

Listas de Exercícios

1 Introdução: Conceitos Básicos e Estruturas de Decisão

1. Identifique o tipo dos dados:

(a) numérico inteiro;

(b) numérico real;

(c) lógico;

(d) literal.

<input type="checkbox"/> True	<input type="checkbox"/> 45.0	<input type="checkbox"/> 2345	<input type="checkbox"/> 0.0
<input type="checkbox"/> 'lista'	<input type="checkbox"/> 'c*d'	<input type="checkbox"/> -456	<input type="checkbox"/> '0 1 2 3'
<input type="checkbox"/> -0.204	<input type="checkbox"/> 30.12	<input type="checkbox"/> '56'	<input type="checkbox"/> -15.0962
<input type="checkbox"/> ''	<input type="checkbox"/> -334.89	<input type="checkbox"/> -12E3	<input type="checkbox"/> false
<input type="checkbox"/> 0	<input type="checkbox"/> 708	<input type="checkbox"/> 'false'	<input type="checkbox"/> -22
<input type="checkbox"/> 32	<input type="checkbox"/> -1.3	<input type="checkbox"/> 'BOLA'	<input type="checkbox"/> '&'
<input type="checkbox"/> 'TRUE'	<input type="checkbox"/> -4.5E-301	<input type="checkbox"/> 0.56E2	<input type="checkbox"/> 'a'
<input type="checkbox"/> 'False'	<input type="checkbox"/> 1.23E+02	<input type="checkbox"/> '1.23E+02'	<input type="checkbox"/> '0.25'

2. Faça um algoritmo que leia dois números inteiros n1 e n2 e mostre o resultado da divisão inteira e o resto da divisão entre eles, onde n1 é o numerador e n2 é o denominador. Observação: o denominador não pode ser zero.

3. Supondo que a, b e c sejam variáveis inteiras com valores iguais a 5, 10 e -2, respectivamente, faça um algoritmo que mostre o resultado das seguintes expressões:

(a) $(b / a) \text{ MOD } c$;

(b) $\sqrt{c^2 + \frac{b}{a}} + 3$

4. O preço de um automóvel é calculado pela soma do preço de fábrica com o preço dos impostos (45% do preço de fábrica) e a percentagem do revendedor (28% do preço de fábrica). Faça um algoritmo que leia o nome do automóvel e o preço de fábrica e, posteriormente calcule e mostre o preço final do veículo e seu nome.

5. Faça um algoritmo que leia separadamente o nome e o sobrenome de uma pessoa e, posteriormente, concatene-os e mostre o resultado na tela.
6. Faça um algoritmo que leia as notas de duas avaliações (N1 e N2) feitas por um aluno, bem como os valores para os respectivos pesos de cada avaliação (P1 e P2). Feito isso, o programa deverá calcular e mostrar o resultado da média aritmética, da média ponderada e da média geométrica, todas formatadas com duas casas decimais.
7. Faça um algoritmo que leia o nome, o preço unitário e a quantidade de três produtos quaisquer e, posteriormente, calcule e apresente o valor total da compra.
8. Faça um algoritmo que leia o salário bruto de um funcionário, o valor do desconto do INSS e do desconto do Imposto de Renda e, posteriormente, calcule e mostre o salário líquido deste funcionário.
9. Faça um algoritmo que calcule e mostre a velocidade média de um determinado veículo.
10. Faça um algoritmo que leia o preço por Kg de um determinado produto e o peso desejado pelo seu comprador. O programa deve calcular e apresentar o total a ser pago pela quantia do referido produto.
11. Faça um algoritmo que solicite uma temperatura em F^0 (Fahrenheit) e faça a conversão da mesma para $^{\circ}\text{C}$ (Celsius), mostrando o resultado na tela. Lembre-se: $C = [(F - 32) * 5] / 9$.
12. Faça um algoritmo que calcule a quantidade necessária de latas de tinta para pintar uma parede de 5m de largura por 2m de altura. Considere que o consumo de tinta é de 3 litros por metro quadrado e a quantidade de tinta por lata é de 2 litros.
13. Faça um algoritmo que leia 2 valores, a operação desejada (adição, subtração, multiplicação ou divisão) e, em seguida, calcule e apresente o resultado.
14. Num determinado estado o Detran cobra um imposto de 3% para veículos com até 10 anos de uso e 2% para veículos com mais antigos. Faça um algoritmo que leia o ano do carro, seu preço e calcule o imposto.
15. Um imposto é calculado com base na tabela abaixo. Faça um algoritmo que leia o valor base, calcule e mostre o imposto a ser pago:
 - Até 1.200,00 - isento
 - De 1.201,00 a 3.500,00 - 10%
 - Acima de 3.500,00 - 20%.

16. Faça um algoritmo que leia nomes de cidades e suas respectivas populações. O processamento deve parar quando o nome da cidade for “fim”. Ao final, deve ser mostrado: quantidade de cidades digitadas, quantidade de vezes que foi digitada a cidade “Campos”, a maior população informada.
17. Faça um algoritmo que leia uma lista de letras terminada pela letra “Z”. Ao final o algoritmo deve mostrar a quantidade lida de cada vogal.
18. Faça um algoritmo que leia um número N e uma lista de N números inteiros positivos e mostre o maior número da lista.
19. Faça um algoritmo que leia N e uma lista de N números e mostre a soma de todos os números da lista.
20. Faça um algoritmo que leia um número e, em seguida, leia uma lista de números até achar um igual ao primeiro número lido. Mostre todos os números lidos.
21. Faça um algoritmo que leia uma lista de números terminada pelo número 9999 e mostre cada número lido. Ao final, o algoritmo deve mostrar a média aritmética de todos os números da lista.
22. Faça um algoritmo que leia um número e divida-o por dois (sucessivamente) até que o resultado seja menor que 1. Mostre o resultado da última divisão efetuada.
23. Faça um programa que leia o valor de dois números inteiros, e guarde cada um deles em uma variável diferente, por exemplo v1 e v2. Depois, o seu programa deve colocar o número que está em v1 na variável v2 e o número que está em v2 na variável v1, ou seja, realizar a troca de valores das variáveis. Escreva na tela os números armazenados em cada uma das variáveis antes e depois da troca. Lembre-se em cada variável só cabe UM valor! Utilize quantas outras variáveis você quiser para fazer este programa.
24. Fornecidos separadamente o nome e o sobrenome pelo usuário, desenvolva um programa que exiba o nome e o sobrenome concatenados.
25. Faça um programa que leia três números e os imprima na ordem contrária à que foram digitados. Por exemplo: se o usuário digitar 4, 2, 9 o programa deve imprimir: 9, 2, 4.
26. Construa um algoritmo que lê um número de 1 a 7 e informa o nome da nota musical (dó, ré, mí, fá, sol, lá, sí) correspondente, sendo a nota musical dó a número 1. Se o número não corresponder a uma nota, é mostrada uma mensagem de erro.
27. Criar um algoritmo que calcule o produto de dois números lidos, através de somas sucessivas. Suponha que os números lidos sejam positivos.
Por exemplo: $3 * 4 = 4 + 4 + 4$

Resultado = 12

28. Avalie os algoritmos a seguir e responda quais serão os valores armazenados pelas variáveis desses algoritmos ao final de sua execução:

```
(a) main() {
    int a, b, c;

    a = 100;
    b = 200;
    c = (a / b) + 2;
    b = c \% (a + 3);
    a = b + c;
    printf (\\%d \%d \%d\\n", a, b, c);
}

(b) main() {
    float x, y, z;

    x = 2.5;
    y = 3.5;
    z = ((x + y) - 2) * (3 + x);
    y = y + 1;
    x = y + 1;
    y = x + y;
    z = z + y + x;

    printf ("%f %f %f\\n", x, y, z);
}

(c) main() {
    int s, r, u;
    float x;

    s = 3;
    r = s;
    u = s * r;
    x = u / 5 + r;

    printf ("%d \%d \%d %f\\n", s, r, u, x);
}

(d) main() {
    float m, p1 , p2 , p3;

    p1 = 10;
    p2 = 10;
    p3 = 10;
    m = p1 + p2 + p3/3;
```

```

        printf ("%%.2f \%.2f \%.2f \%.2f\n", p1, p2, p3, m);
    }

(e) main() {
    float m, p1 , p2 , p3;

    p1 = 10;
    p2 = 10;
    p3 = 10;
    m = (p1 + p2 + p3)/3;

    printf ("%%.2f \%.2f \%.2f \%.2f\n", p1, p2, p3, m);
}

(f) main() {
    float m, p1 , p2 , p3;

    p1 = 10;
    p2 = 10;
    p3 = 10;
    m = p1 + (p2 + p3)/3;

    printf ("%%.2f \%.2f \%.2f \%.2f\n", p1, p2, p3, m);
}

(g) main() {
    int k, y, w;

    k = 34;
    y = 4;
    w = ((k \% 4)/2) + (10 - y);
    k = w * 2;

    printf ("%d %d %d\n", k, y, w);
}

```

29. Supondo que as variáveis NOME, PROF, ID e SALARIO sejam utilizadas para armazenar nome, código da profissão, idade e salário de uma pessoa, escreva o conjunto de declarações necessário para criar estas variáveis e associar às mesmas respectivos tipos pré-definidos.

30. Indique qual o resultado das expressões aritméticas abaixo, sendo:

$x = 8.0$ $y = 2$ $z = 4.0$ $a = 6$ $b = 7.5$ $c = 3.2$
 $d = 12$ $p = 4$ $q = 3$ $r = 10$ $s = 2.5$

(a) $x + y - z * a$

(b) d/y

(c) $d\%y$

(d) y/d

(e) $y\%d$

(f) $((z/a) + b * a) - d$

(g) $100 * (q/p) + r$

(h) $p * (r\%q) - q/2$

31. Faça um programa que receba dois números inteiros, calcule e imprima a soma desses números.

32. Faça um programa que receba dois números reais, calcule e imprima a subtração do primeiro número pelo segundo.

33. Faça um programa que receba dois números inteiros, calcule e imprima a divisão do primeiro pelo segundo.

34. Faça um programa que receba dois números inteiros, calcule e imprima:

(a) soma dos dois números;

(b) subtração do primeiro pelo segundo;

(c) subtração do segundo pelo primeiro;

(d) multiplicação dos dois números;

(e) divisão do primeiro pelo segundo;

(f) divisão do segundo pelo primeiro;

(g) quociente inteiro da divisão do primeiro pelo segundo;

(h) quociente inteiro da divisão do segundo pelo primeiro;

(i) resto da divisão do primeiro pelo segundo;

- (j) resto da divisão do segundo pelo primeiro.
35. Faça um programa que receba três notas de um aluno, calcule e imprima a média aritmética entre essas notas.
36. Faça um programa que receba duas notas de um aluno e seus respectivos pesos, calcule e imprima a média ponderada dessas notas.
37. Faça um programa que receba um número real, calcule e imprima:
- (a) a parte inteira do número;
 - (b) a parte fracionária do número;
 - (c) arredondamento do número.
38. Faça um programa que receba a medida de um ângulo em graus, calcule e imprima o seno, o cosseno e a tangente desse ângulo.
39. Faça um programa que receba um número inteiro, calcule e imprima a tabuada desse número.
40. Escreva um programa que receba o valor de um depósito e o valor da taxa de juros. Calcule e imprima o valor do rendimento e o valor total depois do rendimento.
41. Escreva um programa que calcule o volume de uma esfera de raio R , onde R é fornecido pelo usuário.
42. Faça um programa que, tendo como dados de entrada dois pontos quaisquer do plano, $P(x_1, y_1)$ e $Q(x_2, y_2)$, imprima a distância entre eles. A fórmula da distância é $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.
43. Escreva um programa que calcule a quantidade de latas de tintas necessárias e o custo para pintar tanques cilíndricos de combustível, em que são fornecidos a altura e o raio desse cilindro. Sabe-se que a lata de tinta custa R\$50,00; cada lata contém 5 litros e cada litro de tinta pinta $3m^2$.
44. Faça um programa que receba dois números, calcule e imprima um elevado ao outro.
45. Faça um programa que receba um número inteiro, calcule e imprima a raiz quadrada desse número e, esse número elevado ao quadrado.

46. Escreva um programa que receba o valor do salário de um funcionário e o valor do salário mínimo. Calcule e imprima quantos salários mínimos ganha esse funcionário.
47. Faça um programa que receba uma determinada hora (hora e minuto separados por ponto), calcule e imprima essa hora em minutos.
48. Faça um programa que calcule e imprima:
- área de um triângulo;
 - área de um quadrado;
 - área de um círculo;
 - área de um trapézio;
 - área de um retângulo;
 - área de um losango.
49. Faça um programa que receba o salário de um funcionário, calcule e imprima o valor do imposto de renda a ser pago, sabendo que o imposto equivale a 5% do salário.
50. Faça um programa que receba o salário de um funcionário, calcule e imprima o novo salário sabendo-se que este sofreu um aumento de 22%.
51. Sabe-se que o quilowatt de energia custa um quinto do salário mínimo. Faça um programa que receba o valor do salário mínimo e a quantidade de quilowatts gasta por uma residência. Calcule e imprima:
- (a) o valor, em reais, de cada quilowatt;
 - (b) o valor, em reais, a ser pago por essa residência;
 - (c) o novo valor a ser pago por essa residência, a partir de um desconto de 15%.
52. Faça um programa que receba o peso de uma pessoa, um valor inteiro, calcule e imprima:
- (a) o peso dessa pessoa em gramas;
 - (b) se essa pessoa engordar 5%, qual será seu novo peso em gramas.

53. Faça um programa que leia três valores inteiros, determine e imprima o menor deles.
54. Faça um programa que, dado um número inteiro, verifique se este número é par.
55. Dados três números (x,y,z), faça um programa que imprima-os em ordem decrescente.
56. Dados três valores (x,y,z), verifique se eles podem ser comprimentos dos lados de um triângulo, e caso afirmativo, verificar se o triângulo é equilátero, isósceles ou escaleno. Se não puderem formar um triângulo, escreva uma mensagem.

Propriedade: O comprimento de cada lado de um triângulo é menor do que a soma dos comprimentos dos outros dois lados.

- Definição 1: Chama-se triângulo equilátero ao triângulo que tem os comprimentos dos três lados iguais.
 - Definição 2: Chama-se triângulo isósceles ao triângulo que tem os comprimentos de dois lados iguais.
 - Definição 3: Chama-se triângulo escaleno ao triângulo que tem os comprimentos de seus três lados diferentes.
57. Tendo como dados de entrada a altura (h), o peso e o sexo de uma pessoa, faça um programa que verifique se a pessoa está acima, abaixo ou no seu peso ideal. Utilize as fórmulas abaixo para o cálculo do peso ideal:

- para homens: $72.7 * h - 58$
- para mulheres: $62.1 * h - 44.7$

58. Elabore um programa que, dada a idade de um nadador, classifique-o em uma das seguintes categorias:

- infantil A: 5 - 7 anos;
- infantil B: 8 - 10 anos;
- juvenil A: 11 - 13 anos;
- juvenil B: 14 - 17 anos;
- sênior: maiores de 18 anos.

59. Escreva um programa que leia o código de um determinado produto e mostre a sua classificação. Utilize a tabela abaixo como referência:

Código	Classificação
1	Alimento não-perecível
2, 3 ou 4	Alimento perecível
5 ou 6	Vestuário
7	Higiene pessoal
8 até 15	Limpeza e utensílios domésticos
Qualquer outro código	Inválido

60. A solução (x,y) para o sistema de equações lineares abaixo:

$$\begin{cases} a * x + b * y = u \\ c * x + d * y = v \end{cases}$$

é dada por

$$x = \frac{d}{a*d-b*c} * u - \frac{b}{a*d-b*c} * v$$

e

$$y = \frac{-c}{a*d-b*c} * u + \frac{a}{a*d-b*c} * v$$

Escreva um programa que leia os parâmetros a, b, c, d, u, v do sistema, calcule e imprima a solução x, y do sistema.

61. Escreva um programa que calcule o que deve ser pago por um produto, considerando o preço normal de etiqueta e a escolha da condição de pagamento. Utilize os códigos da tabela a seguir para ler qual condição de pagamento escolhida e efetuar o cálculo adequado.

Código	Condição de pagamento
1	À vista em dinheiro ou cheque, recebe 10% desconto
2	À vista no cartão de crédito, recebe 5% desconto
3	Em 2 vezes, preço normal de etiqueta sem juros
4	Em 3 vezes, preço normal de etiqueta mais juros de 10%

(a) Assinale com um X os identificadores válidos para nomes de variáveis:

<input type="checkbox"/> valor	<input type="checkbox"/> A:B	<input type="checkbox"/> 5A	<input type="checkbox"/> x2
<input type="checkbox"/> B*D	<input type="checkbox"/> M{A}	<input type="checkbox"/> 3x4	<input type="checkbox"/> NOTA[1]
<input type="checkbox"/> O!!	<input type="checkbox"/> XYZ	<input type="checkbox"/> X-Y	<input type="checkbox"/> nota* aluno
<input type="checkbox"/> 'nota'	<input type="checkbox"/> E(13)	<input type="checkbox"/> ail	<input type="checkbox"/> Salario_Bruto
<input type="checkbox"/> VAR-ABC	<input type="checkbox"/> KM/H	<input type="checkbox"/> A312	<input type="checkbox"/> 3/1
<input type="checkbox"/> #12	<input type="checkbox"/> _Y	<input type="checkbox"/> E&E:	<input type="checkbox"/>
<input type="checkbox"/> a14b56	<input type="checkbox"/> _a	<input type="checkbox"/> 'aula'	<input type="checkbox"/> aula
<input type="checkbox"/> 2abx	<input type="checkbox"/> 2-4	<input type="checkbox"/> a-b	<input type="checkbox"/> a_b

- (b) Fazer um programa que imprima a média aritmética dos números 8, 9 e 7. A média dos números 4, 5 e 6. A soma das duas médias e a média das médias.
- (c) Ler um número inteiro e imprimir seu sucessor e seu antecessor.
- (d) Receber um valor qualquer do teclado e imprimir esse valor com reajuste de 10%.
 - i. Informar três números inteiros e imprimir a média.
- (e) Informe o tempo gasto numa viagem (em horas), a velocidade média e calcular a distância.
- (f) Ler um número inteiro e imprimir seu quadrado.
- (g) Informar um saldo e imprimir o saldo com reajuste de 1.5%.
- (h) Solicite o consumo energético de uma residência, calcule e imprima o valor em reais de cada kw, o valor em reais a ser pago e o novo valor a ser pago por uma residência com um desconto de 10%. Dado: 100 kilowatts custa 1/7 do salário mínimo.
- (i) Informar um preço de um produto e calcular novo preço com desconto de 9%.
- (j) Faça o cálculo do salário líquido de um professor. Serão fornecidos o valor da hora aula, número de aulas dadas e o % de desconto do INSS.
- (k) Ler uma temperatura em graus Celsius e transformá-la em Fahrenheit.
- (l) Ler um número e se for maior que 20 imprimir a metade desse número, caso contrário, imprimir o dobro.
- (m) Ler 2 números inteiros e somá-los. Se a soma for maior que 10, mostrar o resultado da soma.
- (n) Ler um número. Se positivo, imprimir a raiz quadrada senão o quadrado.
- (o) Solicitar salário e o valor de prestação. Se prestação for maior que 20% do salário, imprimir: “Empréstimo não pode ser concedido”. Senão imprimir “Empréstimo concedido”.
- (p) 16. Ler um número 'X' e imprimir: “X é maior que 20”, “X é igual a 20” ou “X é menor que 20”.
- (q) Ler um ano de nascimento e ano atual. Imprimir a idade da pessoa.
- (r) Receber um número do teclado e informar se ele é divisível por 10, por 5, por 3, por 2 ou se não é divisível por nenhum destes.

- (s) Um comerciante comprará um produto e quer vendê-lo com lucro de 45% se o valor da compra for menor que R\$ 20,00; caso contrário, o lucro será de 30%. Entrar com o valor do produto e imprimir o valor da venda.
 - (t) Ler a idade de uma pessoa e informar a sua classe eleitoral (não votante, facultativo ou obrigatório).
 - (u) Ler 3 números e imprimir se eles podem ou não ser lados de um triângulo.
 - (v) Ler 2 valores e somar os dois. Caso a soma seja maior que 10, mostrar a soma.
 - (w) Entrar com um número e imprimir a sua raiz quadrada, caso ele seja positivo ou o quadrado caso seja negativo.
 - (x) Ler a idade de uma pessoa e verificar se está compreendido entre 18 e 70. Se tiver, imprimir “voto obrigatório.”, senão imprimir “não votante ou voto facultativo”.
 - (y) Ler o tipo de carro (A, B e C) desejado. Informe o percurso rodado em km e calcule o consumo estimado, conforme o tipo, sendo (A=6, B=9 e C=15) km/litro.
 - (z) Receba três números e imprimá-os em ordem decrescente.
62. Solicitar nome e data de nascimento no formato DD/MM/AAAA e escrever “Fulano nasceu em DD de Nome do mês do ano de AAAA”.
63. Solicite o valor de cinco produtos. Imprima-os em ordem crescente e formatada com números de tamanho 7 e duas casas decimais, bem como o valor total da compra, o valor do desconto de 5%, o total – desconto, o valor do produto mais caro, do mais barato e média dos preços.
- Exemplo:
(entradas)

100.00 200.00 50.00 1000.00 500.00

(saídas)

```

    50.00
  100.00
  200.00
  500.00
 1000.00
    Total bruto = 1850.00
  Desconto    =   92.50
  Total liq.  = 1757.50
  Mais caro   = 1000.00
  Mais barato =   50.00
  Media       =   370.00

```

64. Escreva um programa que calcule o índice de massa corporal (IMC) de uma pessoa indicando se ela está abaixo do peso, normal ou acima do peso. São necessários os dados: sexo, peso e altura. Uma pessoa do sexo feminino é considerada abaixo do peso para um índice menor que 20, normal para índice entre 20 e 24 (inclusive) e acima se superior a 24. Uma pessoa do sexo masculino é considerada abaixo do peso para um índice menor que 20, normal quando entre 20 e 23 (inclusive) e acima quando superior a 23.

O índice de massa corporal é calculado pela expressão $\text{peso}/\text{altura}^2$.

