aritmética sem usar a fórmula fechada da soma dos elementos de uma progressão aritmética. Lembre-se que uma progressão aritmética com valor inicial  $a_1$  e razão  $r$  é a sequência formada por:  $a_1, a_1 + r, a_1 + 2r, \dots, a_1 + (n - 1)r$ . O programa precisa computar a soma dos valores dessa sequência.

### Entrada

A entrada conterá uma linha com três números inteiros separados entre si por um caractere de espaço. O primeiro número corresponde ao valor inicial da progressão aritmética, o segundo, corresponde à razão da mesma e o terceiro corresponde ao número  $n$  de elementos da progressão.

### Saída

O programa deve imprimir o valor da soma dos  $n$  primeiros elementos da progressão, seguido de um caractere de quebra de linha.

### 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.

Entrada
2 4 5
Saída
50

Entrada
7 -2 10
Saída
-20

## 19 Somatório simples



(+)

Faça um programa que leia um valor  $n$ , inteiro e positivo, calcule e mostre a seguinte soma:

$$S = \sum_{k=1}^n \frac{1}{k} = 1 + 1/2 + 1/3 + 1/4 + \dots + 1/n \quad (3)$$

### Entrada

O programa deve ler um número inteiro positivo e maior que 1.

### Saída

O programa deve apresentar uma linha contendo o valor final do somatório com 6 casas decimais. Caso o número lido não atenda as especificações da entrada, o programa deve apresentar a mensagem: "Numero invalido!".

### Observações

Use precisão dupla para o cálculo de  $S$ .

### Exemplo

Entrada
10
Saída
2.928968

## 20 Tempo em segundos



(+)

Fazer um programa que leia um valor de tempo expresso em horas, minutos e segundos e que converta esse tempo para um valor em segundos.

### Entrada

O programa deve ler três linhas na entrada. A primeira contém um valor em horas, a segunda, contém um valor em minutos e a terceira, contém um valor em segundos. Os valores são todos números inteiros.

### Saída

O programa deve imprimir uma linha contendo a frase: O TEMPO EM SEGUNDOS E =  $X$ , onde  $X$  é o valor do tempo convertido em segundos. Após o valor do tempo em segundos, o programa deve imprimir um caractere de quebra de linha: ‘\n’.

### Exemplos

Entrada
5
12
1
Saída
O TEMPO EM SEGUNDOS E = 18721



## 21 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:

$$NF = 0.7 \cdot MP + 0.15 \cdot ML + 0.15 \cdot NT \quad (4)$$

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	-1	-1	-1	-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													

## 22 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

## 23 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

## 24 Arredondamento



(++)

Escreva um algoritmo que leia um número real e realize o arredondamento deste número usando 1, 2 e 3 casas decimais. A apresentação do número deve conter, obrigatoriamente 6 casas decimais. As casas decimais posteriores ao dígito arredondado devem conter o valor 0.

### Considerações

O arredondamento de um número é uma operação que elimina algarismos de menor significância. A regra de arredondamento aplica-se nos algarismos situados após a posição da quantidade de casas decimais desejada. Ou seja, o processo de arredondamento do número 12.318215 considerando 1 casa decimal deve avaliar os números 18215. Para 2 casas decimais deve-se avaliar os números 8215, e assim por diante.

- Se o algarismo seguinte for menor que 5, então o anterior não se modifica
- Se o algarismo seguinte for maior ou igual a 5, então o anterior é incrementado

### Entrada

O programa deve ler 1 valor real.

### Saída

O programa deve imprimir a primeira linha contendo o número arredondado com 1 casa decimal, a segunda com 2 casas decimais e a terceira com 3 casas decimais.

### Exemplo

<b>Entrada</b>
3.1752
<b>Saída</b>
3.200000
3.180000
3.175000
<b>Entrada</b>
0.1825
<b>Saída</b>
0.200000
0.180000
0.183000

## 25 Cálculo da Área de um Triângulo



(++)

Desenvolver um algoritmo para ler os comprimentos dos três lados de um triângulo ( $L_1$ ,  $L_2$  e  $L_3$ ) e calcular a área do triângulo.