As entradas do programa devem ser: sexo (char), peso (real) e altura (real). As saídas: IMC (real, com 2 casas decimais) e classificação (texto). Utilize entradas e saídas **secas** (sem texto explicativo), conforme o exemplo abaixo:

```
M          (entrada 1 = Sexo)
60          (entrada 2 = Peso)
1.67        (entrada 3 = Altura)
21.51       (saída 1 = IMC)
normal      (saída 2 = classificação)
```

65. Qual a diferença entre o tipo `int` e o tipo `unsigned int`? Porque não usar um só?
66. Quais nomes de variáveis são aceitas pelo compilador C?

- | | |
|--------|------------------|
| • 3ab | • ____A |
| • ab3 | • n_a_o |
| • a3b | • A123 |
| • FIM | • papel-branco |
| • _sim | • a* |
| • int | • c++ |
| • \meu | • *nova_variavel |

67. Qual a diferença entre as seguintes atribuições:

```
char a;
a='6';
a=6;
```

68. Qual o valor de cada uma das expressões abaixo: (dados: `int i=1, j = 2, k = 3, n = 2; float x=3.3, y = 4.4;`)

- | | |
|--|--|
| (a) $i < j + 3$ | (f) $i \parallel j - 3 \ \&\& \ 0$ |
| (b) $2 * i - 7 \leq j - 8i + j + k == -2 * -k$ | (g) $i < j \ \&\& \ 2 \geq k$ |
| (c) $!(n - j)$ | (h) $i < j \parallel 2 \geq k$ |
| (d) $!n - j$ | (i) $x \leq 5.0 \ \&\& \ x \neq 1.0 \parallel i > j$ |
| (e) $i \ \&\& \ j \ \&\& \ k$ | |

69. Escreva expressões equivalentes sem usar o operador de negação (!)

- (a) $!(i == j)$

- (b) $!(i + 1 < j - 2)$
- (c) $!(i < j \ \&\& \ n < m)$
- (d) $!(i < 1 \ || \ j < 2 \ \&\& \ n < 3)$

70. Calcular o consumo médio de gasolina de um tanque de automóvel. Pedir para o usuário entrar com a distância (d) em quilômetros e o volume (v) em litros. (litros). $C_m = d/v$.
71. Calcular área de uma esfera. $A = 4\pi r^2$
72. Descreva em poucas palavras o que acontece com os códigos abaixo:

- `if (1) {
 printf("If\n");
} else {
 printf("else");`
- `if (0) {
 printf("If\n");
} else {
 printf("else");`
- `if (2) {
 printf("If\n");
} else {
 printf("else");`

73. Escreva um programa que ordene três números. Tente encontrar uma versão com apenas 3 comandos if.
74. Criar um algoritmo que receba:

- O valor do salário mínimo
- Quantidades de horas trabalhadas
- Número de dependentes do funcionário
- Número de horas extras trabalhadas

Calcular e imprimir o salário a receber de um funcionário, tendo em vista as regras a seguir:

- O valor da hora trabalhada é igual a 1/10 do salário mínimo
- O salário do mês é igual ao número de horas trabalhadas vezes o valor da hora trabalhada
- Para cada dependente acréscimo de 32 reais
- Para cada hora extra trabalhada acréscimo de 50% ao valor da hora trabalhada

75. No Brasil o voto é facultativo para maiores de 16 anos e menores de 18 anos, analfabetos e maiores de 70 anos. Em todos os outros casos, a pessoa é obrigada a votar. Desenvolva um algoritmo que informe se a uma pessoa é ou não obrigada a votar.
76. Faça um algoritmo que informe se um número é par ou ímpar.
77. Escreva um algoritmo que funcione como uma máquina de calcular operando da seguinte maneira:
- (a) primeiro deve pedir os dois operandos;
 - (b) logo a seguir deve pedir o tipo de operação (+, -, *, /);
 - (c) após a escolha do operador, deve apresentar o resultado indicando que tipo de operação foi executada. Ex.: $a + b = 6$;
 - (d) se o operador não for um dos especificados, uma mensagem deve informar o fato.
78. A prefeitura do Rio de Janeiro abriu uma linha de crédito para os funcionários estatutários. O valor máximo da prestação não poderá ultrapassar 30% do salário bruto. Fazer um algoritmo que informe se um empréstimo pode ou não ser concedido.
79. Desenvolva um algoritmo que leia um número inteiro de 3 dígitos e imprima se o algarismo da casa das centenas é par ou ímpar.
80. Faça um algoritmo que informe o estado civil de uma pessoa. Os possíveis estados civis são: solteiro (S), casado (C), divorciado (D) e viúvo (V).
81. Elaborar algoritmo que lê do teclado uma letra que pode ser 'F' ou 'J' e mostra a mensagem "pessoa física", "pessoa jurídica" ou "tipo de pessoa inválido", conforme o caso.
82. Construa um algoritmo que lê um número de 1 a 7 e informa o dia da semana correspondente, sendo domingo o dia de número 1. Se o número não corresponder a um dia da semana, é mostrada uma mensagem de erro.
83. Faça um algoritmo que leia um número inteiro de 1 a 12 e informe o mês correspondente. Caso o usuário digite um número fora desse intervalo, deverá aparecer uma mensagem informando que não existe mês com este número.
84. Escrever um algoritmo que lê um caractere e informa se é letra, dígito, operador aritmético ou nenhum deles.

85. Faça um algoritmo de menu que mostra na tela, sob o título de "Cálculo de Área", três opções: "R – retângulo", "C – círculo" e "T – triângulo", lê do teclado a opção desejada pelo usuário e calcule a área da figura ($R = base * altura$; $C = \pi * r^2$; $T = base * altura / 2$) de acordo com a opção escolhida ou uma mensagem de erro, se a opção for inválida.
86. Faça um algoritmo que converta notas em conceitos. Caso o aluno tenha obtido uma nota entre 0 e 49, o seu conceito deverá ser D. Caso o aluno tenha obtido uma nota entre 50 e 69, o seu conceito deverá ser C. Caso o aluno tenha obtido uma nota entre 70 e 89, o seu conceito deverá ser B. Caso o aluno tenha obtido uma nota entre 90 e 100, o seu conceito deverá ser A.
87. Faça um algoritmo que leia uma idade e mostre uma mensagem de acordo com a seguinte definição:

$0 \leq idade \leq 3$ (bebê); $4 \leq idade \leq 10$ (criança); $11 \leq idade \leq 18$ (adolescente);
outros casos (adulto).

88. Desenvolva um algoritmo que informe o dobro e o quadrado de um número inteiro.
89. Faça um algoritmo informe o resultado da soma, multiplicação, divisão inteira e o resto da divisão inteira entre dois números.
90. Desenvolva um algoritmo que informe a média aritmética das 4 notas bimestrais fornecidas pelo aluno.
91. O preço de um automóvel é calculado pela soma do preço de fábrica com o preço dos impostos (45% do preço de fábrica) e a percentagem do revendedor (28% do preço de fábrica). Faça um algoritmo que informe o preço final de um automóvel a partir do preço de fábrica fornecido pelo gerente.
92. Faça um algoritmo que calcule a velocidade média. Dado que:

$$V_m = \frac{\Delta S}{\Delta T}$$

93. Faça um algoritmo que calcule a área de um triângulo:

$$Área_{\Delta} = \frac{Base * Altura}{2}$$

94. Faça um algoritmo que converta uma temperatura em graus Fahrenheit e para a escala Celsius. A fórmula de conversão é:

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

95. Elabore um algoritmo que calcule o valor da função $f(x) = 3x^2 + 5x - 1$. Considere que o valor de x é fornecido pelo usuário.
96. Elabore um algoritmo que informe o raio de um círculo dado sua área. $\text{Área}_o = \pi * r^2$.
97. Faça um algoritmo que leia três números e os imprima na ordem contrária à que foram digitados. Por exemplo: se o usuário digitar 4, 2, 9 o algoritmo deve imprimir: 9, 2, 4.
98. Fornecidos separadamente o nome e o sobrenome pelo usuário, desenvolva um algoritmo que exiba o nome e o sobrenome concatenados.
99. Escrever um algoritmo que leia um peso na Terra e o número de um planeta e imprima o valor de seu peso neste planeta. A relação de planetas é dada a seguir juntamente com o valor das gravidades relativas à Terra:

Planeta	Gravidade Relativa
Mercúrio	0,37
Vênus	0,88
Marte	0,38
Júpiter	2,64
Saturno	1,15
Urano	1,17

Para calcular o peso no planeta use a fórmula: $P_{\text{planeta}} = P_{\text{terra}}/10 * \text{gravidade}$

100. Um carro “total flex “ pode ser abastecido com álcool ou gasolina. Dado o valor do álcool e da gasolina e sabendo que o álcool tem rendimento de 70% em relação a gasolina, verifique com qual combustível compensa abastecer o carro.
101. Ler 4 números e mostrar os números divisíveis por 2 e 3.
102. Ler 4 números e mostrar os números divisíveis por 2 ou 3.
103. No “par ou ímpar” tradicional cada jogador i escolhe um número n_i e um dos possíveis restos da divisão inteira da soma $(n_1 + n_2)$ por 2: 0 (par) ou 1 (ímpar). No “par ou ímpar” de três jogadores cada jogador i escolhe um número n_i e um dos possíveis restos da divisão inteira de $(n_1 + n_2 + n_3)$ por 3: 0, 1 ou 2. O vencedor é aquele que escolher o resto que foi efetivamente calculado.
Escreva um programa que solicita os três números inteiros n_1 , n_2 e n_3 que correspondem aos números escolhidos pelos jogadores Alice, Bob e Carol, respectivamente, e escreva na tela o nome do jogador vencedor. Considere sempre que Alice escolheu o resto 0, Bob escolheu o resto 1 e Carol escolheu o resto 2.

104. Escreva um programa que mostre na tela um *menu* de pratos (pelo menos 5), cada um associado a um número.

Prato 1 - Miojo
Prato 2 - Ensopado

Quando um número é selecionado, o programa deve exibir uma breve descrição do prato. Por exemplo, ao digitar 1, o programa mostra: “Macarrão instantâneo”

105. Faça um programa de eleição com as opções:

- candidato A
- candidato B
- nulo
- branco

Permita ao usuário votar e mostre a opção escolhida por ele.

106. Dadas as tabelas de transformação de algarismos arábicos para romanos. Ler um número inteiro menor que 2000 e escrever este número em algarismos romanos.

unidade	I	II	III	IV	V	VI	VII	VIII	IX
dezena	X	XX	XXX	XL	L	LX	LXX	LXXX	XC
centena	C	CC	CCC	CD	D	DC	DCC	DCCC	CM
milhar	M	MM	MMM						

107. Verifique o resultado de r para cada uma das expressões abaixo. Considere individualmente e não como uma sequência de atribuições. Considere r como inteiro e como real. Peça ao usuário entrar com os valores de a, b, c

- | | | |
|-------------------------------|-------------------------------------|---|
| (a) $r \leftarrow 5 + a;$ | (e) $r \leftarrow b * d + c * a;$ | (i) $r \leftarrow c;$ |
| (b) $r \leftarrow a/c;$ | (f) $r \leftarrow b * (d + c * a);$ | (j) $r \leftarrow a + b * 0,5/3,0 + d;$ |
| (c) $r \leftarrow (b * a)/c;$ | (g) $r \leftarrow a \bmod c;$ | |
| (d) $r \leftarrow b * (a/c);$ | (h) $r \leftarrow -b;$ | |

108. Faça um programa que receba quatro notas de um aluno, calcule e imprima a média aritmética das notas e a mensagem de aprovado para média superior a 7,0 ou a mensagem de reprovado para média inferior a 7,0.
109. Faça o pseudocódigo de um programa que receba quatro números inteiros, calcule e mostre a soma desses números.
110. Faça o pseudocódigo de um programa que receba três notas, calcule e mostre a média aritmética entre elas.
111. Faça o pseudocódigo de um programa que receba três notas e seus respectivos pesos, calcule e mostre a média ponderada dessas notas.
112. Faça o pseudocódigo de um programa que receba o salário de um funcionário, calcule e mostre o novo salário, sabendo-se que este sofreu um aumento de 25%.
113. Faça o pseudocódigo de um programa que receba o salário de um funcionário e o percentual de aumento, calcule e mostre o valor do aumento e o novo salário.

114. Faça o pseudocódigo de um programa que receba o salário de um funcionário, calcule e mostre o salário a receber, sabendo-se que o funcionário tem gratificação de 5% sobre o salário base e paga imposto de 7% sobre este salário.
115. Faça o pseudocódigo de um programa que receba o salário base de um funcionário, calcule e mostre o seu salário a receber, sabendo-se que o funcionário tem gratificação de R\$ 50,00 e paga imposto de 10% sobre o salário base.
116. Faça o pseudocódigo de um programa que receba o valor de um depósito e o valor da taxa de juros, calcule e mostre o valor do rendimento e o valor total depois do rendimento.
117. Faça o pseudocódigo de um programa que calcule e mostre a área de um triângulo.
118. Faça o pseudocódigo de um programa que calcule e mostre a área de um círculo.
119. Faça o pseudocódigo de um programa que receba um número maior que zero, calcule e mostre:
- (a) o número digitado ao quadrado;
 - (b) o número digitado ao cubo;
 - (c) a raiz quadrada do número digitado;
 - (d) a raiz cúbica do número digitado.
120. Faça o pseudocódigo de um programa que receba dois números maiores que zero, calcule e mostre um elevado ao outro.
121. Sabe-se que:
- (a) 1 pé = 12 polegadas
 - (b) 1 jarda = 3 pés
 - (c) 1 milha = 1.760 jardas
- Faça o pseudocódigo de um programa que receba uma medida em pés, faça as conversões para polegadas, jardas e milhas e mostre os resultados.
122. Faça o pseudocódigo de um programa que receba o ano de nascimento de uma pessoa e o ano atual, calcule e mostre a idade dessa pessoa e quantos anos ela terá em 2050.
123. O custo ao consumidor de um carro novo é a soma do preço de fábrica com o percentual de lucro do distribuidor e dos impostos aplicados ao preço de fábrica. Faça o pseudocódigo de um programa que receba o preço de fábrica de um veículo, o percentual de lucro do distribuidor e percentual de impostos, calcule e mostre: 1) o valor correspondente ao lucro do distribuidor; 2) o valor correspondente aos impostos; e 3) o preço final do veículo.
124. Faça o pseudocódigo de um programa que receba o número de horas trabalhadas e o valor do salário mínimo, calcule e mostre o salário a receber seguindo as regras:
- (a) a hora trabalhada vale a metade do salário mínimo.
 - (b) o salário bruto equivale ao número de horas trabalhadas multiplicado pelo valor da hora trabalhada.
 - (c) o imposto equivale a 3% do salário bruto.
 - (d) o salário a receber equivale ao salário bruto menos o imposto.

125. Um trabalhador recebeu seu salário e o depositou em uma conta bancária. Esse trabalhador emitiu dois cheques e agora deseja saber seu saldo atual. Sabe-se que cada operação bancária de retirada paga CPMF de 0,38% e o saldo inicial da conta está zerado.
126. Pedro comprou um saco de ração com peso em quilos. Ele possui dois gatos, para os quais fornece a quantidade de ração em gramas. A quantidade diária de ração fornecida para cada gato é sempre a mesma. Faça o pseudocódigo de um programa que receba o peso do saco de ração e a quantidade de ração fornecida para cada gato, calcule e mostre quanto restará de ração no saco após cinco dias.
127. Cada degrau de uma escada tem X de altura. Faça o pseudocódigo de um programa que receba essa altura e a altura que o usuário deseja alcançar subindo a escada, calcule e mostre quantos degraus ele deverá subir para atingir seu objetivo, sem se preocupar com a altura do usuário. Todas as medidas devem estar em metros.
128. Sabe-se que o quillowatt de energia custa um quinto do salário mínimo. Faça o pseudocódigo de um programa que receba o valor do salário mínimo e quantidade de quillowatt consumida por um residência, calcule e mostre: 1) o valor de cada quillowatt; 2) o valor a ser pago pela residência; e 3) o valor a ser pago com desconto de 15%.
129. Determinar a idade de uma pessoa (em anos, meses e dias), dadas a data de seu nascimento e a data atual.
130. Ler a altura e o sexo de uma pessoa e calcular seu peso ideal, utilizando as seguintes fórmulas:
- para homens: $(72.7 \times \text{altura}) - 58$
 - para mulheres: $(62.1 \times \text{altura}) - 44.7$
131. Escreva um programa que leia a altura e o raio de um cilindro circular e imprima o volume do cilindro. O volume de um cilindro circular é calculado por meio da seguinte fórmula: $\text{Vol} = 3.141592 * \text{raio} * \text{raio} * \text{altura}$.
132. Em um cercado, há vários patos e coelhos. Escreva um programa que solicite ao usuário o total de cabeças e o total de pés, e determine quantos patos e quantos coelhos se encontram nesse cercado.
133. Dois amigos jogam na loteria toda semana. Escreva um programa que solicite a quantia com que cada jogador participou e o valor do prêmio a ser rateado em partes diretamente proporcionais às quantias de cada um deles. O programa deve imprimir quanto cada um dos amigos receberá caso sejam ganhadores.
134. Um empresa contrata um encanador a R\$ 20,00 por dia. Crie um programa que solicite o número de dias trabalhados pelo encanador e imprima o valor líquido a ser pago, sabendo que são descontados 8% de imposto de renda.
135. Faça um programa que solicite um caractere do teclado. Se for uma letra minúscula, imprima-a em maiúsculo. Caso contrário, imprima o próprio caractere. Use uma expressão condicional.

136. Uma empresa decide dar um aumento de 25% aos funcionários cujo salário é inferior a R\$630,00. Escreva um programa que receba o salário de um funcionário e imprima o valor do salário reajustado ou uma mensagem caso não tenha direito ao aumento.
137. Faça um programa que calcule e imprima o salário reajustado de um funcionário de acordo com a seguinte regra:
- Salários até R\$1.800,00, reajuste de 25,5%.
 - Salários maiores que R\$1.800,00, reajuste de 12,25%.
138. No curso de química, a nota final do estudante é calculada a partir de 3 notas atribuídas respectivamente a um trabalho de laboratório, a uma avaliação semestral e a um exame final. As notas variam de 0 a 10 e, a nota final é a média ponderada das 3 notas mencionadas. A tabela a seguir fornece os pesos das notas:

Laboratório	Peso 3
Avaliação semestral	Peso 2
Exame Final	Peso 5

Faça um programa que receba as 3 notas do estudante, calcule e imprima a média final e o conceito desse estudante. O conceito segue a tabela abaixo:

Média Final	Conceito
8,5 a 10,0	A
7,0 a 8,5	B
6,0 a 7,0	C
5,0 a 6,0	D
Menor que 5,0	E

139. Escreva um programa que, tendo como dados de entrada o preço de um produto e um código de origem, emita o preço junto de sua procedência. Caso o código não seja nenhum dos especificados, o produto deve ser encarado como importado. A procedência obedece a seguinte tabela:

Código de origem	Procedência
1	Sul
2	Norte
3	Leste
4	Oeste
5 ou 6	Nordeste
7, 8 ou 9	Sudeste
10 até 20	Centro-oeste
21 até 30	Nordeste

140. Escreva um programa que receba quatro notas de um aluno, calcule e imprima a média aritmética entre essas quatro notas e uma mensagem que segue a tabela abaixo:

Média	Mensagem
0,0 a 4,9	Reprovado
5,0 a 5,9	SAC
6,0 a 10,0	Aprovado

141. Faça um programa que receba a idade de uma pessoa e classifique-a seguindo o critério a seguir:

Idade	Classificação
0 a 2 anos	Recém-nascido
3 a 11 anos	Criança
12 a 19 anos	Adolescente
20 a 55 anos	Adulto
Acima de 55 anos	Idoso

142. Escreva um programa que receba o código correspondente ao cargo de um funcionário e imprima seu cargo e o percentual de aumento ao qual este funcionário tem direito seguindo a tabela abaixo:

Código	Cargo	Percentual
1	Escriturário	60%
2	Secretário	40%
3	Caixa	25%
4	Gerente	5%
5	Diretor	Não tem aumento

143. Faça um programa que mostre um menu com as seguintes opções:

- (a) soma
- (b) subtração
- (c) multiplicação
- (d) divisão
- (e) potência
- (f) finalizar

O programa deve receber a opção desejada, receber dois valores para a operação de cada opção, realizar a operação e imprimir o resultado. Na opção finalizar nada deve acontecer.

144. Uma companhia de seguros tem três categorias de seguros baseadas na idade e ocupação do segurado. Somente pessoas com pelo menos 18 anos e não mais de 70 anos podem adquirir apólices de seguros. Quanto às classes de ocupações foram definidos três grupos de risco. A tabela a seguir fornece as categorias em função da faixa de idade e do grupo de risco:

Idade	Grupo de Risco		
	Baixo	Médio	Alto
18 a 24	7	8	9
25 a 40	4	5	6
41 a 70	1	2	3

Faça um programa que receba a idade e o grupo de risco, e determine e imprima o código do seguro.

145. Faça um programa que receba a medida de um ângulo em graus (um número inteiro). Determine e imprima o quadrante em que se localiza este ângulo. Considere os quadrantes abaixo:

Angulo	Quadrante
0 a 90	1º
90 a 180	2º
180 a 270	3º
270 a 360	4º
0 a -90	1º
-90 a -180	2º
-180 a -270	3º
-270 a -360	4º

146. Uma empresa decidiu dar uma gratificação de Natal aos seus funcionários, baseada no número de horas extras e no número de horas que o funcionário faltou ao trabalho. O valor do prêmio é obtido pela consulta na tabela a seguir, em que:

$$H = (\text{Número de horas extras}) - (2/3 * (\text{Número de horas-falta}))$$

H (minutos)	Prêmio (\$)
Maior que 2400	500
Entre 1800 e 2400	400
Entre 1200 e 1800	300
Entre 600 e 1200	200
Menor ou igual a 600	100

Faça um programa que receba o número de horas extras e o número de horas-falta em minutos de um funcionário. Imprima o número de horas extras em horas, o número de horas-falta em horas e o valor do prêmio.

147. Faça um programa que receba o valor do salário mínimo, o número de horas trabalhadas, o número de dependentes do funcionário e a quantidade de horas extras trabalhadas. Calcule e imprima o salário a receber seguindo as regras abaixo:

- o valor da hora trabalhada é igual a $1/5$ do salário mínimo;
- o salário do mês é igual a número de horas trabalhadas vezes o valor da hora trabalhada;
- para cada dependente acréscimo de R\$32,00;
- para cada hora extra trabalhada o cálculo do valor da hora trabalhada acrescida de 50
- o salário bruto é igual ao salário do mês acrescido dos valores dos dependentes e dos valores das horas extras;
- o cálculo do valor do imposto de renda retido na fonte segue a tabela abaixo:

IRPF	Salário Bruto
Isento	Inferior a 700
10%	De 700 a 1100
20%	Superior a 1100

- o salário líquido é igual ao salário bruto menos IRPF;
- a gratificação segue a próxima tabela:

Salário Líquido	Gratificação
Até 950	R\$100,00
Superior a 950	R\$50,00

- o salário a receber do funcionário é igual ao salário líquido mais a gratificação.

148. A disciplina de 'Algoritmos e Programação' adota o seguinte método de avaliação:

Serão aplicadas duas provas escritas: P1 e P2, sem consulta. Não haverá prova substitutiva

A média das provas será calculada por $M_{prova} = (P1 + 2P2) / 3$.

Serão cobradas duas listas de exercícios de implementação L1 e L2 com vários programas para serem desenvolvidos.

A média das listas será calculada por $M_{trab} = (L1 + L2) / 2$.

Dessa forma, a média será calculada por $Média = (M_{prova} + M_{trab}) / 2$.

Os critérios de aprovação são:

- Se $Frequencia < 75\%$ ou $Média < 5$ então o aluno é Reprovado. Sendo, a média final dada por $M_{final} = Média$.
- Senão, se $5 \leq Média < 6$ então o aluno terá o direito de fazer o SAC.
- Caso $Média \geq 6$, o aluno é Aprovado. Sendo $M_{final} = Média$.

Sabendo que o número total de aulas é igual a 16, faça um programa que receba o número de faltas do aluno e as notas P1, P2, L1 e L2. Calcule e imprima M_{prova} , M_{trab} e $Média$ e informe a situação final do aluno.

- Escreva um programa que lê número de matrícula e média de no máximo 100 alunos. O programa deve ler informações de entrada até que o usuário digite um número negativo para o número de matrícula. Após inserir todas as notas, o programa deve perguntar se o usuário deseja realizar uma consulta a média de um aluno. Caso a resposta seja positiva, o programa deve buscar pela média de um aluno, dado o seu número de matrícula.
- Escreva um programa que leia um vetor de tamanho n da entrada padrão (n informado pelo usuário) e escreva na saída padrão o índice do maior elemento e o índice do menor elemento desse vetor.
- Escreva um programa que lê uma lista de inteiros positivos e imprime essa lista em ordem inversa. A lista pode ter até 100 elementos e a leitura deve ser interrompida assim que o primeiro inteiro não positivo for digitado. Não será informado previamente o número de inteiros na lista de inteiros.

152. Faça um programa que imprima a tabuada do 9.
153. Faça um programa que imprima a soma dos valores da tabuada do 9.
154. Faça um programa que imprima a tabuada do 1 ao 9.
155. Faça um programa que imprima a soma dos valores da tabuada do 1 ao 9.
156. Faça um programa que receba um número e informe se ele é par, ímpar e primo ou não-primo.
157. Faça um programa que receba idades de pessoas até o usuário digitar 0. Verifique se a idade digitada é um número maior ou igual a zero e solicite novamente caso não seja. Depois imprima a idade do mais velho, do mais novo e a média das idades.
158. Faça um programa que receba dez notas finais de uma turma, calcule e imprima a média aritmética, a maior e a menor nota, quantos alunos obtiveram nota maior que 6.0, quantos alunos ficaram de SAC ($6.0 < \text{nota} \leq 5.0$) e quantos alunos reprovaram.
159. (SPOJ COMETA2 - 8777. Cometa) O cometa Halley é um dos cometas de menor período do Sistema Solar, completando uma volta em torno do Sol a cada 76 anos; na última ocasião em que ele tornou-se visível do planeta Terra, em 1986, várias agências espaciais enviaram sondas para coletar amostras de sua cauda e assim confirmar teorias sobre suas composições químicas.

Tarefa

Escreva um programa que, dado o ano atual, determina qual o próximo ano em que o cometa Halley será visível novamente do planeta Terra. Se o ano atual é um ano de passagem do cometa, considere que o cometa já passou nesse ano (ou seja, considere sempre o próximo ano de passagem, não considerando o ano atual).

Entrada

A única linha da entrada contém um único inteiro A ($2010 \leq A \leq 104$), indicando o ano atual.

Saída

Seu programa deve imprimir uma única linha, contendo um número inteiro, indicando o próximo ano em que o cometa Halley será visível novamente do planeta Terra.

Exemplo

Entrada
2010

Saída
2062

Entrada
10000

Saída
10042

Entrada
2062

Saída
2138

160. (SPOJ JPNEU - 8697. Pneu) Calibrar os pneus do carro deve ser uma tarefa cotidiana de todos os motoristas. Para isto, os postos de gasolina possuem uma bomba de ar. A maioria das bombas atuais são eletrônicas, permitindo que o motorista indique a pressão desejada num teclado. Ao ser ligada ao pneu, a bomba primeiro lê a pressão atual e calcula a diferença de pressão entre a desejada e a lida. Com esta diferença ela esvazia ou enche o pneu para chegar na pressão correta.

Sua ajuda foi requisitada para desenvolver o programa da próxima bomba da SBC - Sistemas de Bombas Computadorizadas.

Tarefa

Escreva um programa que, dada a pressão desejada digitada pelo motorista e a pressão do pneu lida pela bomba, indica a diferença entre a pressão desejada e a pressão lida.

Entrada

A primeira linha da entrada contém um inteiro N que indica a pressão desejada pelo motorista ($1 \leq N \leq 40$). A segunda linha contém um inteiro M que indica a pressão lida pela bomba ($1 \leq M \leq 40$).

Saída

Seu programa deve imprimir uma única linha, contendo a diferença entre a pressão desejada e a pressão lida.

Exemplo

Entrada
30

18

Saída

12

Entrada

27

27

Saída

0

Entrada

27

30

Saída

-3

161. (SPOJ QUADRAD2 - 3829. Quadrados) Dado um inteiro N , determine quanto vale N^2 .

Entrada

A entrada é composta por um único caso de teste, composto por uma única linha que contém o inteiro N .

Saída

Seu programa deve produzir uma única linha, contendo o valor de N^2 .

Restrições

- $|N| \leq 10000$

Exemplo

Entrada

1

Saída

1

Entrada

4

Saída

16

Entrada

2

Saída

4

162. (SPOJ CONTA1 - Conta de Água) A empresa local de abastecimento de água, a Saneamento Básico da Cidade (SBC), está promovendo uma campanha de conservação de água, distribuindo cartilhas e promovendo ações demonstrando a importância da água para a vida e para o meio ambiente.

Para incentivar mais ainda a economia de água, a SBC alterou os preços de seu fornecimento de forma que, proporcionalmente, aqueles clientes que consumirem menos água paguem menos pelo metro cúbico. Todo cliente paga mensalmente uma assinatura de R\$ 7, que inclui uma franquia de 10 m^3 de água. Isto é, para qualquer consumo entre 0 e 10 m^3 , o consumidor paga a mesma quantia de R\$ 7 reais (note que o valor da assinatura deve ser pago mesmo que o consumidor não tenha consumido água). Acima de 10 m^3 cada metro cúbico subsequente tem um valor diferente, dependendo da faixa de consumo. A SBC cobra apenas por quantidades inteiras de metros cúbicos consumidos. A tabela abaixo especifica o preço por metro cúbico para cada faixa de consumo:

Faixa de consumo (m^3)	Preço (m^3)
até 10	incluído na franquia
11 a 30	R\$ 1
31 a 100	R\$ 2
101 em diante	R\$ 5

Assim, por exemplo, se o consumo foi de 120 m^3 , o valor da conta é:

- 7 reais da assinatura básica;
- 20 reais pelo consumo no intervalo 11 - 30 m^3 ;
- 140 reais pelo consumo no intervalo 31 - 100 m^3 ;
- 100 reais pelo consumo no intervalo 101 - 120 m^3 .

Logo o valor total da conta de água é R\$ 267.

Tarefa

Escreva um programa que, dado o consumo de uma residência em m^3 , calcula o valor da conta de água daquela residência.

Entrada

A única linha da entrada contém um único inteiro N , indicando o consumo de água da residência, em m^3 ($0 \leq N \leq 10^3$).

Saída

Seu programa deve imprimir uma única linha, contendo o valor da conta de água daquela residência.

Exemplo

Entrada

8

Saída

7

Entrada

14

Saída

11

Entrada

42

Saída

51

163. (SPOJ JSEDEX - 8696. Sedex) A Copa do Mundo de 2010 será realizada na África do Sul. Bolas de futebol são muito fáceis de transportar, já que elas saem das fábricas vazias e só são enchidas somente pelas lojas ou pelos consumidores finais.

Infelizmente o mesmo não pode ser dito das bolas de boliche. Como elas são completamente sólidas, elas só podem ser transportadas embaladas uma a uma, em caixas separadas.

A SBC - Só Boliche Cascavel - é uma fábrica de bolas de boliche que trabalha somente através de encomendas e envia todas as bolas por SEDEX. Como as bolas têm tamanhos diferentes, a SBC tem vários tamanhos de caixas diferentes para transportá-las

Tarefa

Escreva um programa que, dado o diâmetro de uma bola e as 3 dimensões de uma caixa (altura, largura e profundidade), diz se a bola de boliche cabe dentro da caixa ou não.

Entrada

A primeira linha da entrada contém um inteiro N ($1 \leq N \leq 10.000$) que indica o diâmetro da bola de boliche. A segunda linha da entrada contém 3 números inteiros separados por um espaço cada: a altura A ($1 \leq A \leq 10.000$), seguida da largura L ($1 \leq L \leq 10.000$) e da profundidade P ($1 \leq P \leq 10.000$).

Saída

Seu programa deve imprimir uma única linha, contendo a letra 'S' caso a bola de boliche caiba dentro da caixa ou 'N' caso contrário

Exemplo

Entrada

```
3
3 2 5
```

Saída

```
N
```

Entrada

```
5
5 5 5
```

Saída

```
S
```

Entrada

```
9
15 9 10
```

Saída

```
S
```

164. (SPOJ MARCIAN1 - 8710. Marciano) Estamos no ano 2048 e um dos sonhos da humanidade torna-se finalmente realidade: a colonização do planeta Marte. Nossos primeiros colonizadores acabam de chegar, e já começam a fazer as preparações (como a instalação de cúpulas de oxigênio e tratamento do solo para agricultura) para que mais pessoas possam tentar uma nova vida no planeta vizinho.

Apesar dos avanços tecnológicos e desafios vencidos, ainda resta um grande problema: os foguetes usados para ir a Marte ainda são complicados e caros. Com isso, fica difícil enviar suprimentos para os nossos colonos (enquanto a agricultura ainda não é possível) por muito tempo. Assim, a agência espacial contratou o SBC (Serviço Balístico Cósmico), que desenvolveu um canhão super-potente que consegue disparar esferas até Marte, sem precisar gastar milhões de dólares em equipamento e combustível.

Agora, tudo o que é necessário fazer para enviar suprimentos a Marte é colocar uma caixa com as encomendas dentro de uma esfera e disparar a mesma até seu destino.

Tarefa

Dadas as dimensões de uma caixa com suprimentos e o raio interno da esfera que é disparada pelo canhão, seu programa deverá dizer se é possível enviar tal caixa para Marte usando tal esfera.

Entrada

Cada entrada contém apenas uma linha com quatro inteiros L , A , P e R , ($0 \leq L, A, P, R \leq 1000$) que representam, respectivamente, a largura, altura e profundidade da caixa, e o raio da esfera.

Saída

Seu programa deve imprimir um único caractere: 'S' (sem aspas) se é possível colocar a caixa dentro da esfera, ou 'N' (sem aspas) caso contrário.

Exemplo

Entrada

10 20 30 30

Saída

S

Entrada

10 10 10 7

Saída

N

Entrada

2 4 4 3

Saída

S

165. (SPOJ PEDAGIO1 - 8705. Pedágio) A invenção do carro tornou muito mais rápido e mais barato realizar viagens de longa distância. Realizar uma viagem rodoviária tem dois tipos de custos: cada quilômetro percorrido na rodovia tem um custo associado (não só devido ao consumo de combustível mas também devido ao desgaste das peças do carro, pneus, etc.), mas também é necessário passar por vários pedágios localizados ao longo da rodovia.

Os pedágios são igualmente espaçados ao longo da rodovia; o começo da estrada não possui um pedágio, mas o seu final pode estar logo após um pedágio (por exemplo, se a distância

entre dois pedágios consecutivos for de 37 km e a estrada tiver 111 km, o motorista deve pagar um pedágio aos 37 km, aos 74 km e aos 111 km, logo antes de terminar a sua viagem)

Tarefa

Dadas as características da rodovia e os custos com gasolina e com pedágios, calcule o custo total da viagem.

Entrada

A entrada consiste de duas linhas. A primeira linha da entrada contém dois inteiros L e D ($1 \leq L, D \leq 10^4$), indicando o comprimento da estrada e a distância entre pedágios, respectivamente. A segunda linha contém dois inteiros K e P ($1 \leq K, P \leq 10^4$), indicando o custo por quilômetro percorrido e o valor de cada pedágio. O primeiro pedágio está localizado no quilômetro D da estrada (ou seja, a distância do início da estrada para o primeiro pedágio é D quilômetros).

Saída

Seu programa deve imprimir uma única linha contendo um único inteiro, indicando o custo total da viagem.

Exemplo

Entrada

111 37

1 10

Saída

141

Entrada

100 30

3 14

Saída

342

Entrada

20 70

9 17

Saída

180

2 Funções

1. Escreva um programa que calcule a resolução dos seguintes tópicos através do menu a seguir:
 - (a) Área de um paralelepípedo (solicitar três entradas);
 - (b) Diagonal de um paralelepípedo ($D = \sqrt{A^2 + B^2 + C^2}$) (solicitar três entradas);
 - (c) Área de um triângulo (solicitar duas entradas);
 - (d) Área de um quadrado (solicitar uma entrada);
 - (e) Diagonal do quadrado (solicitar uma entrada);
 - (f) Área de um retângulo (solicitar duas entradas);
 - (g) Área de um círculo (solicitar uma entrada);
 - (h) Comprimento de um círculo (solicitar uma entrada);
 - (i) Sair.
2. Um número natural é triangular se ele é produto de três números naturais consecutivos. Faça uma rotina que dado N natural, verifique se N é triangular.

Exemplo: 120 é triangular, pois $4 * 5 * 6 = 120$.

3. Um número N inteiro positivo é perfeito se for igual a soma de seus divisores positivos diferentes de N. Escreva uma rotina que verifique se um dado número inteiro positivo é perfeito. Escreva um programa, com reprocessamento, que utilize esta rotina.

Exemplo: 6 é perfeito, pois $1 + 2 + 3 = 6$.

4. Para evitar erros de digitação de sequências de números de importância fundamental, como a matrícula de um aluno, o CPF do Imposto de Renda, o número da conta bancária, geralmente se adiciona ao número um dígito verificador. Por exemplo, o número de matrícula 811057 é usado como 8110573, onde 3 é o dígito verificador calculado da seguinte maneira:

- (a) cada algarismo do número é multiplicado por um peso começando por 2 e crescendo de 1 da direita para a esquerda;

Exemplo: $8 * 7, 1 * 6, 1 * 5, 0 * 4, 5 * 3, 7 * 2$

- (b) somam-se as parcelas obtidas;

Exemplo: $56 + 6 + 5 + 0 + 15 + 14 = 96$

- (c) obtém-se o resto da divisão desta soma por 11;

Exemplo: 96 dividido por 11 dá resto 8.

- (d) subtrai-se de 11 o resto obtido;

Exemplo: $11 - 8 = 3$

- (e) se o valor encontrado for 10 ou 11, o dígito verificador será 0, nos outros casos o dígito verificador é o próprio valor encontrado.

Escreva uma função que retorne o dígito verificador de números de matrícula informados pelo usuário até que ele informe o 000000 para sair.

5. Para obter o dia da semana (ds) dada uma determinada data (Dia/Mes/Ano), utiliza-se a seguinte relação:

- 1) $a = (14 - \text{Mes}) / 12$
- 2) $y = \text{Ano} - a$
- 3) $m = \text{Mes} + 12a - 2$
- 4) $q = \text{Dia} + 31m/12 + y + y/4 - y/100 + y/400$
- 5) $ds = q \bmod 7$ (resto da divisão por 7)

$ds = 0 \rightarrow$ domingo, $1 \rightarrow$ segunda-feira, $2 \rightarrow$ terça-feira, ..., $6 \rightarrow$ sábado

Exemplo (21/07/2011):

Dia = 21; Mes=7; Ano=2011

- 1) $a = (14 - 7)/12 = 0$
- 2) $y = 2011 - 0 = 2011$
- 3) $m = 7 + 12 \cdot 0 - 2 = 5$
- 4) $q = 21 + 31 \cdot 5 / 12 + 2011 + 2011 / 4 - 2011 / 100 + 2011 / 400$
- 4) $q = 21 + 12 + 2011 + 502 - 20 + 5 = 2531$
- $ds = 2537 \bmod 7 = 4$ (quinta-feira)

Faça um programa que receba datas no formato dd/mm/aaaa e imprima o dia da semana correspondente. O programa deverá ser encerrado somente quando o usuário informar Dia = 00.

6. Escreva uma função que recebe como parâmetro dois valores a e b e retorna o valor de a^b . Desconsidere a função `pow`.
7. Implemente o programa abaixo e responda:

```
#include <stdio.h>
#include <stdlib.h>

void troca(int a, int b);

main() {
    int a, b;
    printf("Forneça o valor de a: ");
    scanf("%d", &a);
    printf("Forneça o valor de b: ");
    scanf("%d", &b);
    printf("Antes da chamada a função troca a = %d e b = %d.\n", a, b);
    troca(a,b);
    printf("Apos a chamada a função troca a = %d e b = %d.\n", a, b);
}
```

```

    system("PAUSE");
}

void troca(int a, int b){
    int tmp;
    tmp = a;
    a = b;
    b = tmp;
    printf("Apos a troca, os valores de a e b sao: %d e %d.\n", a, b);
}

```

- (a) Se forem fornecidos os valores 2 para a e 3 para b, qual será a saída do programa, isto é, o que você verá na tela?
- (b) Explique o que você observa em termos do comportamento de variáveis locais.
- (c) Altere o programa para que, ao invés de a função fazer uso de variáveis da lista de parâmetros para receber os valores dos dois inteiros, ela os acesse a partir de variáveis globais. Após a alteração responda novamente à pergunta feita no item (a).

8. Escreva a saída produzida pela execução do programa abaixo:

<pre> #include <stdio.h> void a() { int x, y; x = 4; y = 2; printf ("a: %d %d\n", x, y); } void b(int y) { printf ("b: %d\n", y); } </pre>	<pre> main () { x = 10; y = 3; a(); printf ("%d %d\n", x, y); b(5); printf ("%d %d\n", x, y); } </pre>
--	--

9. Escreva a saída produzida pela execução do programa abaixo:

<pre> #include <stdio.h> int x, y; void a(int x, int y) { printf ("a: %d %d\n", x, y); } void b() { int x, y; x = 7; y = 4; printf ("b: %d %d\n", x, y); } </pre>	<pre> void c() { printf ("c: %d %d\n", x, y); } main () { x = 10; y = 3; a(5,6); printf ("%d %d\n", x, y); b(); printf ("%d %d\n", x, y); c(); printf ("%d %d\n", x, y); } </pre>
---	--

10. Implemente um programa que receba dois valores reais e exiba um menu com as seguintes opções: 1 - Calcular a soma entre eles; 2 - Calcular o produto entre eles; 3 - Calcular a diferença entre eles; 4 - Calcular a razão entre eles; 5 - Todas as operações; 6 - Encerrar o programa. Faça a validação da escolha do menu, na função main, usando um laço do-while. Determine o valor escolhido usando o comando switch-case. A execução de cada opção deve ser feita mediante o chamado a uma função. As funções têm uma lista de parâmetros consistindo de dois parâmetros do tipo float e retornam um dado do tipo float. O programa não faz uso de variáveis globais, apenas locais.
11. Escreva uma função que, dado um vetor de dimensão n , retorne o índice do maior elemento daquele vetor. O cabeçalho da função deve ser:


```
int indice_maximo (int v[], int n);
```
12. Escreva uma função que receba como parâmetro o número de segundos e mostre o número de dias, horas, minutos e segundos que corresponde a aquela quantidade de segundos.
13. Escreva procedimento para as seguintes operações com vetores:
 - (a) Inicializa – define um valor que indica vazio
 - (b) Insere – depende do local de inserção (início, meio ou fim)
 - (c) Remove – depende do local de remoção (início, meio ou fim)
 - (d) Tamanho – indica a quantidade de elementos em um vetor
 - (e) Busca – localiza a posição de um elemento no vetor
 - (f) Inverte – devolver o vetor v com os seus elementos na ordem inversa
14. Implemente uma função que calcula e imprime as raízes de uma equação do segundo grau, do tipo $ax^2 + bx + c$. Essa função deve obedecer ao protótipo:


```
void raizes (float a, float b, float c);
```
15. Implemente uma função que receba como parâmetro um vetor de números inteiros de tamanho n e retorne quantos números pares estão armazenados nesse vetor. Obedeça ao protótipo:


```
int pares (int n, int vet[]);
```
16. Implemente uma função que receba uma *string* como parâmetro e retorne como resultado o número de vogais nessa *string*. Essa função deve obedecer ao protótipo:


```
int conta_vogais (char str[]);
```
17. Escreva um programa que calcula o fatorial de um número. O cálculo do fatorial deve ser realizado por uma função.
18. Escreva um programa que calcula o valor do co-seno de x através de 20 termos da série seguinte:

$$\text{co-seno}(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$

O cálculo do valor do co-seno de x deve ser realizado por uma função. Note que o uso da função fatorial desenvolvida no exercício anterior simplificará a resolução deste exercício.