### Considerações

A área de um triângulo pode ser computada pela fórmula:

$$A = \sqrt{T(T - L_1)(T - L_2)(T - L_3)}$$

onde

$$T = \frac{L_1 + L_2 + L_3}{2}$$

A função `sqrt()` computa a raiz quadrada de uma expressão. Para usar essa função você deve incluir o arquivo de cabeçalho `math.h`, inserindo a seguinte diretiva de pré-processamento logo no início do seu arquivo com o programa em C: **#include** <math.h>

### Entrada

O programa deve ler três valores reais na entrada, cada um correspondendo ao comprimento de um lado do triângulo. Cada valor ocorre em uma linha diferente na entrada.

### Saída

O programa deve imprimir uma linha contendo a frase: A AREA DO TRIANGULO E =  $X$ , onde  $X$  é o valor da área do triângulo e deve conter no máximo 2 casas decimais. Após o valor da área do triângulo, o programa deve imprimir um caractere de quebra de linha: `"\n"`.

### Exemplo

Entrada
4
5
6
Saída
AREA DO TRIANGULO E = 9.92

## 26 Custo Final de um Carro



(++)

O custo ao consumidor de um carro novo é a soma do custo de fábrica com a porcentagem do distribuidor e dos impostos (aplicados ao custo de fábrica). Supondo que a porcentagem do distribuidor seja de  $x\%$  do preço de fábrica e os impostos de  $y\%$  do preço de fábrica, fazer um programa para ler o custo de fábrica de um carro, a porcentagem do distribuidor e o percentual de impostos, calcular e imprimir o custo final do carro ao consumidor.

### Entrada

O programa deve ler três valores na entrada: o preço de fábrica do carro, o percentual do distribuidor e o percentual de impostos. Cada valor aparece em uma linha de entrada. Todos os valores são do tipo float.

### Saída

O programa deve imprimir uma linha, contendo a frase O VALOR DO CARRO E = Z, onde Z é o valor do preço final do carro ao consumidor. O valor de Z deve ter duas casas decimais. Após imprimir o valor do preço final, o programa deve imprimir o caractere de quebra de linha '\n'.

### Observações

### Exemplo

Entrada
25000
12
30
Saída
O VALOR DO CARRO E = 35500.00

## 27 Decolagem



(++)

Escrever um algoritmo que leia a massa (em toneladas) de um avião, sua aceleração ( $m/s^2$ ) e o tempo (s) que levou do repouso até a decolagem. O programa deve calcular e escrever a velocidade atingida (Km/h), o comprimento da pista (m) e o trabalho mecânico realizado (J) no momento da decolagem.

### Dicas

- $v$  = velocidade;  $a$  = aceleração;  $t$  = tempo;
  - $m$  = massa;
  - $s$  = espaço percorrido;
  - $W$  = trabalho mecânico realizado;
  - Um double deve ser lido com "%lf"
- $1 \text{ m/s} = 3,6 \text{ Km/h}$ ;
  - $v = a * t$ ;
  - $s = \frac{at^2}{2}$ ;
  - $W = \frac{mv^2}{2}$ ;
  - A massa utilizada no trabalho é em Kg

### Entrada

O programa deve ler três linhas de entrada. A primeira linha contém um valor do tipo *double* representando a massa do avião em toneladas. A segunda linha, contém um valor do tipo *double* correspondente à aceleração de avião. A terceira, linha contém um valor do tipo *double* correspondente ao tempo em segundos gasto na decolagem.

### Saída

O programa deve imprimir três linhas. A primeira, contém a frase: VELOCIDADE =  $x$ , onde  $x$  é o valor da velocidade do avião em Km/h. A segunda, contém a frase: ESPACO PERCORRIDO =  $y$ , onde  $y$  corresponde ao espaço em metros percorrido pelo avião durante a decolagem. A terceira linha contém a frase: TRABALHO REALIZADO =  $z$ , onde  $z$  corresponde ao valor do trabalho em Joules, realizado pelo avião durante a decolagem. Os valores de  $x$ ,  $y$  e  $z$  devem ser do tipo *double* e devem conter duas casas decimais e após esses valores deve vir o caractere de quebra de linha  $\backslash n$ .