19. Considere a função:

```

Function F1 (k, l: Integer): Integer;
Begin
    If k > l Then F1 := 1
    Else If k = l Then F1 := 0
        Else F1 := -1;
End; { F1 }

```

Calcule:

(a) $F1(2, 7)$

(b) $F1(3, 3) + F1(6, 4) + F1(2, 7)$

(c) $F1(2 * F1(1, 6), 5 * F1(8, 2) + 2 * F1(1, 1))$

20. Faça um algoritmo que leia o valor de dois números inteiros, e guarde cada um deles em uma variável diferente, por exemplo v1 e v2. Depois, o seu algoritmo deve colocar o número que está em v1 na variável v2 e o número que está em v2 na variável v1, ou seja, realizar a troca de valores das variáveis. Escreva na tela os números armazenados em cada uma das variáveis antes e depois da troca. Lembre-se em cada variável só cabe UM valor! Utilize quantas outras variáveis você quiser para fazer este algoritmo.
21. Faça um algoritmo que lê três valores e calcule a média ponderada para pesos 1, 2 e 3, respectivamente (multiplique cada nota pelo seu peso, some os produtos e divida o resultado pela soma dos pesos).
22. Faça um algoritmo que informe se um aluno foi ou não aprovado, a partir da sua média final. Considere aprovado caso a média seja maior ou igual a 6.0.
23. Num determinado Estado, o Detran cobra um imposto de 3% para veículos com até 10 anos de uso e 2% para veículos mais antigos. Dados o preço do carro e a quantidade de anos de uso, desenvolva um algoritmo que informe o valor do imposto.
24. Faça um algoritmo que leia o nome de um produto, seu preço e se há desconto. Se a resposta for afirmativa, solicite o percentual, calcule o desconto e apresente o novo preço.
25. Escreva um algoritmo que apresente o maior entre dois números, caso sejam diferentes. Se eles forem iguais, o algoritmo deve informar o fato.
26. Faça um algoritmo que leia o nome (nome1 e nome2) e a idade (idade1 e idade2) de duas pessoas, informando quem é maior e quem é menor de idade, quem é a pessoa mais velha e quem é a mais nova, bem como a diferença de idade entre elas.

27. Em um triângulo, cada lado é menor do que a soma dos outros dois. Escreva um algoritmo que lê três valores e informa se estes não podem constituir um triângulo ou, caso contrário, se o triângulo formado é equilátero (três lados iguais), isósceles (dois lados iguais) ou escaleno (lados diferentes).
28. Faça um algoritmo que simule a folha de pagamento de um funcionário. Inicialmente o algoritmo deve ler o nome do funcionário, nº de filhos, nº de salários mínimos e nº de horas extras. O algoritmo deve calcular e mostrar o salário bruto, o salário família, o desconto do INSS, desconto de IRPF, valor das horas extras a receber, valor do abono e salário líquido conforme a tabela abaixo:

ITEM	CALCULO
Salário Bruto	Quantidade de salários mínimos x R\$ 200,00
Salário Família	Número de filhos x R\$ 23,45
INSS	Salário bruto >= R\$ 360,00 (7,82% do salário) Salário bruto < R\$ 360,00 (6,62% do salário)
IRPF	Salário bruto > R\$ 280,00 (9% do salário) Salário bruto <= R\$ 280,00 (7% do salário)
Horas Extras	Salário bruto >= R\$ 750,00 (R\$ 10,00 por hora extra) Salário bruto < R\$ 750,00 (R\$ 7,00 por hora extra)
Abono	12% do salário bruto
Salário Líquido	Bruto + Família + Abono + Horas Extras – INSS - IRPF

29. Faça um algoritmo que leia informações de um aluno (matricula, frequência, nota1, nota2, nota3), calcule a média final de acordo com a fórmula $((2 * nota1) + (3 * nota2) + (4 * nota3))/9$ e mostre o valor da média, o percentual de frequência e a mensagem de acordo com a tabela abaixo:

MEDIA FINAL	FREQUÊNCIA	MENSAGEM
$media \geq 7,0$	Frequência < 50%	Final
	Frequência >= 50%	Aprovado
$5,0 \leq media < 7,0$	Frequência < 50%	Reprovado
	Frequência >= 50%	Final
$media < 5,0$	Frequência = 100%	Final
	Frequência <> 100%	Reprovado

30. Considere a função:

```
Function F2 (a, b, c: Integer): Integer;
Begin
  If (a > b) And (b > c) Then F2 := a
  Else If (a < b) And (b < c) Then F2 := a
  Else F2 := (a+b+c)/2
End; { F2 }
```

Calcule:

(a) $F2(10, 8, -1)$;

(b) $F2(1, 4, 5)$;

(c) $F2(5, 2, -2)$;

(d) $F2(X(10, 5, -1), 6 * F2(1, 3, 5) - 1, 2 - F2(1, 1, 1))$.

31. Escreva uma função que retorne o cubo de seu argumento.
32. Faça uma rotina que toque o sinal sonoro do computador.
33. Escreva uma função que receba um caractere como argumento e, se for uma letra minúscula, retorne-a em maiúsculo, caso contrário retorne o próprio caractere.
34. Escreva uma função que retorne 1 se o argumento for um número ímpar e, 0 se for par.
35. Escreva uma função que retorne 1 se o argumento for um caractere de 0 a 9 e, 0 caso contrário.
36. Faça um procedimento para trocar o conteúdo de duas variáveis reais. Escreva o modo de chamada desta rotina.
37. Escreva uma função que retorne o absoluto de um número (Não utilize a função ABS do compilador). Faça um programa, com reprocessamento, que utilize esta rotina.
38. Faça um programa que receba um inteiro $n > 1000$ e imprima todos números primos menores que ele.
39. Uma empresa possui dez funcionários com as seguintes características: código, número de horas trabalhadas no mês, turno de trabalho (M – matutino, V – vespertino ou N – noturno), categoria (O – operário ou G – gerente) e valor da hora trabalhada. Sabendo-se que essa empresa deseja informatizar sua folha de pagamento, faça um programa que:
 - (a) Leia as informações dos funcionários, exceto o valor da hora trabalhada, não permitindo que sejam informados turnos nem categorias inexistentes. Trabalhe sempre com a digitação de letras maiúsculas.
 - (b) Calcule o valor da hora trabalhada, conforme a tabela a seguir. Adote o valor de R\$ 622,00 para o salário mínimo.

CATEGORIA	TURNO	VALOR HORA TRABALHADA
G	N	18% do salário mínimo
G	M ou V	15% do salário mínimo
O	N	13% do salário mínimo
O	M ou V	10% do salário mínimo

- (c) Calcule o salário inicial dos funcionários com base no valor da hora trabalhada e no número de horas trabalhadas.
- (d) Calcule o valor do auxílio alimentação recebido por funcionário de acordo com seu salário inicial, conforme a tabela a seguir.

SALARIO INICIAL	AUXILIO ALIMENTAÇÃO
Até R\$ 500,00	20% do salário inicial
Entre R\$ 500,00 e R\$ 1000,00	15% do salário inicial
Acima de R\$ 1000,00	8% do salário inicial

- (e) Mostre o código, número de horas trabalhadas, valor da hora trabalhada, salário inicial, auxílio alimentação e salário final (salário inicial + auxílio alimentação)
40. Faça um programa que leia o número de termos, determine e mostre os valores de acordo com a série a seguir.

Série = 2, 7, 3, 4, 21, 12, 8, 63, 48, 16, 189, 192, 32, 567, 768, ...

41. Qualquer número natural de quatro algarismos pode ser dividido em duas dezenas formadas pelos seus dois primeiros e dois últimos dígitos.

Exemplo: 1278: 12 e 78.

Escreva um programa que imprima todos os milhares (números de 4 algarismos) cuja raiz quadrada seja a soma das dezenas formadas pela divisão acima.

42. Indique a saída impressa pelo programa abaixo:

```

Program Exercicio10;
Var x, y: Integer;
{+-----+}
Procedure A;
Begin
    WriteLn ('Passou pelo A');
End; { A }
{+-----+}
Procedure B (x: Integer);
Begin
    WriteLn (x);
End; { B }
{+-----+}
Procedure C (y: Integer);
Begin
    WriteLn ('Valor de y: ',y);

```

```

        y := 1;
        B(y);
End; { C }
{+-----+}
Function D (r, s: Integer): Integer;
Begin
    If r > s Then D := x
    Else D := y;
End; { D }
{+-----+}
Begin { P.P. }
    x := 20;
    y := 30;
    B(3);
    B(4);
    A;
    x := D(D(7,2),3);
    WriteLn (x);
    C(x);
    C(x-3);
    C(y);
    A;
End.

```

43. Escreva uma rotina para imprimir o cabeçalho:

```

        UFSCAR - SISTEMA DE PROGRAMAÇÃO
        XX/XX/XX                PÁG. XXX

```

A data e o número da página devem ser fornecidos como parâmetros.

44. Faça uma rotina que inverta a ordem de um número inteiro dado como parâmetro de entrada. Escreva o modo de chamada desta rotina.
45. Indique a saída impressa do programa a seguir:

```

Program Exercicio16;
Var x, y: Integer;
{+-----+}
Procedure A;
Begin
    WriteLn ('Passou pelo A');
End; { A }
{+-----+}
Procedure B (x: Integer);
Begin
    WriteLn (x);
End; { B }
{+-----+}
Procedure C (y: Integer);

```

```

Begin
    WriteLn ('Valor de y: ',y);
    y := 1;
    B(y);
End; { C }
{+-----+}
Function D (r, s: Integer): Integer;
Begin
    r := r + s;
    Inc (s);
    If r > s+10 Then D := r
    Else D := s;
End; { D }
{+-----+}
Begin { P.P. }
    x := 15;
    y := 10;
    A;
    B(3);
    x := D(x,y);
    WriteLn (D(x-y,x+1),x,y);
    C(x);
    C(y+5);
End.

```

46. Escreva uma rotina que retorne uma sequência de caracteres, resultado da concatenação de duas outras sequências fornecidas como parâmetros. Escreva, ainda, um programa que, fazendo uso desta rotina, leia separadamente o prenome e o sobrenome de uma pessoa e imprima seu nome completo. Faça reprocessamento.

47. Indique a saída impressa do programa abaixo:

```

Program Exercicio20;
Var a, b, c: Integer;
{+-----+}
Procedure P1 (a: Integer);
Begin
    WriteLn (a);
End; { P1 }
{+-----+}
Procedure P2 (b: Integer);
Begin
    b := 2*b + 1;
    P1(b);
End; { P2 }
{+-----+}
Procedure P3 (Var b: Integer);
Begin
    b := 2*b + 1;

```

```

    P1(b);
End; { P3 }
{+-----+}
Function P4 (a, b, c: Integer): Integer;
Begin
    If (a > b) And (b > c) Then
        Begin
            P3(a);
            P4 := a+2;
        End
    Else
        If (a < b) Or (c < b) Then
            Begin
                P2(b);
                P4 := 2*b+c
            End
        Else
            P4 := 0;
        End
    End; { P4 }
{+-----+}
Begin { P.P.}
    a := 3;
    b := 4;
    c := -1;
    WriteLn (P4(a,b,c));
    P1(a);
    P2(a);
    P3(a);
    WriteLn (P4(a,b-c,c));
    P2(b);
    P3(b);
    WriteLn (P4(a+2,b,c+b-1));
    P1(a);
    P1(b);
    P1(c);
End.

```

48. Dados dois números inteiros positivos, escreva uma rotina que retorne o máximo divisor comum entre eles usando o algoritmo de Euclides. Faça um programa que utilize esta rotina.

Exemplo:

$$\begin{array}{c|c|c|c|c}
 & 1 & 1 & 1 & 2 \\
 \hline
 25 & 15 & 9 & 6 & 3 \\
 \hline
 9 & 6 & 3 & 0 &
 \end{array} = \text{mdc}(24,25)$$

49. Números palíndromos são aqueles que escritos da direita para a esquerda tem o mesmo valor. Exemplo: 545, 97379, etc. Escreva uma função que, recebendo como parâmetro um

número inteiro, retorne este número escrito ao contrário. A seguir, escreva um programa que determine e imprima, usando a função acima, todos os números palíndromos entre 1 e 9999.

50. Escreva uma função que receba dois números inteiros, positivos, e determine o produto dos mesmos, utilizando o seguinte método de multiplicação:

- (a) dividir sucessivamente o primeiro número por 2 até que obtenha 1 como quociente;
- (b) paralelamente, dobrar, sucessivamente, o segundo número;
- (c) somar os números da segunda coluna que tenham como correspondente na primeira coluna um número ímpar. O total obtido é o produto procurado.

Exemplo: cálculo de $9 * 6$

96 \rightarrow 6

412

224

148 $\rightarrow \frac{+48}{54}$

A seguir escreva um programa que leia 10 pares de números e calcule os respectivos produtos, usando a função acima.

51. Um número primo é qualquer inteiro positivo que é divisível apenas por si próprio e por 1. Escreva uma função que receba um inteiro positivo e, retorne se este número é primo.
52. Dados N números inteiros positivos, calcule a soma dos que são primos. Utilize a função do exercício anterior.
53. Dado um número inteiro positivo, escreva uma rotina que determine a sua decomposição em fatores primos. Faça um programa que utilize esta rotina.
54. Escreva uma rotina que retorne o número de palavras em uma frase. Faça um programa, com reprocessamento, que utilize esta rotina.
55. Escreva uma rotina que retorne o número de palavras iniciadas por consoantes em uma frase. Faça um programa, com reprocessamento, que utilize esta rotina.
56. Escreva uma rotina que retorne o número de palavras terminadas por vogais em uma frase. Faça um programa, com reprocessamento, que utilize esta rotina.
57. Escreva uma rotina que retorne o número de palavras repetidas em uma frase. Faça um programa, com reprocessamento, que utilize esta rotina.

58. Escreva uma rotina que retorne a maior palavra de uma frase. Faça um programa, com reprocessamento, que utilize esta rotina.
59. Escreva uma rotina que retorne a menor palavra de uma frase. Faça um programa, com reprocessamento, que utilize esta rotina.
60. Um programador está implementando um processador de textos e quer adicionar uma facilidade para a composição de cartas, colocando um comando que com apenas os dados da data (dia, mês e ano), apresenta o seguinte cabeçalho:

< Dia da semana>, <dia> de <mês> de <ano>

Exemplo: Colocando a data 01/01/1901, tem-se o seguinte cabeçalho:

Terça-feira, 1 de janeiro de 1901

Para implementar esta facilidade, ajude este programador construindo uma função que tem como parâmetros o dia, o mês e o ano e retorna uma string contendo o cabeçalho como no exemplo acima. Considere que as datas estão no intervalo de 01/01/1901 a 01/01/2099.

61. Escreva uma rotina que some dois vetores de mesma dimensão. Faça um programa, com reprocessamento, que utilize essa rotina.
62. A composição dos custos das diversas atividades de construção de um prédio é feita a partir da elaboração de um quadro de quantitativos dos diversos recursos envolvidos em cada atividade. Estes recursos são de vários tipos e envolvem principalmente os custos mais diretos, como por exemplo, matérias-primas, mão-de-obra, hora de equipamento, entre outros.
- Sendo conhecidos os custos unitários para cada recurso envolvido, chega-se facilmente ao custo final unitário de cada atividade. A este custo são acrescidos os percentuais de custos indiretos (administrativos), impostos, depreciação de equipamentos, leis sociais e outros, totalizando o preço final para a execução de cada fase. Este procedimento básico é adotado em várias empreiteiras de obras, e o objetivo deste exercício é fazer um programa que execute estes cálculos para auxiliar o analista de custos de uma empreiteira.

Supondo-se que na execução do prédio são realizados quatro tipos de atividades e que cada uma consome os recursos especificados no quadro dado a seguir, e que as despesas indiretas (administração) são dados levantados a cada mês, escreva um programa que:

- (a) leia o percentual de administração do mês;
- (b) leia os custos unitários dos sete recursos envolvidos;
- (c) leia um conjunto indeterminado de dados (no máximo 15) contendo os quantitativos de recursos envolvidos em cada atividade;

- (d) calcule e imprima:
- i. o preço unitário de custo (direto+administração) de cada atividade;
 - ii. o preço unitário que a empreiteira deve cobrar em cada atividade para que tenha 36
 - iii. considerando o percentual de 16% para as leis sociais, incidentes sobre a mão-de-obra, quanto deve ser recolhido para cada unidade de atividade;
 - iv. considerando o percentual de administração fornecido +36% de lucro +16% de leis sociais, qual será o preço a ser cobrado pela empreiteira para a construção de uma obra que envolva as seguintes atividades:
 - A. 50 m de fundação,
 - B. 132 m de alvenaria,
 - C. 200 m de estrutura,
 - D. 339 m de acabamento;
63. Escreva um programa que receba a quantidade de alunos de uma turma, as médias de cada aluno e imprima na tela a média geral da turma e a maior e a menor média.
64. Complemente o programa do exercício anterior. Calcule e imprima o número de alunos que ficaram com média acima e abaixo da média geral da turma.
65. (SPOJ FEYNMAN - 3742. Feynman) Richard Phillips Feynman era uma físico americano muito famoso e ganhador do Prêmio Nobel de Física. Ele trabalhava em física teórica e também foi pioneiro no campo da computação quântica. Ele visitou a América do Sul por dez meses, dando palestras e aproveitando a vida nos trópicos. Ele também é conhecido pelos livros "Surely You're Joking, Mr. Feynman!" e "What Do You Care What Other People Think?", que inclui algumas de suas aventuras abaixo do equador.

Sua paixão da vida inteira era resolver e criar quebra-cabeças, trancas e códigos. Recentemente, um fazendeiro idoso da América do Sul, que hospedou o jovem físico em 1949, achou alguns papéis e notas que acredita-se terem pertencido a Feynman. Entre anotações sobre mesões e eletromagnetismo, havia um guardanapo onde ele escreveu um simples desafio: "quantos quadrados diferentes existem em um quadriculado de $N \times N$ quadrados?".

No mesmo guardanapo havia um desenho, que está reproduzido abaixo, mostrando que para $N = 2$, a resposta é 5.



Entrada

A entrada contém diversos casos de teste. Cada caso de teste é composto de uma única linha, contendo apenas um inteiro N , representando o número de quadrados em cada lado do quadriculado ($1 \leq N \leq 100$).

O final da entrada é indicado por uma linha contendo apenas um zero.

Saída

Para cada caso de teste na entrada, seu programa deve imprimir uma única linha, contendo o número de diferentes quadrados para a entrada correspondente.

Exemplo de entrada

```
2
1
8
0
```

Saída para o exemplo de entrada

```
5
1
204
```

66. (SPOJ JGARCOM - 8698. Garçom) Parte do treinamento de um novo garçom é carregar uma grande bandeja com várias latas de bebidas e copos e entregá-las todas numa mesa do restaurante. Durante o treinamento é comum que os garçons deixem cair as bandejas, quebrando todos os copos.

A SBC - Sociedade Brasileira de Copos - analisou estatísticas do treinamento de diversos garçons e descobriu que os garçons em treinamento deixam cair apenas bandejas que têm mais latas de bebidas que copos.

Por exemplo, se uma bandeja tiver 10 latas e 4 copos, certamente o garçom em treinamento a deixará cair, quebrando os 4 copos. Já se a bandeja tiver 5 latas e 6 copos, ele conseguirá entregá-la sem deixar cair.

Tarefa

Escreva um programa que, dado o número de latas e copos em cada bandeja que o garçom tentou entregar, imprime o total de copos que ele quebrou.

Entrada

A primeira linha da entrada contém um inteiro N ($1 \leq N \leq 100$) representando o número de bandejas que o garçom tentou entregar. $A \leq N$ linhas seguintes representam as N bandejas. Cada linha contém dois inteiros L e C , indicando o número de latas e o número de copos naquela bandeja, respectivamente ($0 \leq L, C \leq 100$).

Saída

Seu programa deve imprimir uma única linha, contendo um único inteiro, indicando o número total de copos que o garçom quebrou.

Exemplo

Entrada

```
3
10 5
6 8
3 3
```

Saída

```
5
```

Entrada

```
4
10 6
8 8
5 1
100 100
```

Saída

```
7
```

67. (SPOJ PAPRIMAS - 2284. Palavras Primas) Um número primo é um número que possui somente dois divisores: ele mesmo e o número 1. Exemplos de números primos são: 1, 2, 3, 5, 17, 101 e 10007.

Neste problema você deve ler um conjunto de palavras, onde cada palavra é composta somente por letras no intervalo a-z e A-Z. Cada letra possui um valor específico, a letra a vale 1, a letra b vale 2 e assim por diante, até a letra z, que vale 26. Do mesmo modo, a letra A vale 27, a letra B vale 28 e a letra Z vale 52.

Você deve escrever um programa para determinar se uma palavra é uma palavra prima ou não. Uma palavra é uma palavra prima se a soma de suas letras é um número primo.

Entrada

A entrada consiste de um conjunto de palavras. Cada palavra está sozinha em uma linha e possui L letras, onde $1 \leq L \leq 20$. A entrada é terminada por fim de arquivo (EOF).

Saída

Para cada palavra você imprimir: It is a prime word., se a soma das letras da palavra é um número primo, caso contrário você deve imprimir It is not a prime word..

Exemplo

Entrada:

UFRN

contest

AcM

Saída:

It is a prime word.

It is not a prime word.

It is not a prime word.

3 Vetores

1. Escreva um programa que receba e some dois vetores de mesma dimensão. Imprima a soma dos elementos do vetor resultante.
2. Escreva uma programa que leia dois vetores de mesma dimensão, calcule e imprima seu produto escalar.
3. Dado um vetor de no máximo 100 elementos numéricos, faça uma rotina que verifique se existe um elemento igual a K(chave) no vetor. Se existir, imprima a posição (índice) onde foi encontrada a chave; se não, imprima uma mensagem avisando o usuário que a chave não foi encontrada.
4. Escreva um programa que leia um conjunto de elementos numéricos e remova todos os valores duplicados, movendo os valores únicos em direção ao início do vetor. O programa deve mostrar o vetor resultante.

Exemplo:

Vetor lido $\rightarrow n = 8 : A = 1, 3, 1, 2, 0, 1, 0, 4$

Vetor resultante $\rightarrow A = 1, 3, 2, 0, 4$

5. Escreva um programa que receba um vetor numérico de no máximo 20 elementos. Classifique-o em ordem crescente e imprima o vetor resultante.
6. Escreva um programa que leia dois vetores A e B, de tamanho máximo 10 e 20, respectivamente, e gere o vetor X correspondente à união dos vetores A e B, gere o vetor Y com os elementos comuns de A e B e gere o vetor Z com os elementos de A que não estão em B.
7. Considere dois vetores de números inteiros A e B de tamanho T1 e T2, respectivamente. Faça um programa que leia os vetores A e B e, gere o vetor C intercalando os elementos de A e B.

Exemplo:

A	23	8	9					
B	69	4	1.2	3.5	27			
C	23	69	8	4	9	1.2	3.5	27

8. Dados um inteiro positivo N e uma sequência de N números inteiros, escreva um programa que determine o comprimento de um segmento crescente de comprimento máximo.
Exemplos:

Na sequência 5,10,2,5,7,8,4,2,3 o comprimento do segmento crescente máximo é 4.

Na sequência 9,5,4,3,1 o comprimento do segmento crescente máximo é 1.

9. Dadas duas sequências com N números inteiros entre 0 e 9, interpretadas como dois números inteiros de N algarismos, escreva um programa que calcule a sequência de números que representa a soma dos dois inteiros.

Exemplo: N = 8

$$\begin{array}{r}
 \text{1a sequência} \quad 8 \ 2 \ 4 \ 3 \ 4 \ 2 \ 5 \ 1 \\
 \text{2a sequência} \quad + \quad 3 \ 3 \ 7 \ 5 \ 2 \ 3 \ 3 \ 7 \\
 \hline
 1 \ 1 \ 6 \ 1 \ 8 \ 6 \ 5 \ 8 \ 8
 \end{array}$$

10. Dada uma sequência de N números reais, faça um programa que determine os números que compõem a sequência e o número de vezes que cada um deles ocorre na mesma.

Exemplo: N = 8

Sequência: -1.7, 3.0, 0.0, 1.5, 0.0, -1.7, 2.3, -1.7

Saída:

-1.7 ocorre 3 vezes

3.0 ocorre 1 vez

0.0 ocorre 2 vezes

1.5 ocorre 1 vez

2.3 ocorre 1 vez

11. Dados dois números naturais M e N e duas sequências ordenadas com M e N números inteiros, escreva um programa que mostre uma única sequência ordenada contendo todos os elementos das sequências originais sem repetição.
12. Leia 20 números inteiros e armazene em um vetor NIT. O algoritmo deve percorrer o vetor NIT e separar os números pares em um vetor PARES e os ímpares em um vetor IMPARES. Ao final listar os vetores PARES e IMPARES.
13. Fazer um algoritmo que leia 10 caracteres, que serão fornecidos pelo usuário armazene em um vetor LETRA, do tipo L (tamanho 10). O algoritmo deverá ser capaz de classificar a letra lida, se for vogal armazenar num vetor VOG (mesmo tipo e tamanho) se for CONSOANTE, num vetor CONS (mesmo tipo e tamanho). Ao final listar os vetores VOG e CONS.

14. Implemente um programa que receba uma frase e calcule e mostre a quantidade de palavras da frase digitada.
15. Implemente um programa para criptografar uma frase entrada pelo usuário, trocando as vogais da frase por *.
Exemplo:
Entrada: Eu estou estudando.
Saída: ** *st** *st*d*nd*.
16. Implemente um programa que receba uma frase, calcule e mostre a quantidade de vogais da frase digitada.
17. Implemente um programa que receba uma frase, calcule e mostre a quantidade de consoantes da frase digitada.
18. Implemente um programa que receba uma frase e mostre cada palavra desta frase em uma linha separada.
19. Dada uma sequência x_1, x_2, \dots, x_k de números inteiros, escreva um programa que verifique se existem dois segmentos consecutivos iguais nesta sequência, isto é, se existem i e m tais que:

$$x_i, x_{i+1}, \dots, x_{i+m-1} = x_{i+m}, x_{i+m+1}, \dots, x_{i+2m-1}$$

O programa deve mostrar, caso existam, os valores de i e m .

Exemplo: Na sequência 7,9,5,4,5,4,8 existem $i = 3$ e $m = 2$.

20. Um armazém trabalha com 20 mercadorias diferentes identificadas pelos números inteiros de 1 a 20. O dono do armazém anota a quantidade de cada mercadoria vendida durante o mês. Ele tem uma tabela que indica para cada mercadoria o preço de venda. Escreva um programa para calcular o faturamento mensal do armazém, isto é:

As tabelas de preço e de quantidade são armazenadas em dois vetores distintos, sendo que um conjunto contém a quantidade vendida e o outro o preço de cada mercadoria.

$$faturamento = \sum_{i=1}^{40} (quantidade_i * preço_i)$$

21. Faça um programa que receba dois conjuntos A e B. O seu programa deverá executar uma das quatro opções abaixo a ser escolhida pelo usuário:

(a) Opção 1 $A \cap B$

(b) Opção 2 $A - B$

(c) Opção 3 $A \subset B$

(d) Opção 4 Sair do programa

Fase 1: *Entrada dos Dados*

Em primeiro lugar, solicite ao usuário a quantidade de elementos do conjunto A. Os conjuntos A e B não podem ser vazios, ou seja, o número de elementos deve ser maior que zero. Além disso, o número MÁXIMO de elementos nesses conjuntos não pode ser maior que 50. Caso o usuário digite um valor fora do intervalo $[1, 50]$, solicite novamente o número de elementos do conjunto A. Repita esse processo até que um número dentro desse intervalo seja digitado. Após saber a quantidade de elementos do conjunto A, solicite cada elemento e armazene-o no conjunto. Os elementos deverão ser informados em ordem crescente e sem repetição. Repita esse mesmo procedimento para o conjunto B.

Fase 2: *Consulta*

Após os conjuntos terem sido informados, o usuário poderá escolher uma das quatro opções abaixo:

1 Calcular e imprimir na tela $A \cap B$

2 Calcular e imprimir na tela $A - B$

3 Imprimir na tela 1 (VERDADEIRO) se $A \subset B$ ou 0 (FALSO), caso contrário

4 Sair do programa

O programa deverá continuar na Fase de Consulta enquanto o usuário não escolher a opção 4.

Dados de entrada (na ordem de aparecimento para o usuário):

- Quantidade de elementos do conjunto A $[1, 50]$;
- Elementos do conjunto A em ordem crescente e sem repetição;
- Quantidade de elementos do conjunto B $[1, 50]$;
- Elementos do conjunto B em ordem crescente e sem repetição;
- Opção.

Dados de saída:

Dependendo da opção fornecida, exibir:

(1) $A \cap B$

Os elementos do conjunto $A \cap B$ devem ser impressos em ordem crescente e separados por um espaço. Caso o resultado da operação seja um conjunto vazio, a mensagem vazio deve ser impressa.

(2) $A - B$

Os elementos do conjunto $A - B$ devem ser impressos em ordem crescente e separados por um espaço. Caso o resultado da operação seja um conjunto vazio, a mensagem vazio deve ser impressa.

(3) $A \subset B$

Se $A \subset B$ então a saída deverá ser igual a 1 (VERDADEIRO). Caso contrário, a saída deverá ser 0 (FALSO).

(4) Sai do programa.

Exemplo de execução do programa (azul → entrada — vermelho → saída):

```
5 //Quantidade de elementos do conjunto A
1 3 5 7 9 //Conjunto A em ordem crescente e sem repetição
10 //Quantidade de elementos do conjunto B
0 1 2 3 4 5 6 7 8 9 //Conjunto B em ordem crescente e sem repetição
1 //Opção 1
1 3 5 7 9 //A ∩ B
2 //Opção 2
vazio //A - B
3 //Opção 3
1 //A ⊂ B
4 //Opção 4
```

22. Faça a declaração em linguagem C de duas variáveis do tipo numérica inteira, três variáveis do tipo numérica real, duas variáveis de um único caractere e três variáveis do tipo cadeia de caracteres com 20 caracteres.

23. Escreva o número de elementos de cada um dos vetores abaixo e a quantidade de memória que cada um ocupa.

(a) vet: Array [-15..15] Of Word;

(b) nome: Array [0..12] Of String[30];

(c) nota: Array [1..35] Of Real;

(d) v: Array [1..n] Of Byte;

(e) y: Array [-22..29] Of LongInt;

(f) x: Array [-31...110] Of Integer;

24. Atribua ao quinto elemento de um vetor numérico o valor $b^2 - 4 * a * c$.

25. Dado o vetor vet definido por:

```
Type v = Array [1..100] Of Integer;  
Var vet: v;
```

(a) preencha vet com o valor 30;

(b) preencha o vetor com os números 1, 2, 3,..., 100;

(c) preencha vet[j] com 1, se j é um quadrado perfeito, e com 0, nos demais casos.

26. Dado o seguinte vetor:

VET	3	2	4	1	2	5	8	3
-----	---	---	---	---	---	---	---	---

Qual será a sua configuração depois de executados os comandos:

```
{ ... }  
For i := 8 DownTo 5 Do  
Begin  
    aux := vet[i];  
    vet[i] := vet[8-i+1];  
    vet[8-i+1] := aux;  
End;  
vet[3] := vet[1];  
vet[vet[3]] := vet[vet[2]];  
{ ... }
```

27. Escreva um programa, com reprocessamento, que calcule a soma dos elementos de um vetor.

28. O que será impresso no programa abaixo?

```
{ $R+ }  
Type v = Array [1..6] Of Integer;  
      c = Array [1..6] Of String[3];  
Var i: Integer;  
    ve: v;  
    ca: c;  
Begin  
    ve[1] := 1;  
    ve[2] := 1;  
    ve[3] := 2;  
    ve[4] := 2;
```

```

ve[5] := 5;
ve[6] := 6;
ca[1] := 'seg';
ca[2] := 'ter';
ca[3] := 'qua';
ca[5] := 'sex';
ca[6] := 'sab';
i := 1;
While i \leq 6 Do
Begin
    WriteLn (ca[ve[i]]);
    Inc (i,2);
End;
WriteLn (ca[ve[ve[3]]]);
End.

```

29. Qual será o valor de x impresso pelo programa abaixo?

```

{$R+}
Type vetor = Array [1..5] Of Real;
Var v: vetor;
    x: Real;
    i: Integer;
Begin
    v[1] := 2;
    v[2] := 4;
    v[3] := 1;
    v[4] := 3;
    v[5] := 5;
    x := v[1] + v[5];
    WriteLn (x);
    x := v[2] - v[5];
    WriteLn (x);
    x := v[4] * v[1] - x;
    WriteLn (x);
    i := 3;
    x := v[i];
    WriteLn (x);
    x := v[i] / v[v[i]];
    WriteLn (x);
End.

```

30. O que pode estar errado com o programa abaixo?

```

Var a, b, i, l, p, c: Integer;
Begin
    a := 1;
    b := 1;
    ReadLn (l, p);

```



```

i := 1;
While i \leq 1 Do
Begin
    c := a + b;
    WriteLn (c);
    a := b;
    b := c;
    Inc (i, p);
End;
End.

```

31. Escreva um programa que leia um conjunto A de 20 elementos, calcule e imprima o valor de S, onde:

$$S = (A_1 - A_{20})^2 + (A_2 - A_{19})^2 + \dots + (A_{10} - A_{11})^2$$

32. Escreva um programa, com reprocessamento, que some dois vetores de mesma dimensão.
33. Escreva um programa, com reprocessamento, que leia dois vetores de mesma dimensão, calcule e imprima seu produto escalar.
34. Dado um vetor de no máximo 120 elementos numéricos, faça um programa que verifique se existe um elemento igual a K (chave) no vetor. Se existir, imprima a posição onde foi encontrada a chave; se não, imprima uma mensagem avisando o usuário que a chave não foi encontrada.
35. Escreva um programa que preencha um vetor numérico, de no máximo 40 posições, com valores aleatórios distintos de 150 a 320
36. Dado o seguinte vetor de caracteres:

VET

T	R	X	S		E	O	B	A	!
---	---	---	---	--	---	---	---	---	---

Qual será a sua configuração após serem executados os comandos a seguir?

```

{ ... }
aux := vet[6];
vet[6] := vet[9];
vet[9] := aux;
For i := 1 To 4 Do
Begin
    aux := vet[i];
    vet[i] := vet[9-i];

```

```

    vet[9-i] := aux;
End;
vet[6] := vet[2];
{ ... }

```

37. Faça um programa, com reprocessamento, para calcular o número de alunos que tiraram nota acima da média da turma. A turma tem no máximo 40 alunos.
38. Escreva um programa, com reprocessamento, que leia um conjunto A de no máximo 100 elementos e, construa e imprima um outro conjunto B formado da seguinte maneira:
- os elementos de ordem par são os correspondentes de A divididos por 2;
 - os elementos de ordem ímpar são os correspondentes de A multiplicados por 3.

Exemplo:

A	23	8	0.4	7	9
B	69	4	1.2	3.5	27

39. Faça um programa, com reprocessamento, que imprima o maior e o menor elemento de um vetor inteiro de no máximo 25 elementos.
40. Escreva um programa que leia um conjunto de elementos numéricos (máximo 100) e remova todos os valores duplicados, movendo os valores únicos em direção ao início do vetor. O programa deve mostrar o vetor resultante.

Exemplo:

Vetor lido:								
n=8								
A	1	3	1	2	0	1	0	4
Vetor resultante:								
n=5								
A	1	3	2	0	4			

41. Escreva um programa, com reprocessamento, que classifique um vetor numérico de no máximo de 20 elementos em ordem crescente.
42. Escreva um programa, com reprocessamento, que classifique um vetor numérico de no máximo de 20 elementos em ordem decrescente.

43. Escreva um programa, com reprocessamento, que classifique um vetor alfanumérico de no máximo de 20 elementos em ordem crescente.
44. Uma lista com 30 nomes deveria estar em ordem alfabética, suspeita-se que algum desarranjo ocorreu devido à correção de um determinado nome. Escreva um programa que percorra a lista e verifique se a classificação foi alterada. Caso isso tenha ocorrido, reposicione o nome alterado, deslocando os outros nomes.
45. Escreva um programa, com reprocessamento, que leia dois vetores A e B, de tamanho máximo 10 e 20, respectivamente, e gere o vetor X correspondente à união dos vetores A e B, gere o vetor Y com os elementos comuns de A e B e gere o vetor Z com os elementos de A que não estão em B.
46. Escreva um programa, com reprocessamento, que procure uma cadeia de caracteres em um vetor alfanumérico de n posições. O término da busca ocorre quando se acha a cadeia de caracteres ou quando se percorreu todo o vetor e o dado não foi encontrado.
47. Considere dois vetores de números inteiros A e B de tamanho T1 e T2, respectivamente. Faça um programa, com reprocessamento, que leia os vetores A e B e, gere o vetor C intercalando os elementos de A e B.

Exemplo:

A	23	8	9					
B	69	4	1.2	3.5	27			
C	23	69	8	4	9	1.2	3.5	27

48. Considere dois vetores alfanuméricos A e B de tamanho T1 e T2, respectivamente. Faça um programa que leia os vetores A e B, classifique-os e, gere o vetor C classificado.

Exemplo:

Vetor A	Maria	João	José	Ana			
Vetor B	Cida	Zeca	Benedito				
Vetor C	Ana	Benedito	Cida	João	José	Maria	Zeca

49. Escreva um programa, com reprocessamento, que leia dois conjuntos de caracteres de no máximo 50 caracteres cada e, determine e imprima o conjunto intersecção e união entre estes conjuntos de caracteres.
50. Escreva o número de elementos, a dimensão e a quantidade de memória ocupada pelas matrizes declaradas abaixo:

```

Var m1: Array [0..2,1..15] Of Real;
    m2: Array [-10..10,-15..15] Of Integer;
    m3: Array [1..3,2..6,10..14] Of Char;
    m4: Array [0..5,0..5,0..5,0..5] Of Byte;
    m5: Array [1..15,1..15] Of String[10];

```

51. Escreva um programa, com reprocessamento, que leia dois vetores de mesma dimensão, calcule e imprima seu produto escalar. Utilize o conceito de rotina.
52. Dado um vetor de no máximo 120 elementos numéricos, faça uma rotina que verifique se existe um elemento igual a K (chave) no vetor. Se existir, imprima a posição onde foi encontrada a chave; se não, imprima uma mensagem avisando o usuário que a chave não foi encontrada. Faça um programa que utilize essa rotina.
53. Faça uma rotina para calcular o número de alunos que tiraram nota acima da média da turma. A turma tem no máximo 60 alunos. Faça um programa, com reprocessamento, que utilize essa rotina.
54. Faça uma rotina que imprima o maior e o menor elemento de um vetor inteiro de no máximo 25 elementos. Faça um programa que utilize essa rotina.
55. Escreva uma rotina que classifique um vetor numérico de no máximo de 45 elementos em ordem crescente. Faça um programa que utilize essa rotina.
56. Escreva um programa, com reprocessamento, que classifique um vetor alfanumérico de no máximo de 40 elementos em ordem decrescente. Utilize o conceito de rotina.
57. Uma lista com 35 nomes deveria estar em ordem alfabética, suspeita-se que algum desarranjo ocorreu devido à correção de um determinado nome. Escreva um programa que percorra a lista e verifique se a classificação foi alterada. Caso isso tenha ocorrido, reposicione o nome alterado, deslocando os outros nomes. Utilize o conceito de rotina.
58. (SPOJ AERO - 818. Aeroporto) A crescente utilização do transporte aéreo preocupa os especialistas, que prevêem que o congestionamento em aeroportos poderá se tornar um grande problema no futuro. Os números atuais já são alarmantes: relatórios oficiais demonstram que na Europa, em junho de 2001, houve uma média de 7.000 atrasos de vôos por dia. Preocupada com a previsão dos seus especialistas em tráfego aéreo, a Associação de Transporte Aéreo Internacional (ATAI) está começando um estudo para descobrir quais são os aeroportos onde o tráfego aéreo pode vir a ser mais problemático no futuro.

Tarefa

Como programador recém contratado pela ATAI você foi encarregado de escrever um programa para determinar, a partir de uma listagem de aeroportos e vôos, qual aeroporto possui maior probabilidade de congestionamento no futuro. Como medida da probabilidade de congestionamento será utilizado neste estudo o número total de vôos que chegam

ou que partem de cada aeroporto.

Entrada

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de teste contém dois números inteiros A e V , que indicam respectivamente o número de aeroportos e o número de vôos. Os aeroportos são identificados por inteiros de 1 a A . As V linhas seguintes contém cada uma a informação de um vôo, representada por um par de números inteiros positivos X e Y , indicando que há um vôo do aeroporto X para o aeroporto Y . O final da entrada é indicado quando $A = V = 0$.

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas. A primeira linha identifica o conjunto de teste, no formato “Teste n ”, onde n é numerado a partir de 1. A segunda linha deve conter o identificador do aeroporto que possui maior tráfego aéreo. Caso mais de um aeroporto possua este valor máximo, você deve listar todos estes aeroportos, em ordem crescente de identificação, e separados por pelo menos um espaço em branco. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo

Entrada:

```
5 7
1 3
2 1
3 2
3 4
4 5
3 5
2 5
3 5
1 3
1 2
3 2
1 2
2 1
0 0
```

Output:

```
Teste 1
3
```

```
Teste 2
1 2
```

Restrições

- $0 \leq A \leq 100$ ($A = 0$ apenas para indicar o fim da entrada)
- $0 \leq V \leq 10000$ ($V = 0$ apenas para indicar o fim da entrada)
- $1 \leq X \leq A$
- $1 \leq Y \leq A$
- $X! = Y$

‘

59. (SPOJ CALCULA - 1334. Calculando) A disseminação dos computadores se deve principalmente à capacidade de eles se comportarem como outras máquinas, vindo a substituir muitas destas. Esta flexibilidade é possível porque podemos alterar a funcionalidade de um computador, de modo que ele opere da forma que desejarmos: essa é a base do que chamamos programação.

Tarefa

Sua tarefa é escrever um programa que faça com que o computador opere como uma calculadora simples. O seu programa deve ler expressões aritméticas e produzir como saída o valor dessas expressões, como uma calculadora faria. O programa deve implementar apenas um subconjunto reduzido das operações disponíveis em uma calculadora: somas e subtrações.

Entrada

A entrada é composta de vários conjuntos de testes. A primeira linha de um conjunto de testes contém um número inteiro m ($1 \leq m \leq 100$), indicando o número de operandos da expressão a ser avaliada. A segunda linha de um conjunto de testes contém a expressão aritmética a ser avaliada, no seguinte formato:

$$X_1 s_1 X_2 s_2 \dots X_{m-1} s_{m-1} X_m$$

onde

- $X_i, 1 \leq i \leq m$, é um operando ($0 \leq X_i \leq 100$);
- $s_j, 1 \leq j \leq m$, é um operador, representado pelos símbolos ‘+’ ou ‘-’;
- não há espaços em branco entre operandos e operadores.

O final da entrada é indicado pelo valor $m = 0$.

Exemplo de Entrada

```
3
3+7-22
3
5-10-77
10
1+2+3+4+5+6+7+8+9+10
0
```

Saída

Para cada conjunto de testes da entrada seu programa deve produzir três linhas. A primeira linha deve conter um identificador da expressão, no formato "Teste n", onde n é numerado a partir de 1. Na segunda linha deve aparecer o resultado encontrado pelo seu programa. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Saída

```
Teste 1
-12

Teste 2
-82

Teste 3
55
(esta saída corresponde ao exemplo de entrada acima)
```

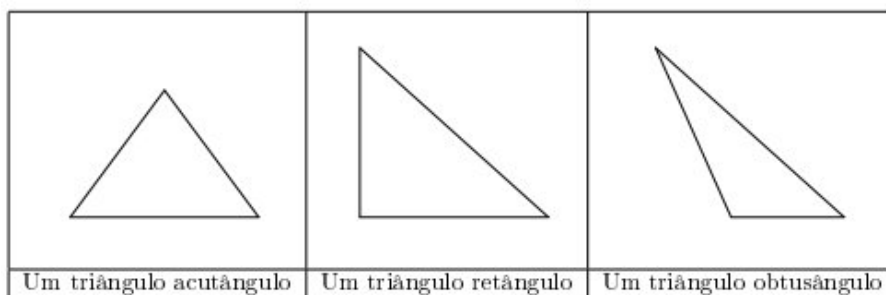
Restrições

- $1 \leq m \leq 100$
- $0 \leq X_i \leq 100$ para todo $1 \leq i \leq m$

60. (SPOJ TRIANG11 - 11004. Triângulos) Caio estava brincando de construir triângulos com palitos de diferentes tamanhos. Ele fazia isso juntando as pontas de três palitos sobre uma mesa. Ele notou que podia agrupar os triângulos formados em três grupos:

- Triângulos acutângulos, que são aqueles em que todos os ângulos internos medem menos de 90° ;
- Triângulos retângulos, que são aqueles que possuem um ângulo interno que mede exatamente 90° ;
- Triângulos obtusângulos, que são aqueles que possuem um ângulo interno que mede mais de 90° .

Ele também percebeu que nem sempre é possível formar um triângulo com três palitos.