### Exemplos

Entrada
10
5
90
Saída
VELOCIDADE = 1620.00
ESPACO PERCORRIDO = 20250.00
TRABALHO REALIZADO = 1012500000.00



<b>Entrada</b>
3
30
25
<b>Saída</b>
VELOCIDADE = 2700.00
ESPACO PERCORRIDO = 9375.00
TRABALHO REALIZADO = 843750000.00

## 28 Distância entre dois pontos



(++)

Dados dois pontos  $A$  e  $B$ , cujas coordenadas  $A(x_1, y_1)$  e  $B(x_2, y_2)$  serão informadas via teclado, desenvolver um programa que calcule a distância entre  $A$  e  $B$ .

### Entrada

O programa deve ler os quatro valores reais correspondendo às coordenadas dos dois pontos :  $x_1, y_1, x_2, y_2$ , nessa ordem, e um valor por linha.

### Saída

O programa deve imprimir uma linha contendo a frase: A DISTANCIA ENTRE A e B =  $X$ , onde  $X$  é o valor da distância entre os dois pontos e deve conter no máximo 2 casas decimais. Após o valor da distância, o programa deve imprimir um caractere de quebra de linha: ‘\n’.

### Observações

A distância entre dois pontos é computada pela fórmula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Você pode usar a função `sqrt()` para calcular a raiz quadrada na fórmula da distância. Para computar o quadrado de um valor  $x$  você pode usar a função `pow(x,2)`. Para usar essas funções, você precisa colocar `#include <math.h>` no início do texto do programa.

### Exemplo

Entrada
3
4
5
6
Saída
A DISTANCIA ENTRE A e B = 2.83

## 29 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

## 30 Gerador de tabuada



(++)

Escreva um programa em linguagem C 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  $K$  valores, iniciando em  $i$  em incrementos de  $s$ .

### 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 formato: " $n$  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

## 31 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

## 32 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:  $k$ ", onde  $k$  é o tamanho do maior segmento crescente encontrado.

### Exemplo

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

### 33 Maior segmento igual de uma sequência



(++) (POLI 87) Dados  $n$  e uma sequência de  $n$  números inteiros, determinar quantos segmentos de números iguais consecutivos compõem essa sequência.

#### 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 de numeros iguais e:  $k$ ", onde  $k$  é o tamanho do maior segmento crescente encontrado.

#### Exemplo

<b>Entrada</b>
2
1 1
<b>Saída</b>
O comprimento do segmento de numeros iguais e: 2
<b>Entrada</b>
7
0 0 2 5 5 5 6
<b>Saída</b>
O comprimento do segmento de numeros iguais e: 3

## 34 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!



## 35 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 contrário. 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

## 36 Número primo



(+)

Faça um programa que leia um número inteiro  $N$  e informe se o número é primo ou não.

### Entrada

O programa deverá ler um número inteiro  $N$ .

### Saída

O programa deverá apresentar a mensagem "PRIMO" caso  $N$  seja primo e "NAO PRIMO" caso contrário. Caso o valor de  $N$  não seja um número positivo, o programa deve apresentar a mensagem "NUMERO INVALIDO"

### Exemplo

Entrada
7
Saída
PRIMO

Entrada
9
Saída
NAO PRIMO

## 37 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

## 38 Quatro Algarismos



(++)

Dado um número inteiro de três algarismos, construir outro número inteiro de quatro algarismos de acordo com a seguinte regra: os três primeiros algarismos, contados da esquerda para a direita são iguais ao número dado. O quarto algarismo é um dígito de controle calculado da seguinte forma: primeiro algarismo + segundo algarismo  $\times 3$  + terceiro algarismo  $\times 5$ . O dígito de controle é igual ao resto da divisão dessa soma por 7.

### Entrada

O programa deve ler uma linha de dados contendo apenas um número com três algarismos.

### Saída

O programa deve imprimir uma linha contendo a frase: O NOVO NUMERO E =  $X$ , onde  $X$  é o novo número inteiro com quatro algarismos, seguido por um caractere de quebra de linha: ‘\n’.