Sua tarefa é, dados os comprimentos A, B e C de três palitos, dizer se é possível formar um triângulo com esses palitos e, em caso afirmativo, dizer a qual grupo o triângulo formado pertence.

Entrada

A entrada consiste de uma única linha, contendo três inteiros A, B e C separados por espaço.

Saída

Imprima uma linha contendo apenas uma letra minúscula:

- 'n' se não for possível formar um triângulo;
- 'a' se o triângulo formado for acutângulo;
- 'r' se o triângulo formado for retângulo;
- 'o' se o triângulo formado for obtusângulo.

Restrições

- $1 \leq A \leq 104$
- $1 \leq B \leq 104$
- $1 \leq C \leq 104$

Exemplos

Entrada

1 1 1

Saída

a

Entrada

1 2 1

Saída

n

Entrada

5 4 3

Saída

r

Entrada

6 3 4

Saída

o

61. (SPOJ POLIGONO - 3747. Encolhendo Polígonos) Um polígono é dito inscrito em um círculo quando todos seus vértices estão naquele círculo. Nesse problema você receberá um polígono inscrito em um círculo, e você deve determinar o número mínimo de vértices que devem ser removidos para transformar o polígono dado em um polígono regular, i.e., um polígono que é equiângulo (todos ângulos são congruentes) e equilátero (todos lados têm o mesmo comprimento).

Quando você remove um vértice v você primeiro remove o vértice e os segmentos de reta conectando-o aos seus vértices adjacentes w_1 e w_2 , e então você cria um novo segmento de reta conectando w_1 e w_2 . A figura (a) abaixo ilustra um polígono inscrito em um círculo, com dez vértices, e a figura (b) mostra um pentágono (polígono regular com cinco lados) formado ao remover cinco vértices do polígono em (a).

Nesse problema consideraremos que qualquer polígono deve ter pelo menos três lados.

Entrada

A entrada contém diversos casos de teste. A primeira linha de um caso de teste contém um inteiro N indicando o número de vértices do polígono inscrito ($3 \leq N \leq 10^4$). A segunda linha contém N inteiros X_i separados por espaços ($1 \leq X_i \leq 10^3$ para $0 \leq i \leq N - 1$). Cada X_i representa o comprimento do arco definido no círculo circunscrito, no sentido horário, pelos vértices i e $(i + 1) \bmod N$. Lembre-se que um arco é um segmento da circunferência de um círculo; não o confunda com corda, que é um segmento de linha cujos extremos

ambos estão no círculo.

O final da entrada é indicado por uma linha contendo apenas um zero.

Saída

Para cada caso de teste, seu programa deve imprimir uma única linha, contendo o número mínimo de vértices que precisam ser removidos do polígono dado para formar um polígono regular. Se não for possível formar um polígono regular, a linha deve conter apenas o valor -1.

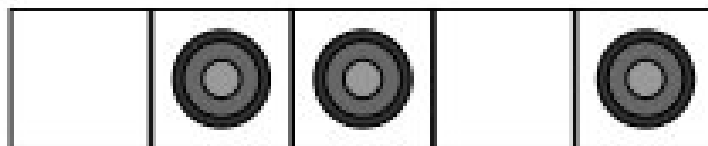
Exemplo de entrada

```
3
1000 1000 1000
6
1 2 3 1 2 3
3
1 1 2
10
10 40 20 30 30 10 10 50 24 26
0
```

Saída para o exemplo de entrada

```
0
2
-1
5
```

62. (SPOJ MINADO12 - 10864. Campo Minado) Leonardo Viana é um garoto fascinado por jogos de tabuleiro. Nas férias de janeiro, ele aprendeu um jogo chamado "Campo minado", que é jogado em um tabuleiro com N células dispostas na horizontal. O objetivo desse jogo é determinar, para cada célula do tabuleiro, o número de minas explosivas nos arredores da mesma (que são a própria célula e as células imediatamente vizinhas à direita e à esquerda, caso essas existam). Por exemplo, a figura abaixo ilustra uma possível configuração de um tabuleiro com 5 células:



A primeira célula não possui nenhuma mina explosiva, mas é vizinha de uma célula que possui uma mina explosiva. Nos arredores da segunda célula temos duas minas, e o mesmo acontece para a terceira e quarta células; a quinta célula só tem uma mina explosiva em seus arredores. A próxima figura ilustra a resposta para esse caso.

1	2	2	2	1
---	---	---	---	---

Leonardo sabe que você participa da OBI e resolveu lhe pedir para escrever um programa de computador que, dado um tabuleiro, imprima o número de minas na vizinhança de cada posição. Assim, ele poderá conferir as centenas de tabuleiros que resolveu durante as férias.

Entrada

A primeira linha da entrada contém um inteiro N indicando o número de células no tabuleiro. O tabuleiro é dado nas próximas N linhas. A i -ésima linha seguinte contém 0 se não existe mina na i -ésima célula do tabuleiro e 1 se existe uma mina na i -ésima célula do tabuleiro.

Saída

A saída é composta por N linhas. A i -ésima linha da saída contém o número de minas explosivas nos arredores da i -ésima célula do tabuleiro.

Restrições

- $1 \leq N \leq 50$

Exemplos

Entrada

```
5
0
1
1
0
1
```

Saída

```
1
2
2
2
1
```

Entrada

```
5
0
```

1
1
1
0

Saída

1
2
3
2
1

63. (SPOJ LUA - 815. Temperatura Lunar) Sem as proteções da atmosfera e do cinturão magnético que existem na Terra, a Lua fica exposta ao ataque do Sol, que é um astro em constante explosão atômica. As explosões do Sol emitem ondas letais de partículas. Uma pessoa que ficasse desprotegida na superfície da Lua, num lugar onde o Sol incidisse diretamente, sofreria um bombardeio radioativo tão intenso quanto se estivesse nas imediações da usina russa de Chernobyl no momento do acidente que matou 31 pessoas, em 1986. Além da radiação solar, outro efeito desta falta de proteção contra o Sol que existe na Lua é a enorme variação de temperatura. Nas regiões próximas do equador lunar, a variação de temperatura é brutal, passando de cerca de 130 graus positivos durante o dia a 129 graus negativos à noite.

Para estudar com mais precisão as variações de temperatura na superfície da Lua, a NASA enviou à Lua uma sonda com um sensor que mede a temperatura de 1 em 1 minuto. Um dado importante que os pesquisadores desejam descobrir é como se comporta a média da temperatura, considerada em intervalos de uma dada duração (uma hora, meia hora, oito horas, etc.). Por exemplo, para a sequência de medições 8, 20, 30, 50, 40, 20, -10, e intervalos de quatro minutos, as médias são respectivamente $108/4=27$, $140/4=35$, $140/4=35$ e $100/4=25$.

Tarefa

Você foi recentemente contratado pela NASA, e sua primeira tarefa é escrever um programa que, conhecidos a sequência de temperaturas medidas pelo sensor, e o tamanho do intervalo desejado, informe qual a maior e qual a menor temperatura média observadas, considerando o tamanho do intervalo dado.

Entrada

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de teste contém dois números inteiros positivos N e M , que indicam respectivamente o número total de medições de temperatura de uma sequência obtida pelo sensor, e o tamanho dos intervalos, em minutos, em que as médias devem ser calculadas. As N linhas seguintes contém um número inteiro cada, representando a sequência de medidas do sensor. O final da entrada é indicado quando $N = M = 0$.

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas. A primeira linha identifica o conjunto de teste, no formato “Teste n”, onde n é numerado a partir de 1. A segunda linha deve conter dois números inteiros X e Y, separados por ao menos um espaço em branco, representando respectivamente os valores da menor e da maior média de temperatura, conforme determinado pelo seu programa. O valor da média deve ser truncado, se a média não for um número inteiro (ou seja, deve ser impressa apenas a parte inteira). A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo

Entrada:

```
4 2
-5
-12
0
6
7 4
35
-35
5
100
100
50
50
0 0
```

Saída:

```
Teste 1
-8 3
```

```
Teste 2
26 75
```

Restrições

- $0 \leq N \leq 10000$ ($N = 0$ apenas para indicar o fim da entrada)
- $-200 \leq \text{Temperatura} \leq 200$
- $1 \leq M \leq N$

4 Matrizes

1. Escreva um programa que inicialize uma matriz 10×10 com 0s em todas as posições. O usuário irá digitar o índice da linha e o índice da coluna e em seguida o valor das posições não nulas. A leitura será feita enquanto os índices forem não negativos. Após a leitura

imprima a matriz na tela.

Exemplo:

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \dots & & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} 0 & 0 & -1 \\ 1 & 2 & 2 \\ 9 & 9 & 1 \\ -1 & -1 \end{matrix}$$

2. Escreva um programa que leia todas as posições de uma matriz 10×10 . Em seguida, mostra o índice da linha, o índice da coluna e o valor das posições não nulas. No final, exibe o número de posições não nulas.

Exemplo:

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \dots & & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} 0 & 0 & -1 \\ 1 & 2 & 2 \\ 9 & 9 & 1 \\ 3 & \text{elementos não} \\ & \text{nulos} \end{matrix}$$

3. Dados $M \in R^{n \times n}$ e um escalar k , escreva um programa que leia os elementos da matriz e a escreva após ter multiplicado os elementos da diagonal principal k .
4. Escreva um programa que leia uma tabela de $m \times n$ elementos, calcule e imprima a soma de cada linha e a soma de todos os elementos.
5. Escreva um programa que lê todos os elementos de uma matriz 4×4 e mostra a matriz e a sua transposta na tela.

Matriz	Transposta
$\begin{bmatrix} 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 2 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 \end{bmatrix}$

6. Escreva um programa que lê 2 matrizes 5×5 , mostre-as na tela e exiba a soma entre as duas matrizes em seguida.

Exemplo:

A		B		C
$\begin{bmatrix} 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \end{bmatrix}$	+	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \end{bmatrix}$	=	$\begin{bmatrix} 0 & 1 & 0 & 4 & 3 \\ 1 & 2 & 1 & 3 & 4 \\ 0 & 1 & 0 & 2 & 3 \\ 2 & 3 & 2 & 4 & 5 \\ 3 & 4 & 3 & 5 & 6 \end{bmatrix}$

7. Escreva um programa que leia uma matriz de ordem n , calcule e imprima a soma dos elementos situados abaixo da diagonal principal da matriz, incluindo os elementos da própria diagonal principal
8. Escreva um programa que lê duas matrizes 5×5 , mostre-as na tela e então calcule o produto entre as duas matrizes, mostrando-o em seguida.
Exemplo:

$$\begin{matrix} & & \text{A} & & & & \text{B} & & & & \text{C} \\ \begin{bmatrix} 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \\ 0 & 1 & 0 & 2 & 3 \end{bmatrix} & * & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 \end{bmatrix} & = & \begin{bmatrix} 14 & 14 & 14 & 14 & 14 \\ 14 & 14 & 14 & 14 & 14 \\ 14 & 14 & 14 & 14 & 14 \\ 14 & 14 & 14 & 14 & 14 \\ 14 & 14 & 14 & 14 & 14 \end{bmatrix} \end{matrix}$$

9. Escreva um programa que lê uma matriz e depois verifica se esta é uma matriz triangular inferior.
A imagem abaixo é Triangular Inferior:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 & 3 \end{bmatrix}$$

10. Os elementos a_{ij} de uma matriz inteira A representam os custos de transporte da cidade i para a cidade j . Dada a matriz de custo A, n itinerários, cada um com k cidades, escreva um programa que determine o custo total para cada itinerário.
Exemplo: A =

$$\begin{pmatrix} 4 & 1 & 2 & 3 \\ 5 & 2 & 1 & 400 \\ 2 & 1 & 3 & 8 \\ 7 & 1 & 2 & 5 \end{pmatrix}$$

O custo do itinerário (1 4 2 4 4 3 2 1) é: $a_{14} + a_{42} + a_{24} + a_{44} + a_{43} + a_{32} + a_{21} = 417$

11. Uma matriz quadrada inteira é um quadrado mágico se a soma dos elementos de cada linha, a soma dos elementos da cada coluna e a soma dos elementos das diagonais principal e secundária são todas iguais.
Exemplo: A matriz abaixo é um quadrado mágico

$$\begin{pmatrix} 8 & 0 & 7 \\ 4 & 5 & 6 \\ 3 & 10 & 2 \end{pmatrix}$$

Escreva um programa que verifique se uma matriz quadrada é um quadrado mágico.

12. Sudoku é jogado numa malha de 9×9 quadrados, dividida em sub-malhas de 3×3 quadrados, chamada "quadrantes". O objetivo do jogo é preencher os quadrados com números entre 1 e 9 de acordo com as seguintes regras:

- Cada número pode aparecer apenas uma vez em cada linha.
- Cada número pode aparecer apenas uma vez em cada coluna.
- Cada número pode aparecer apenas uma vez em cada quadrante.

Exemplo:

9	5	3	4	8	6	2	7	1
1	2	7	9	3	5	8	4	6
6	8	4	7	1	2	9	3	5
5	6	8	3	9	1	4	2	7
4	9	1	2	6	7	3	5	8
3	7	2	8	5	4	1	6	9
7	4	9	5	2	8	6	1	3
2	3	6	1	7	9	5	8	4
8	1	5	6	4	3	7	9	2

Escreva um programa que lê um jogo de *Sudoku* (matriz 9×9 , toda preenchida com números de 1 a 9) e verifica se é um jogo válido ou não. Um jogo válido respeita as três regras acima.

13. Criar um programa que armazene dados inteiros em uma matriz quadrada de ordem máxima 5 e mostre toda a matriz e a raiz quadrada da soma dos quadrados dos números ímpares armazenados na matriz.
14. Construir uma grade 10×10 na tela, utilizando os símbolos do teclado: '|', ' ', '-|'.
15. Ler valores inteiros para a matriz $A[3][5]$. Gerar e mostrar o vetor SOMA_LINHA, onde cada elemento é a soma dos elementos de uma linha da matriz A. Faça o trecho, que gera a matriz, separado da entrada e da saída.
16. Crie o vetor SOMA_COLUNA para o exercício anterior.
17. Entrar com os valores para uma matriz $C[2][3]$. Gerar e mostrar C^t . A matriz transposta é gerada trocando linha por coluna.

18. Entrar com uma matriz $M_{2 \times 2}$. Calcular e mostrar o determinante. Para o cálculo do determinante de uma matriz de ordem 2, faça:
- (a) Multiplique os elementos da diagonal principal ($a = M_{1,1} * M_{2,2}$)
 - (b) Multiplique os elementos da diagonal secundária ($b = M_{1,2} * M_{2,1}$)
 - (c) Subtraia b de a.
19. Crie um algoritmo que leia, armazene e mostre os elementos de uma matriz inteira $M_{5 \times 5}$. A seguir, troque:
- (a) A segunda linha pela quarta linha
 - (b) A terceira coluna pela quinta coluna
 - (c) A diagonal principal pela diagonal secundária
20. Ler diversas cadeias (pelo menos uma e parar de ler cadeias quando o usuário digitar a cadeia vazia) de no máximo 20 e no mínimo 5 caracteres cada uma. Fazer a validação desses limites, emitindo a mensagem Cadeia inválida - redigite e fazer o cursor voltar para o início da cadeia digitada. Não esqueça de apagar a mensagem quando for digitada a cadeia válida. Se os primeiros dois caracteres da cadeia forem iguais aos dois últimos, reimprimir a cadeia acrescida de seus três caracteres iniciais no fim e seus três finais no início (por exemplo, cadeia digitada: abcdefab; cadeia impressa: fababcdefababc); caso contrário, imprimir a cadeia sem o caractere do meio (se comprimento for ímpar) ou sem os dois do meio (se comprimento for par). Quando o usuário finalizar a entrada (digitar a cadeia vazia), exibir o resumo abaixo considerando as cadeias originais válidas e encerrar o programa:
- Cadeia mais longa: XX caracteres
Comprimento médio das cadeias: XX caracteres
Cadeias com a letra "s": XXX cadeias
- Dica:** veja as funções: *delete*, *copy*, *pos*, *concat*
21. Faça um programa que leia linha a linha os elementos de uma matriz de 5 X 20 e mostre-os coluna a coluna.
22. Faça um programa que monte um vetor tridimensional 10 X 30 X 3, onde o conteúdo de cada elemento é igual à soma dos valores de seus índices.
23. Faça um programa que leia duas matrizes inteiras 3 X 3, calcule a soma e a subtração entre elas. Ao final, o programa deverá mostrar as duas matrizes lidas bem como as matrizes resultantes de cada uma das operações.

24. Faça um programa que calcule a matriz dada por $M1(2 \times 3) = 2i + j$, calcule sua transposta e, ao final, mostre a matriz M1 e a sua transposta.
25. Faça um programa que leia um número K e uma matriz M(3X4), calcule uma nova matriz dada pela multiplicação K por M e mostre, ao final, a matriz M lida e a matriz resultante da operação.
26. As antigas lendas imperiais romanas dizem que o imperador César usava um esquema muito simples para criptografar as mensagens que enviava para seus generais. O esquema funcionava da seguinte maneira, cada letra da mensagem era substituída pela letra seguinte no alfabeto. A última letra era substituída pela primeira letra do alfabeto. Este esquema era chamado de codificação com salto igual a 1. A sua tarefa é escrever um programa que decodifique uma mensagem escrita com este código de salto 1.

Exemplo 1

- entrada: bvmb qsbujdb
- saída: aula pratica

Exemplo 1

- entrada: dftbs jnqfsbeps spnbop
- saída: ...?

27. Escreva um programa que leia da entrada padrão uma matriz de tamanho $n \times m$ (n e m fornecidos pelo usuário) e escreva na saída padrão o valor máximo e o valor mínimo existentes naquela matriz. Considere que o usuário nunca fornecerá n e m maiores que 100.
28. Escreva um programa que leia uma matriz $m \times n$ e gere um vetor **vetMenores** de dimensão n composto pelos menores elementos de cada coluna da matriz.
29. Escreva um programa que leia um valor n da entrada padrão e preencha uma matriz de forma que ela torne-se a matriz identidade de tamanho n (I_n) e depois imprima essa mesma matriz na saída padrão. O tamanho máximo da matriz deve ser limitado pela constante LIM, definida como **#define LIM 100** no começo do seu programa. Uma matriz identidade I é definida por:

$$I_n = (i_{x,y})_n$$

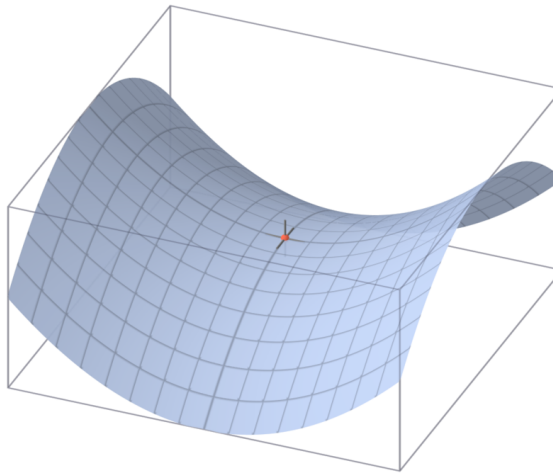
$$i_{x,y} = \begin{cases} 1 & , se \ x = y \\ 0 & , se \ x \neq y \end{cases}$$

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

30. Escreva um programa que leia uma matriz quadrada M de dimensões $n \times n$ (n fornecido pelo usuário), imprima a sua transposta M^t . Considere que o usuário nunca digitará valores de n maiores que 100.

$$M = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}, \quad M^t = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$$

31. Em cálculo, um *ponto de sela* é um ponto no domínio de uma função de duas variáveis que é um local estacionário mas não um local extremo. Nesse ponto, em geral, a superfície é semelhante a uma sela que desce em uma direção e sobe em outra, como mostra a figura abaixo.



Dada uma matriz quadrada M de ordem n composta por números inteiros, escreva um programa que verifica se ela possui um ponto de sela. Um ponto de sela em M é um elemento na posição (i, j) de tal modo que M_{ij} é máximo na linha i e mínimo na coluna j ou (i, j) é mínimo na linha i e máximo na coluna j . Se existir, o programa deverá exibir a resposta *sim* seguido da linha i e da coluna j do ponto. Caso contrário, o programa deverá imprimir a resposta *nao*.

Dados de Entrada:

- Ordem da matriz $M[1, 50]$;
- Elementos da matriz M .

Dados de Saída

- Resposta. Se houver ponto de sela na matriz M , imprimir a resposta *sim* seguido do índice da linha e da coluna do ponto na matriz M . Caso contrário, a resposta deverá ser *nao*.

Exemplo de execução (entrada / saída):

```
4           // Ordem da matriz M
0 1 9 3     // Primeira linha de elementos de M
8 5 6 7     // Segunda linha de elementos de M
1 3 7 2     // Terceira linha de elementos de M
6 2 8 0     // Quarta linha de elementos de M

sim         // Saída do programa
2 2         // Posições i e j do ponto de sela }
```

32. O que será impresso no programa abaixo?

```

{$R+}
Type m = Array [1..3,1..2] Of Integer;
Var i, j: Byte;
    m1: m;
Begin
    m1[1,1] := 1;
    m1[1,2] := 2;
    m1[2,1] := 3;
    m1[2,2] := 4;
    m1[3,1] := 5;
    m1[3,2] := 6;
    For i := 1 To 3 Do
        Begin
            For j := 1 To 2 Do
                Write (m1[i,j], ' ');
            WriteLn;
        End;
    For i := 1 To 2 Do
        Begin
            For j := 1 To 3 Do
                Write (m1[j,i], ' ');
            WriteLn;
        End;
    End.

```

33. Descreva o que será produzido, depois de executado os comandos abaixo, se:

A =

1	2	1
3	1	4

C =

1	3
2	1
1	4

```

For i := 1 To 2 Do
    For j := 1 To 2 Do
        For k := 1 To 3 Do
            WriteLn (a[i,k] + c[k,j]);

```

34. O que será impresso no programa abaixo?

```

{$R+}
Type matriz1 = Array [1..3,1..4] Of Integer;
    matriz2 = Array [1..2,1..2] Of Char;
Var m: matriz1;
    n: matriz2;
    i, j: Byte;
Begin
    j := 2;
    For i := 1 To 3 Do
        Begin

```

```

    m[i,j] := 2;
    m[i,j+2] := 2;
    m[i,j-1] := 1;
    m[i,j+1] := 1;
End;
For i := 1 To 2 Do
    For j := 1 To 2 Do
        If i = j Then n[i,j] := 'a'
        Else n[i,j] := 'z';
    For i := 1 To 3 Do
        For j := 1 To 5 Do
            WriteLn ('m[' , i , ' , ' , j , ' ] = ' , m[i,j]);
    For i := 1 To 2 Do
        For j := 1 To 2 Do
            WriteLn ('n[' , i , ' , ' , j , ' ] = ' , n[i,j]);
End.

```

35. Considere a seguinte variável composta bidimensional A:

	175	225	10	9000	3.7	4.75
	9.8	100	363	432	156	18
A =	40	301	30.2	6381	1	0
	402	4211	7213	992	442	7321
	21	3	2	1	9000	2000

- Quantos elementos fazem parte do conjunto?
- Qual o conteúdo do elemento identificado por $A[4, 5]$?
- Qual o conteúdo de X após a execução do comando $X := A[3, 2] + A[5, 1]$?
- O que aconteceria caso fosse referenciado o elemento $A[6, 2]$ no programa?
- Escreva um trecho de programa que some os elementos da quarta coluna.
- Escreva um trecho de programa que some os elementos da terceira linha.

36. Dada a matriz MAT abaixo:

	O	Q	.	I
MAT =	E	A	E	S
	R	E	U	T
	A	.	.	S

Escreva a configuração de MAT depois de executado o programa abaixo:

```

{$R+}
Type matriz = Array [1..4,1..4] Of Char;
Var i, j: Byte;
    aux: Char;
    mat: matriz;

```

```

Begin
  For i := 1 To 4 Do
    For j := 1 To 4 Do
      ReadLn (mat[i,j]);
    { Suponha que a matriz acima seja a matriz lida }
  For i := 1 To 4 Do
    For j := i+1 To 4 Do
      Begin
        aux := mat[i,j];
        mat[i,j] := mat[j,i];
        mat[j,i] := aux;
      End;
    aux := mat[1,1];
    mat[1,1] := mat[4,4];
    mat[4,4] := aux;
    aux := mat[2,2];
    mat[2,2] := mat[3,3];
    mat[3,3] := aux;
  End.

```

37. Dada a matriz M tridimensional definida por

```

Type mat = Array [0..1,1..4,1..3] Of Integer;
Var m: mat;

```

e dada por

M	1				2				3			
	1	2	3	4	1	1	1	1	0	0	1	1
	5	-5	3	0	-3	2	0	0	-1	-1	2	2
	1				2				3			

E, dado o vetor VET definido por

```

Type vetor = Array [1..4] Of Integer;
Var vet: vetor;

```

VET	2	3	4	0
-----	---	---	---	---

Determine os elementos:

- (a) $M[1, 1, 2]$
- (b) $VET[3]$
- (c) $VET[M[0, 1, 1]]$

- (d) $M[VET[4], VET[2], VET[1]]$
 (e) $M[M[VET[4], 4, 3], VET[M[0, 3, 1]], 2]$

38. Considere os programas

```
{R+}
Program Programa1;
Type vet = Array [1..10] Of Integer;
      mat = Array[1..10,1..10] Of Integer;
Var a: vet;
      b: mat;
      i, j, soma1, soma2: Integer;
Begin
  For i := 1 To 10 Do
    ReadLn (a[i]);
  For i := 1 To 10 Do
    For j := 1 To 10 Do
      ReadLn (b[i,j]);
    soma1 := 0;
    soma2 := 0;
  For i := 1 To 10 Do
    For j := 1 To 10 Do
      Begin
        soma1 := soma1 + a[i];
        soma2 := soma2 + b[i,j];
      End;
    WriteLn (soma1,soma2);
  End.
```

```
{R+}
Program Programa2;
Type vet = Array [1..10] Of Integer;
      mat = Array[1..10,1..10] Of Integer;
Var a: vet;
      b: mat;
      i, j, soma1, soma2: Integer;
Begin
  For i := 1 To 10 Do
    ReadLn (a[i]);
  For i := 1 To 10 Do
    For j := 1 To 10 Do
      ReadLn (b[i,j]);
    soma1 := 0;
    soma2 := 0;
  For i := 1 To 10 Do
    Begin
      soma1 := soma1 + a[i];
      For j := 1 To 10 Do
        soma2 := soma2 + b[i,j];
```

```

End;
WriteLn (soma1,soma2);
End.

```

Responda:

- (a) O que executa cada programa?
- (b) Os dois programas fornecem as mesmas respostas?
- (c) No programa Programa1, quantas vezes são executados os comandos:

```

soma1 := soma1 + a[i];
soma2 := soma2 + b[i,j]

```

- (d) No programa Programa2, quantas vezes são executados os comandos:

```

soma1 := soma1 + a[i];
soma2 := soma2 + b[i,j]

```

- (e) Qual o programa mais eficiente?

39. Indique qual será a saída impressa do programa abaixo:

```

{$R+}
Var mat: Array [1..3,1..3] Of Integer;
    i, j, inteiro, x, y, xant, yant, n: Integer;
Begin
    inteiro := 1;
    n := 3;
    FillChar (mat,SizeOf(mat),0);
    x := (n + 1) / 2;
    y := n;
    While inteiro < 10 Do
    Begin
        mat[x,y] := inteiro;
        xant := x;
        yant := y;
        Inc (inteiro);
        Inc (x);
        Inc (y);
        If x > n Then
            x := 1;
        If y > n Then
            y := 1;
        If a[x,y] = 0 Then
            x := xant
        Else
            y := yant - 1;
    End;
    For i := 1 To n Do
        For j := 1 To n Do

```



```
WriteLn ('mat[' ,i ,',',j ,'] = ',mat[i,j]);
```

End.

40. Dados $M \in R^{n \times n}$, escreva um programa, com reproprocessamento, que leia os elementos da matriz e a escreva após ter multiplicado os elementos da diagonal principal por uma constante k .
41. Escreva um programa, com reproprocessamento, que leia uma tabela de $m \times n$ elementos, calcule e imprima a soma de cada linha e a soma de todos os elementos.
42. Escreva um programa, com reproprocessamento, que calcule e imprima a soma de todos os elementos de uma matriz de ordem $m \times n$.
43. Faça um programa, com reproprocessamento, que leia duas matrizes reais de dimensão $m \times n$, calcule e imprima a soma destas matrizes.
44. Escreva um programa, com reproprocessamento, que efetue um produto matricial. Seja $A \in R^{m \times n}$ e $B \in R^{n \times p}$, calcular $C \in R^{m \times p}$, onde $C = A * B$ e $m \leq 40, n \leq 60, p \leq 80$. O programa deve imprimir as matrizes A e B e, a matriz C calculada.
45. Faça um programa que gere e imprima a seguinte matriz de ordem 6:

1	1	1	1	1	1
1	2	2	2	2	1
1	2	3	3	2	1
1	2	3	3	2	1
1	2	2	2	2	1
1	1	1	1	1	1

46. Suponha que a ordem da matriz do exercício anterior seja $2 * n$ (n ímpar). Escreva um programa, com reproprocessamento, que gere a matriz de ordem $2 * n$. Exemplos:

N = 1	N = 2	N = 3	N = 4																																																													
<table><tr><td>1</td></tr></table>	1	<table><tr><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td></tr></table>	1	1	1	1	<table><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>2</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	1	2	1	1	1	1	<table><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>2</td><td>2</td><td>1</td></tr><tr><td>1</td><td>2</td><td>2</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	1	1	2	2	1	1	2	2	1	1	1	1	1																															
1																																																																
1	1																																																															
1	1																																																															
1	1	1																																																														
1	2	1																																																														
1	1	1																																																														
1	1	1	1																																																													
1	2	2	1																																																													
1	2	2	1																																																													
1	1	1	1																																																													
N = 5	N = 6																																																															
<table><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>2</td><td>2</td><td>2</td><td>1</td></tr><tr><td>1</td><td>2</td><td>3</td><td>2</td><td>1</td></tr><tr><td>1</td><td>2</td><td>3</td><td>2</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	1	1	1	2	2	2	1	1	2	3	2	1	1	2	3	2	1	1	1	1	1	1	<table><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr><tr><td>1</td><td>2</td><td>2</td><td>2</td><td>2</td><td>1</td></tr><tr><td>1</td><td>2</td><td>3</td><td>3</td><td>2</td><td>1</td></tr><tr><td>1</td><td>2</td><td>3</td><td>3</td><td>2</td><td>1</td></tr><tr><td>1</td><td>2</td><td>2</td><td>2</td><td>2</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr></table>	1	1	1	1	1	1	1	2	2	2	2	1	1	2	3	3	2	1	1	2	3	3	2	1	1	2	2	2	2	1	1	1	1	1	1	1		
1	1	1	1	1																																																												
1	2	2	2	1																																																												
1	2	3	2	1																																																												
1	2	3	2	1																																																												
1	1	1	1	1																																																												
1	1	1	1	1	1																																																											
1	2	2	2	2	1																																																											
1	2	3	3	2	1																																																											
1	2	3	3	2	1																																																											
1	2	2	2	2	1																																																											
1	1	1	1	1	1																																																											

47. Escreva um programa, com reproprocessamento, que leia uma matriz de ordem n de elementos inteiros, calcule e imprima a soma dos elementos situados abaixo da diagonal principal da matriz, incluindo os elementos da própria diagonal principal.

48. Escreva um programa, com reprocessamento, que leia uma matriz A de no máximo 20×50 elementos, determine a matriz transposta de A e imprima a matriz A e sua transposta.
49. Faça um programa, com reprocessamento, que leia uma matriz $A \in R^{m \times n}$, com $m \leq 20$ e $n \leq 30$. O programa deve imprimir a matriz A lida, calcular e imprimir uma matriz modificada $B \in R^{m \times (n+1)}$, sendo que os elementos da $(n+1)$ -ésima colunas são formados com o produto dos elementos da mesma linha. Exemplo:

MATRIZ A

2	1	4
3	4	5

MATRIZ B

2	1	4	8
3	4	5	60

50. Uma biblioteca possui oito departamento. Cada departamento contém 40 estantes capazes de conter, cada uma, 150 livros. Supondo que o livro padrão tenha 200 páginas de 35 linhas por 60 colunas de caracteres, declare uma variável capaz de conter todos os caracteres presentes nos livros da biblioteca. Escreva a quantidade de memória ocupada por esta variável.
51. Um grupo de pessoas respondeu a um questionário composto de 10 perguntas. Cada pergunta contém cinco opções ou respostas possíveis, codificadas de 1 a 5. Cada pergunta é respondida com a escolha de apenas uma opção entre as cinco possíveis. São fornecidos os nomes das pessoas e suas respectivas respostas. Escreva um programa, com reprocessamento, que leia e imprima os dados lidos e, calcule e imprima o número de pessoas que responderam a cada uma das cinco opções de cada pergunta.
52. Escreva um programa, com reprocessamento, que leia uma matriz quadrada real A, de dimensão $m \times m$, e verifique se a matriz é simétrica, ou seja, se $a_{ij} = a_{ji}$ para todo $i, j \leq m$.
53. Escreva um programa para imprimir uma tabela com o índice de afinidade existente entre cada moça e cada rapaz de um grupo de M moças e um grupo de R rapazes ($R \leq 30$ e $M \leq 20$). Foi distribuído entre eles um questionário de 20 perguntas, tais como:

01. Você se incomoda que seu parceiro fume?
02. Você é vidrado em música sertaneja?
03. Você gosta de cebola?
-
20. Você gosta do Esporte Clube Noroeste?

Cada resposta tem as seguintes opções: sim, não e indiferente.

54. Considere o seguinte sistema de equações:

59. Suponha que exista 6 cidades ligadas por ferrovia, da seguinte forma:

- a cidade 1 se comunica com as cidades 2 e 3;
- a cidade 2 se comunica com as cidades 1, 3, 4 e 5;
- a cidade 3 se comunica com as cidades 1, 2, 4 e 5;
- a cidade 4 se comunica com as cidades 2, 3 e 6;
- a cidade 5 se comunica com as cidades 2, 3 e 6;
- a cidade 6 se comunica com as cidades 4 e 5;

A distância entre as cidades está disposta na matriz a seguir:

	1	2	3	4	5	6
1	-	100	15	-	-	-
2	100	-	40	180	200	-
3	15	40	-	45	35	-
4	-	180	45	-	-	105
5	-	200	95	-	-	120
6	-	-	-	105	120	-

Escreva um programa, com reprocessamento, que leia duas cidades (origem e destino), encontre o menor caminho entre elas e escreva este caminho.

60. Faça um programa, com reprocessamento, que leia os palpites de um jogador na loteria esportiva e calcule o valor a ser pago pelo apostador. O programa deverá ler e imprimir os dados referentes ao teste e ao revendedor, como especificado abaixo.

O valor a ser pago é calculado por: $VALOR = V * 2 * 3$, onde V é uma constante a ser lida, D o número de palpites duplos e T o número de palpites triplos.

A estrutura de dados a ser manipulada pelo programa deve ser uma matriz bidimensional de 3 linhas por 13 colunas.

61. Faça um programa que calcule a soma dos elementos da *i-ésima* linha de uma matriz real de ordem $m \times n$.

62. Faça um programa que calcule o produto dos elementos da *i-ésima* coluna de uma matriz real de ordem $m \times n$.

63. Escreva um procedimento que recebe uma matriz de caracteres 8x8 representando um tabuleiro de xadrez e calcula o valor total das peças do jogo. Espaços vazios do tabuleiro são codificados como casas com ' ' (branco) e tem valor 0 (zero). Os valores das demais peças são dados de acordo com a tabela:

Peça	Peão	cavalo	bispo	torre	rainha	rei
Valor	1	3	3	5	10	50

64. Dada uma matriz real A de ordem $m \times n$, e, um vetor real v de ordem n , escreva um programa, com reprocessamento, que determine o produto de A por v .
65. Um vetor real x com n elementos é apresentado como resultado de um sistema de equações lineares $A.x = b$, cujos coeficientes estão armazenados em uma matriz real A e, os termos independentes em um vetor real b de m elementos. Faça um programa, com reprocessamento, que leia a matriz A e os vetores b e x e, verifique se x é solução do sistema $A.x = b$.
66. Uma matriz $A \in R^{n \times n}$ é um quadrado latino de ordem n se em cada linha e em cada coluna aparecem todos os inteiros $1, 2, 3, \dots, n$ (ou seja, cada linha e coluna é permutação dos inteiros $1, 2, 3, \dots, n$). Escreva um programa que verifique se uma dada matriz inteira A de ordem $n \times n$ é um quadrado latino de ordem n .

Exemplo:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \\ 4 & 1 & 2 & 3 \\ 3 & 4 & 1 & 2 \end{pmatrix} \text{ é um quadrado latino de ordem 4.}$$

67. Dadas n datas em uma matriz $DATA_{n \times 3}$, onde a primeira coluna corresponde ao dia, a segunda ao mês e a terceira ao ano, escreva um programa que coloque essas datas em ordem cronológica crescente.

Exemplo: $n = 6$

$$DATA = \begin{pmatrix} 25 & 6 & 1965 \\ 16 & 6 & 1965 \\ 13 & 12 & 1941 \\ 21 & 4 & 1965 \\ 6 & 2 & 1989 \\ 1 & 10 & 1973 \end{pmatrix} \text{ terá como saída } \begin{pmatrix} 13 & 12 & 1941 \\ 21 & 4 & 1965 \\ 16 & 6 & 1965 \\ 25 & 6 & 1965 \\ 1 & 10 & 1973 \\ 6 & 2 & 1989 \end{pmatrix}.$$

68. Dada uma matriz $A \in R^{m \times n}$, escreva um programa, com reprocessamento, que verifique se existem elementos repetidos em A .
69. Uma matriz inteira A é uma matriz de permutação se em cada linha e em cada coluna houver $n-1$ elementos nulos e um único elemento igual a 1.

Exemplos:

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ é uma matriz de permutação.}$$

$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ não é uma matriz de permutação.}$$

Escreva um programa, com reprocessamento, que verifique se $A \in Z^{n \times n}$ é de permutação.

70. Um determinado cinema possui capacidade de 100 lugares (máximo). Certo dia cada espectador respondeu a um questionário no qual constava:
- sua idade;
 - sua opinião em relação ao filme, segundo: ótimo = *****; bom = ****; regular = ***, ruim = **; péssimo = *.

Elabore um programa que, lendo estes dados, calcule e imprima:

- (a) a quantidade de respostas ótimas;
 - (b) a diferença percentual entre respostas bom e regular;
 - (c) a média de idade das pessoas que responderam ruins;
 - (d) a percentagem de respostas péssimo e a maior idade que utilizou esta opção;
 - (e) a diferença de idade entre a maior que respondeu ótimo e a maior idade de quem respondeu ruim.
71. Uma matriz $D_{8 \times 8}$ pode representar a posição atual de um jogo de damas, sendo que 0 indica uma casa vazia, 1 indica uma casa ocupada por uma peça branca e -1 indica uma casa ocupada por uma peça preta. Supondo que as peças pretas estão se movendo no sentido crescente das linhas da matriz D, escreva um programa que determine as posições das peças pretas que:
- (a) podem tomar peças brancas;
 - (b) podem mover-se sem tomar peças;
 - (c) não podem se mover.
72. Uma matriz quadrada inteira é um quadrado mágico se a soma dos elementos de cada linha, a soma dos elementos da cada coluna e a soma dos elementos das diagonais principal e secundária são todas iguais.

Exemplo:

$$\begin{pmatrix} 8 & 0 & 7 \\ 4 & 5 & 6 \\ 3 & 10 & 2 \end{pmatrix} \text{ é um quadrado mágico.}$$

Escreva um programa, com reprocessamento, que verifique se uma matriz quadrada é um quadrado mágico.

73. (SPOJ CHUVA11 - 10868) Bob trabalha no OBM (Órgão Brasileiro de Meteorologia), que é a organização responsável pela medição dos índices pluviométricos (quantidade de chuva acumulada) em todo o país. Eles são muito eficientes no que fazem, mas estão com um problema: eles não sabem como proceder para calcular a quantidade acumulada de chuva que caiu em cada região em dois períodos consecutivos, muito embora eles saibam os dados de cada período separadamente.

Como a chefia do Órgão estava muito ocupada, acabou ficando a cargo de Bob, o estagiário, a tarefa de implementar um programa que some, para cada região, a quantidade de chuva acumulada em dois períodos consecutivos.

O mapa que o OBM usa é dividido em NN regiões, sendo que para cada região, a cada período, é determinado um número inteiro indicando a quantidade de chuva acumulada. A quantidade de chuva acumulada total em cada região em dois períodos consecutivos é a soma das quantidades de chuva em cada um dos períodos.

Mas como Bob é só um estagiário e não está acostumado a fazer nada mais do que tirar cópias de documentos, ele pediu sua ajuda para implementar o programa que calcula a quantidade de chuva acumulada total nos dois períodos para cada uma das regiões, dadas as quantidades de chuva acumulada em cada período para cada região.

Entrada

A primeira linha da entrada contém um inteiro N indicando a dimensão dos dois mapas que devem ser lidos. Nas próximas $2N$ linhas são dados os dois mapas, cada mapa indicando a quantidade de chuva acumulada nas regiões em um período. Cada mapa é descrito em N linhas consecutivas, cada linha contendo N inteiros, sendo que cada inteiro indica a quantidade de chuva acumulada, no período, em uma região.

Saída

A saída deverá conter N linhas, com N inteiros em cada linha, indicando a quantidade de chuva acumulada total em cada uma das regiões nos dois períodos considerados.

Restrições

- $1 \leq N \leq 100$.
- $0 \leq \text{quantidade de chuva acumulada em cada região de cada mapa} \leq 100$.

Exemplos

Entrada

2
1 2
3 4
10 11
12 13

Saída
11 13
15 17

Entrada
3
1 1 1
1 2 2
1 2 3
3 2 1
2 2 1
1 1 1

Saída
4 3 2
3 4 3
2 3 4

74. (SPOJ COFRE - 840. Cofrinhos da Vó Vitória) Vó Vitória mantém, desde o nascimento dos netos Joãozinho e Zezinho, um ritual que faz a alegria dos meninos. Ela guarda todas as moedas recebidas como troco em dois pequenos cofrinhos, um para cada neto. Quando um dos cofrinhos fica cheio, ela chama os dois netos para um alegre almoço, ao qual entrega aos garotos as moedas guardadas nos cofrinhos de cada um.

Ela sempre foi muito zelosa quanto à distribuição igualitária do troco arrecadado. Quando, por força do valor das moedas, ela não consegue depositar a mesma quantia nos dois cofrinhos, ela memoriza a diferença de forma a compensá-la no próximo depósito.

Tarefa

Vó Vitória está ficando velha e tem medo que deslizos de memória a façam cometer injustiças com os netos, deixando de compensar as diferenças entre os cofrinhos. Sua tarefa é ajudar Vó Vitória, escrevendo um programa de computador que indique as diferenças entre os depósitos, de forma que ela não tenha que preocupar-se em memorizá-las.

Entrada

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de teste contém um número inteiro N , que indica o número de depósitos nos cofrinhos. As N linhas seguintes descrevem cada uma um depósito nos cofrinhos; o depósito é indicado por dois valores inteiros J e Z , separados por um espaço em branco, representando respectivamente os valores, em centavos, depositados nos cofres de Joãozinho e Zezinho. O final da entrada é indicado por $N = 0$.

Saída

Para cada conjunto de teste da entrada seu programa deve produzir um conjunto de linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste n”, onde n é numerado seqüencialmente a partir de 1. A seguir seu programa deve escrever uma linha para cada depósito do conjunto de testes. Cada linha deve conter um inteiro que representa a diferença (em centavos) entre o valor depositado nos cofrinhos do Joãozinho e do Zezinho. Deixe uma linha em branco ao final de cada conjunto de teste. A grã mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo

Entrada:

```
3
20 25
10 5
10 10
4
0 5
12 0
0 20
17 1
0
```

Saída:

Teste 1

```
-5
0
0
```

Teste 2

```
-5
7
-13
3
```

Restrições

- $0 \leq N \leq 100$ ($N = 0$ apenas para indicar o fim da entrada)
- $0 \leq J \leq 100$ (valor de cada depósito no cofre de Joãozinho)
- $0 \leq Z \leq 100$ (valor de cada depósito no cofre de Zezinho)

75. (SPOJ MAGICO11 - 11013. Quadrado mágico) Arnaldo e Bernardo são dois garotos que compartilham um peculiar gosto por curiosidades matemáticas. Nos últimos tempos, sua principal diversão tem sido investigar propriedades matemáticas de tabuleiros quadrados

preenchidos com inteiros. Recentemente, durante uma aula de matemática, os dois desafiaram os outros alunos da classe a criar quadrados mágicos, que são quadrados preenchidos com números de 1 a N^2 , de tal forma que a soma dos N números em uma linha, coluna ou diagonal principal do quadrado tenham sempre o mesmo valor. A ordem de um quadrado mágico é o seu número de linhas, e o valor do quadrado mágico é o resultado da soma de uma linha. Um exemplo de quadrado mágico de ordem 3 e valor 15 é mostrado na figura abaixo:

2	7	6
9	5	1
4	3	8

Para surpresa de Arnaldo e Bernardo, os outros alunos criaram um grande número de quadrados, alguns enormes, e alegaram que todos eram quadrados mágicos. Arnaldo e Bernardo agora precisam de sua ajuda, para verificar se os quadrados criados são realmente mágicos.