### Observações

### Exemplo

<b>Entrada</b>
123
<b>Saída</b>
O NOVO NUMERO E = 1231

## 39 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  $ax^2 + bx + c$ ,  $\Delta$  (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

<b>Entrada</b>
15 17 89
<b>Saída</b>
RAIZES IMAGINARIAS

## 40 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 várias 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

### Observações

Para ler uma linha com os três valores, utilize a função **scanf**: `scanf("%d %f %f", &A, &B, &C);` e, em seguida, a função `getchar()`. A função `getchar()` 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	

## 41 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 sequências 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 sequências 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 sequência 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



## 42 Soma dos 3 menores



(++)

Fazer um programa para ler 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

## 43 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

<b>Entrada</b>
30022001
<b>Saída</b>
Data invalida!
<b>Entrada</b>
12092017
<b>Saída</b>
12 de setembro de 2017

## 44 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  $k = 1, \dots, \infty$  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:

$$r_k = \frac{r_{k-1} + \frac{n}{r_{k-1}}}{2} \quad (5)$$

### Entrada

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

### Saída

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

### Exemplo

Entrada	
2	
0.00001	
Saída	
r:	1.500000000, err: 0.250000000
r:	1.416666667, err: 0.006944444
r:	1.414215686, err: 0.000006007

## 45 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

## 46 Companhia de Teatro



(+++)

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

### Entrada

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

### Saída

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

"Melhor valor final: xxx.xx"

"Lucro: xxx.xx"

"Numero de ingressos: xx"

### Observações

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

## Exemplo

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

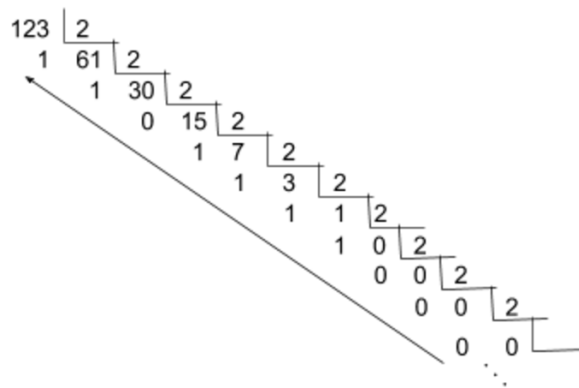
Entrada
0.5 0 2
Saída
V: 0.00, N: 145, L: -207.25 V: 1.00, N: 90, L: -114.50 V: 2.00, N: 30, L: -141.50 Melhor valor final: 0.00 Lucro: 0.00 Numero de ingressos: 0

## 47 Conversão de decimal para binário



(+++)

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



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

### Entrada

O programa deve ler um número inteiro qualquer.

### Saída

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

### Observações

Neste problema, todos os números binários deverão conter 8 bits. O número zero (em decimal), por exemplo, tem sua representação binária 00000000. O número 1 = 00000001, o 2 = 00000010 e assim por diante.

### Exemplo

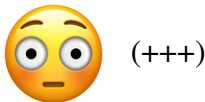
Entrada
0
Saída
00000000

Entrada
123
Saída
01111011

<b>Entrada</b>
128
<b>Saída</b>
10000000



# 48 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.

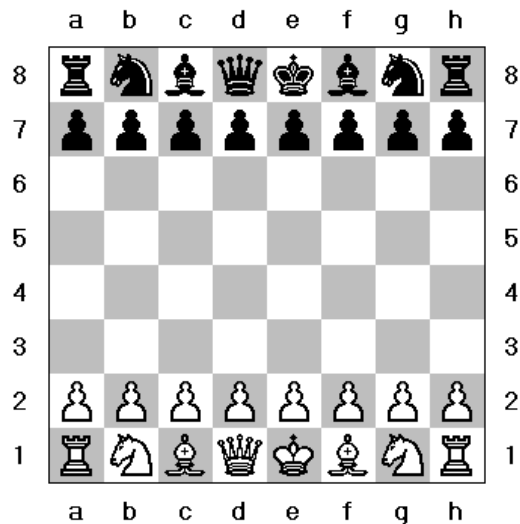


Figura 1: Tabuleiro de xadrez.

## 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

## 49 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 bruto e o imposto líquido;
- A mensagem `IMPOSTO A RECEBER`, se o valor de  $w$  for negativo ou a mensagem `IMPOSTO QUITADO`, se  $w$  for igual a zero, ou a mensagem `IMPOSTO A PAGAR`, caso  $w$  for maior que zero.

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

### Exemplo

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

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

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

## 50 Índices da matriz inferior



(+++)

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

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \quad (6)$$

### Entrada

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

### Saída

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

### Exemplo

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

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

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

## 51 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.

### 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  $c_2$  n", onde  $h$  é uma hipotenusa inteira,  $c_1$  e  $c_2$  são seus catetos inteiros.

### 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

## 52 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%			

## 53 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
$1 * 1 * 1 = 1$
$2 * 2 * 2 = 3 + 5$
$3 * 3 * 3 = 7 + 9 + 11$
$4 * 4 * 4 = 13 + 15 + 17 + 19$



## 54 Número Invertido



(+++)

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

### Entrada

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

### Saída

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

### Exemplos

Entrada
123
Saída
321
Entrada
987
Saída
789

## 55 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

<b>Entrada</b>
6
<b>Saída</b>
6 = 1 + 2 + 3 = 6 (NUMERO PERFEITO)

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

## 56 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

## 57 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

## 58 Sistemas de Equação Linear



(+++)

Dado um sistema de equações lineares do tipo:

$$ax + by = c$$

$$dx + ey = f$$

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

### Entrada

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

### Saída

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

### Exemplo

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

## 59 Triângulo ou trapézio?



(+++)

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

Perimetro = XX.X

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

Area = XX.X

### Entrada

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

### Saída

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

### Observações

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

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

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

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

## 60 Valor em Notas e Moedas



(+++)

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

### Entrada

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

### Saída

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

### Exemplo

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

## 61 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.

### 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



## 62 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  $N$ .

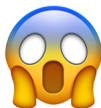
### Saída

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

## 63 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)

## 64 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:

$$\cos(x) = \sum_{n=0}^N \frac{(-1)^n x^{2n}}{(2n)!} = \frac{x^0}{0!} - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + \frac{(-1)^N x^{2N}}{(2N)!} \quad (7)$$

, 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 " $\cos(x) = y$ ", 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  $2\pi$ . Use a constante `M_PI` da biblioteca `<math.h>`. Como sugestão de desafio à solução do problema, tente escrever um algoritmo que use apenas um laço de repetição.

### Exemplo

<b>Entrada</b>
2
9
<b>Saída</b>
<code>cos (2.00) = -0.416147</code>
<b>Entrada</b>
3.14
6
<b>Saída</b>
<code>cos (3.14) = -0.999899</code>
<b>Entrada</b>
1
4
<b>Saída</b>
<code>cos (1.00) = 0.540303</code>

## 65 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:

$$e^x = \sum_{n=0}^N \frac{x^n}{(n)!} = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \dots + \frac{x^N}{(N)!} \quad (8)$$

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

## 66 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:

$$\sin(x) = \sum_{n=0}^N \frac{(-1)^n x^{2n+1}}{(2n+1)!} = \frac{x^1}{0!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + \frac{(-1)^N x^{2N+1}}{(2N+1)!} \quad (9)$$

, 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  $2\pi$ . Use a constante `M_PI` da biblioteca `<math.h>`. Como sugestão de desafio à solução do problema, tente escrever um algoritmo que use apenas um laço de repetição.

### Exemplo

<b>Entrada</b>
2
9
<b>Saída</b>
seno(2.00) = 0.909297
<b>Entrada</b>
3.14
6
<b>Saída</b>
seno(3.14) = 0.001614
<b>Entrada</b>
1
4
<b>Saída</b>
seno(1.00) = 0.841471

## 67 Decomposição em fatores primos



(+++++)

Todo número natural maior que 1 pode ser escrito na forma de uma multiplicação em que todos os fatores são números primos. Por exemplo, o número 36 pode ser representado pela multiplicação  $2 \times 2 \times 3 \times 3$ . A essa representação multiplicativa dá-se o nome de Decomposição em Fatores Primos ou Fatoração, que é um produto de fatores primos. O processo de fatoração de  $N$  segue um método prático de divisões sucessivas pelo seu menor fator primo. A cada passo, deve-se encontrar o menor divisor primo do quociente da divisão anterior. A Figura 2 mostra dois exemplos 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 2: Exemplo de fatoração dos números 36 e 120.

### Entrada

O programa deve ler um número inteiro  $N$ .

### Saída

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

### Exemplo

<b>Entrada</b>
554
<b>Saída</b>
554 = 2 x 277

<b>Entrada</b>
-1
0
120
<b>Saída</b>
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

<b>Entrada</b>
2
<b>Saída</b>
2 = 2