Você deve escrever um programa que, dado um quadrado, verifique se ele é realmente mágico.

Entrada

A primeira linha da entrada contém um único número inteiro N , indicando a ordem do quadrado (seu número de linhas). As N linhas seguintes descrevem o quadrado. Cada uma dessas linhas contém N números inteiros separados por um espaço em branco.

Saída

Seu programa deve imprimir uma única linha. Caso o quadrado seja mágico, a linha deve conter o valor do quadrado (ou seja, a soma de uma de suas linhas). Caso contrário, a linha deve conter o número 0.

Restrições

- $3 \leq N \leq 1000$.
- $1 \leq \text{valor de cada célula} \leq 10^9$.

Exemplos

Entrada

```

3
1 1 1
1 1 1
1 1 1

```

```

Saída
0

```

```

Entrada
4
16 3 2 13
5 10 11 8
9 6 7 12
4 15 14 1

```

```

Saída
34

```

```

Entrada
3
4 8 9
11 7 3
6 5 10

```

```

Saída
0

```

76. (SPOJ MINHOCA - 2607. Campo de Minhocas) Minhocas são muito importantes para a agricultura e como insumo para produção de ração animal. A Organização para Bioengenharia de Minhocas (OBM) é uma entidade não governamental que promove o aumento da produção, utilização e exportação de minhocas.

Uma das atividades promovidas pela OBM é a manutenção de uma fazenda experimental para pesquisa de novas tecnologias de criação de minhocas. Na fazenda, a área destinada às pesquisas é de formato retangular, dividida em células quadrangulares de mesmo tamanho. As células são utilizadas para testar os efeitos, na produção de minhocas, de variações de espécies de minhocas, tipos de terra, de adubo, de tratamento, etc. Os pesquisadores da OBM mantêm um acompanhamento constante do desenvolvimento das minhocas em cada célula, e têm uma estimativa extremamente precisa da produtividade em cada uma das células. A figura abaixo mostra um mapa da fazenda, mostrando a produtividade estimada de cada uma das células.

81	28	240	10
40	10	100	240
20	180	110	35

Um pesquisador da OBM inventou e construiu uma máquina colhedeira de minhocas, e quer testá-la na fazenda. A máquina tem a largura de uma célula, e em uma passada pelo terreno de uma célula colhe todas as minhocas dessa célula, separando-as, limpando-as e empacotando-as. Ou seja, a máquina eliminara uma das etapas mais intensivas de mão de obra no processo de produção de minhocas. A máquina, porém, ainda está em desenvolvimento e tem uma restrição: não faz curvas, podendo movimentar-se somente em linha reta.

Decidiu-se então que seria efetuado um teste com a máquina, de forma a colher o maior número possível de minhocas em uma única passada, em linha reta, de lado a lado do campo de minhocas. Ou seja, a máquina deve colher todas as minhocas de uma ‘coluna’ ou de uma ‘linha’ de células do campo de minhocas (a linha ou coluna cuja soma das produtividades esperadas das células é a maior possível).

Tarefa

Escreva um programa que, fornecido o mapa do campo de minhocas, descrevendo a produtividade estimada em cada célula, calcule o número esperado total de minhocas a serem colhidas pela máquina durante o teste, conforme descrito acima.

Entrada

A primeira linha da entrada contém dois números inteiros N e M , representando respectivamente o número de linhas ($1 \leq N \leq 100$) e o número de colunas ($1 \leq M \leq 100$) de células existentes no campo experimental de minhocas. Cada uma das N linhas seguintes contém M inteiros, representando as produtividades estimadas das células correspondentes a uma linha do campo de minhocas.

Saída

A saída deve ser composta por uma única linha contendo um inteiro, indicando o número esperado total de minhocas a serem colhidas pela máquina durante o teste.

Exemplo 1

Entrada:

```
3 4
81 28 240 10
```

40 10 100 240
20 180 110 35

Saída:
450

Exemplo 2

Entrada:
4 1
100
110
0
100

Saída:
310

Restrições

- $1 \leq N \leq 100$
- $1 \leq M \leq 100$
- $0 \leq \text{Produtividade de uma célula} \leq 500$
- $0 \leq \text{Produtividade de uma linha ou coluna de células} \leq 50000$

77. (SOPJ PARPROX - 3827. Pontos) Escreva um programa que, dados vários pontos no plano, imprima a distância do par de pontos mais próximos.

Entrada

A entrada contém um único caso de teste. A primeira linha contém um número inteiro positivo N , que indica o número de pontos a serem considerados. As N linhas seguintes contém dois números inteiros cada, representando as coordenadas X e Y de cada ponto.

Saída

Seu programa deve imprimir uma única linha, contendo um número que corresponde a distância do par de pontos mais próximos, com precisão na terceira casa decimal.

Restrições

- $0 \leq N \leq 1000$

- $-2000000 \leq X, Y \leq 2000000$

Exemplo

Entrada

2
0 0
2 2

Saída

2.828

Entrada

5
100 -80
-9000 100
-200 100
500 -915
214 5

Saída

142.201

78. (SPOJ PEDAGIO - 819. Pedágio) Como prêmio pela primeira colocação na Olimpíada Brasileira de Informática, Juquinha e sua família ganharam uma viagem de uma semana à Coréia do Sul. Como o país é deslumbrante, com tradições, cultura, arquitetura e culinária muito diferentes das do Brasil, o pai de Juquinha, o Sr. Juca, decidiu alugar um carro para conhecer melhor o país. As estradas são muito bem cuidadas; todas são de sentido duplo, e duas cidades podem ser ligadas diretamente por mais de uma estrada. No entanto, em todas as estradas paga-se um pedágio de valor fixo (há um pedágio em cada direção, entre duas cidades). Como o Sr. Juca não tem muito dinheiro para gastar, as viagens com o carro devem ser muito bem planejadas.

Tarefa

Escreva um programa que, conhecidas as cidades e estradas existentes no país, e a cidade onde Juquinha e sua família estão, encontre cada cidade (que não a cidade onde eles estão) que possa ser visitada por eles, dada a restrição de que o Sr. Juca deseja pagar no máximo P pedágios (considerando apenas a viagem de ida).

Entrada

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de teste contém quatro números inteiros C , E , L e P . Os valores C e E indicam respectivamente o número de cidades e o número de estradas existentes. As cidades são identificadas por inteiros de 1 a C . Os valores L e P indicam, respectivamente, a cidade onde a família de Juquinha está no momento e o número máximo de pedágios que o Sr. Juca está disposto a pagar. As E linhas seguintes contêm cada uma a informação de uma estrada, representada por um par de números inteiros positivos X e Y , indicando que há uma estrada (de sentido

duplo) da cidade X para a cidade Y. O final da entrada é indicado por $C = E = L = P = 0$.

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste n”, onde n é numerado a partir de 1. Na segunda linha devem aparecer os identificadores das cidades que podem ser alcançadas, em ordem crescente, separados por pelo menos um espaço em branco. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo

Entrada:

```
5 4 2 1
1 2
2 3
3 4
4 5
9 12 1 2
2 1
1 5
2 1
3 2
9 3
3 4
4 8
4 7
7 6
5 6
4 5
3 7
0 0 0 0
```

Output:

```
Teste 1
1 3
```

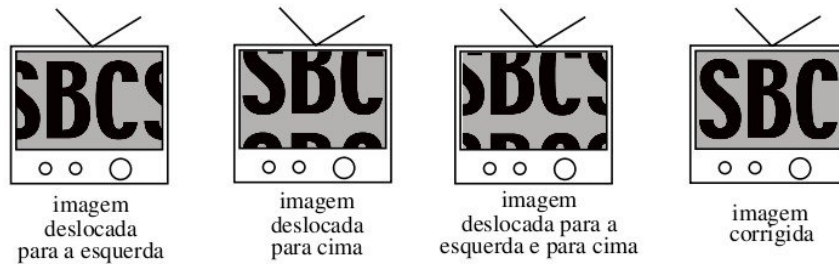
```
Teste 2
2 3 4 5 6
```

Restrições

- $0 \leq C \leq 50$ ($C = 0$ apenas para indicar o fim da entrada)
- $0 \leq E \leq 2500$ ($E = 0$ apenas para indicar o fim da entrada)
- $0 \leq L \leq C$ ($L = 0$ apenas para indicar o fim da entrada)

- $0 \leq P \leq C$ ($P = 0$ apenas para indicar o fim da entrada)
- $1 \leq X \leq C$
- $1 \leq Y \leq C$

79. (SPOJ TV - 1366. TV da Vovó) A vovó tem um televisor muito antigo, que ultimamente está exibindo um defeito incômodo: a imagem aparece 'deslocada' (para cima ou para baixo, para o lado direito ou para o lado esquerdo). Quando a imagem está deslocada para cima, a parte da imagem que deixa de ser vista na parte superior reaparece na parte de baixo da tela. Da mesma forma, quando a imagem está deslocada a direita, a parte da imagem que deixa de ser vista à direita reaparece na tela do lado esquerdo.



A imagem do televisor pode ser vista como uma matriz de pontos organizados em linhas e colunas. Para consertar o televisor da vovó, você pode ajustar a imagem introduzindo uma série de 'comandos de correção' em um painel de ajuste. Cada comando de correção desloca a imagem de um certo número de linhas (para cima ou para baixo) e um certo número de colunas (para a direita ou para a esquerda).

Tarefa

Dada uma matriz que representa uma imagem defeituosa e uma série de comandos de correção, seu programa deve calcular a matriz que representa a imagem resultante após todos os comandos terem sido aplicados sequencialmente.

Entrada

A entrada possui vários conjuntos de teste. Cada conjunto de teste inicia com a descrição da matriz que representa a imagem do televisor. A primeira linha contém dois inteiros M e N representando o número de linhas e o número de colunas da matriz ($1 \leq M \leq 1000$ e $1 \leq N \leq 1000$). As M linhas seguintes da entrada contém cada uma N inteiros, descrevendo o valor de cada ponto da imagem. Após a descrição da imagem, segue-se a descrição dos comandos de correção. Cada comando de correção é descrito em uma linha contendo dois inteiros X e Y . O valor de X representa o deslocamento na direção horizontal (valor positivo representa deslocamento para a direita, valor negativo para a esquerda), e o valor de Y representa o deslocamento da direção vertical (valor positivo para cima, valor negativo para baixo). O final da lista de comandos é indicado por $X = Y = 0$, e o final da entrada é indicado por $M = N = 0$.

Exemplo de Entrada

```
3 3
1 2 3
4 5 6
7 8 9
1 0
1 -1
0 0
3 4
6 7 8 5
10 11 12 9
2 3 4 1
-3 2
0 0
0 0
```

Saída

Para cada conjunto de teste, o seu programa deve produzir uma imagem na saída. A primeira linha da saída deve conter um identificador do conjunto de teste, no formato "Teste n", onde n é numerado seqüencialmente a partir de 1. A seguir deve aparecer a matriz que representa a imagem resultante, no mesmo formato da imagem de entrada. Ou seja, as N linhas seguintes devem conter cada uma M inteiros que representam os pixels da imagem. Após a imagem deixe uma linha em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Saída

Teste 1

```
8 9 7
```

```
2 3 1
```

```
5 6 4
```

Teste 2

```
1 2 3 4
```

```
5 6 7 8
```

```
9 10 11 12
```

(esta saída corresponde ao exemplo de entrada acima)

Restrições

- $0 \leq N \leq 1000$ ($N = 0$ apenas para indicar o final da entrada)
- $0 \leq M \leq 1000$ ($M = 0$ apenas para indicar o final da entrada)
- $0 \leq X \leq 1000$
- $0 \leq Y \leq 1000$
- $0 \leq \text{número de comandos de correção em cada conjunto de teste} \leq 1000$

5 Estruturas de Repetição

1. Ler números naturais e imprimir o dobro de cada número lido até que o usuário digite -1 .
2. Calcular a soma dos 100 primeiros números inteiros.
3. Calcular a soma dos 100 números a partir de um número determinado. Por exemplo, para 125: $125 + 126 + \dots + 224$.
4. Calcular o fatorial de um número.
5. Escreva um programa que imprime todos os números primos entre 2 e n , onde n é fornecido pelo usuário.
6. Escreva um programa que solicite um número inteiro de até 5 dígitos ao usuário e inverta a ordem de seus algarismos. Por exemplo, para a entrada 12345, a saída deverá ser 54321.
7. Escreva um programa que calcule o n -ésimo número de Fibonacci, n informado pelo usuário. A série de Fibonacci é calculada da seguinte forma:

$$Fibonacci(n) = \begin{cases} 0 & \text{caso } n \leq 0 \\ 1 & \text{caso } n = 1 \text{ ou } n = 2 \\ Fibonacci(n-1) + Fibonacci(n-2) & \text{caso contrário} \end{cases}$$

8. Fazer um programa para calcular o valor da seguinte somatória:

$$S = \frac{2^1}{50} + \frac{2^2}{49} + \frac{2^3}{48} + \dots + \frac{2^{50}}{1}$$

9. Fazer um programa que calcule o faturamento de uma sessão de cinema. O preço normal do ingresso deve ser lido no início do programa. Pagam meia entrada os menores de 18 e os maiores de 65 anos. O limite de uma sala é de 200 pessoas.
10. Faça uma função que retorne x^y , sendo dados x e y inteiros como parâmetros de entrada. Use apenas a biblioteca `<stdio.h>`.
11. Faça uma função que retorne o fatorial de um número inteiro e positivo, sendo este número dado como parâmetro de entrada.
12. Utilizando a função do exercício anterior, faça um programa que calcule e imprima o valor do coeficiente binomial, dado pela expressão a seguir:

$$\binom{a}{b} = \frac{n!}{k!(n-k)!}$$

13. Dado um inteiro positivo N , faça uma rotina que retorne o valor da seguinte soma:

$$S = \frac{1}{N} + \frac{2}{N-1} + \frac{3}{N-2} + \dots + \frac{N-1}{2} + \frac{N}{1}$$

14. Escreva uma função que calcule o valor de e^x através da série

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

sendo dados por parâmetros x e o número de termos.

15. Um matemático italiano da idade média conseguiu modelar o ritmo de crescimento da população de coelhos através de uma seqüência de números naturais que passou a ser conhecida como seqüência de Fibonacci. O n -ésimo número da seqüência de Fibonacci F é dado pela seguinte fórmula de recorrência:

$$\begin{cases} F_1 = 1 \\ F_2 = 1 \\ F_n = F_{n-1} + F_{n-2}, \text{ para } n \geq 3 \end{cases}$$

Escreva uma função que dado n calcule F_n . Faça um programa, com reprocessamento, que utilize esta função.

16. Pelo calendário gregoriano, instituído em 1582 pelo Papa Gregório XIII, os anos bissextos são aqueles cujo ano são divisíveis por 4, exceto os anos que são divisíveis por 100 e não por 400. Por exemplo, os anos 1600, 2000 são anos bissextos enquanto que os anos 1700, 1800 e 1900 não são. Sabendo disso, faça um programa que imprima na tela todos os anos bissextos de 1600 até 2012.
17. Um armazém trabalha com 40 mercadorias diferentes identificadas pelos números inteiros de 1 a 40. O dono do armazém anota a quantidade de cada mercadoria vendida durante o mês. Ele tem uma tabela que indica para cada mercadoria o preço de venda. Escreva um programa para calcular o faturamento mensal do armazém, isto é:

$$\text{faturamento} = \sum_{i=1}^{40} (\text{quantidade}_i * \text{preco}_i)$$

As tabelas de preço e de quantidade são armazenadas em dois vetores distintos, sendo que um conjunto contém a quantidade vendida e o outro o preço de cada mercadoria.

18. Considere

$$p(x) = a_0 + a_1.x^1 + a_2.x^2 + \dots + a_n.x^n$$

Escreva um programa que leia o valor de n ($n \leq 20$) e os coeficientes do polinômio $p(x)$, calcule o valor de $p(x)$ para 10 valores de x e, imprima o valor de x e o valor de $p(x)$ correspondente.

19. Escreva um programa para gerar a série de Fibonacci com $n \leq 25$ termos e imprima conforme o modelo a seguir (no exemplo, $n = 7$).

```

1
1 1
1 1 2
1 1 2 3
1 1 2 3 5
1 1 2 3 5 8
1 1 2 3 5 8 13
1 1 2 3 5 8
1 1 2 3 5
1 1 2 3
1 1 2
1 1
1

```

20. Dados N e uma seqüência de N números inteiros, escreva um programa que determine quantos segmentos de números iguais consecutivos compõem essa seqüência.

Exemplo: A seqüência 5, 2, 2, 3, 6, 6, 6, 6, 6, 9, 9, 9, 9 é formada por 5 segmentos de números iguais.

21. Dados um inteiro positivo N e uma seqüência de N números inteiros, escreva um programa, com reprocessamento, que determine o comprimento de um segmento crescente de comprimento máximo.

Exemplos:

(a) Na seqüência 5, 10, 3, 2, 5, 7, 8, 4, 2 o comprimento do segmento crescente máximo é 4.

(b) Na seqüência 9, 5, 4, 3, 1 o comprimento de um segmento crescente máximo é 1.

22. Dados dois números naturais M e N e duas seqüências ordenadas com M e N números inteiros, escreva um programa que mostre uma única seqüência ordenada contendo todos os elementos das seqüências originais sem repetição.

23. Dado o polinômio $p(x) = a_0 + a_1.x^1 + a_2.x^2 + \dots + a_n.x^n$, escreva um programa que calcule o polinômio $q(x)$ tal que $p(x) = (x - \alpha).q(x) + p(\alpha)$ (Utilize o método de Briot-Ruffini). O programa deve calcular o polinômio $q(x)$ para M valores distintos de α .

24. Dados os polinômios reais

$$p(x) = a_0 + a_1.x^1 + a_2.x^2 + \dots + a_n.x^n \text{ e } q(x) = b_0 + b_1.x^1 + b_2.x^2 + \dots + b_n.x^n$$

Escreva um programa que determine o produto de 5 pares desses polinômios.

25. Dada a sequência x_1, x_2, \dots, x_k de números inteiros, escreva um programa que determine o segmento de soma máxima. O programa deve determinar o segmento de soma máxima de N sequências diferentes.

Exemplo: Na sequência 5, 2, -2, -7, 3, 14, 10, -3, 9, -6, 4, a soma do segmento é 33.

26. Chama-se sequência de Farey relativa a n, a sequência das frações racionais irredutíveis, dispostas em ordem crescente, com denominadores positivos e não maiores que n.

Exemplo: Se $n = 5$, os termos α da sequência de Farey, tais que $0 \leq \alpha \leq 1$ são:

$$\frac{0}{1}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \frac{1}{1}$$

Para geração dos termos α de uma sequência de Farey tais que $0 \leq \alpha \leq 1$, pode-se usar o seguinte processo. Começa-se com as frações $\frac{0}{1}$ e $\frac{1}{1}$, e entre cada duas frações consecutivas $\frac{i}{j}$ e $\frac{k}{m}$, se introduz a fração $\frac{i+k}{j+m}$ e assim sucessivamente enquanto $j+m \leq n$. Quando não for mais possível introduzir novas frações tem-se gerado todos os termos α da sequência de Farey relativa a n, tais que $0 \leq \gamma \leq 1$.

Usando o processo descrito, escreva um programa que determine os termos α , $0 \leq \alpha \leq 1$, da sequência de Farey relativa a n, n inteiro positivo. O programa deve determinar a sequência de Farey de M números distintos.

Sugestão: Gere os numeradores e os denominadores em dois vetores.

27. (a) Escreva um programa que imprima as N primeiras linhas do triângulo de Pascal.

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1

```

- (b) Escreva um programa que imprima as N primeiras linhas do triângulo de Pascal usando apenas um vetor.

28. Escreva um programa que receba um inteiro positivo n e imprima o valor do *n-ésimo* termo da sequência de Fibonacci F.

29. Um número natural é triangular se ele é produto de três números naturais consecutivos. Faça uma rotina que dado N natural, verifique se N é triangular.

Exemplo: 120 é triangular, pois $4 * 5 * 6 = 120$.

30. Um número N inteiro positivo é perfeito se for igual a soma de seus divisores positivos diferentes de N . Escreva uma rotina que verifique se um dado número inteiro positivo é perfeito. Escreva um programa, com reprocessamento, que utilize esta rotina.

Exemplo: 6 é perfeito, pois $1 + 2 + 3 = 6$.

31. Anos bissextos são aqueles cujo ano são divisíveis por 4, exceto os anos que são divisíveis por 100 e não por 400. Sabendo disso, faça um programa que imprima na tela todos os anos bissextos de 2012 até 3012.
32. Faça um programa que leia um inteiro e positivo N , calcule e mostre o valor de E , conforme a fórmula a seguir.

$$E = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{N!}$$

33. Faça um programa que leia o número de termos e um valor positivo para X , calcule e mostre o valor da série a seguir.

$$S = \frac{-X^2}{1!} + \frac{X^3}{2!} + \frac{-X^4}{3!} + \frac{X^5}{4!} + \frac{X^6}{5!} + \frac{X^7}{6!} + \dots$$

34. Se listarmos todos os números naturais menores que 10 que são múltiplos de 3 ou 5 obteremos 3, 5, 6 e 9. A soma desses números é igual a 23. Encontre a soma de todos os múltiplos de 3 ou 5 menores que 1000.
35. Faça um programa para calcular e imprimir um termo qualquer das sequências:

(a) $S(1) = 10, S(n) = S(n - 1) + 10$ para $n \geq 2$;

(b) $B(1) = 1, B(n) = B(n - 1) + n^2$ para $n \geq 2$;

(c) $T(1) = 1, T(n) = T(n - 1) + 3$ para $n \geq 2$.

36. Cada novo termo da sequência de Fibonacci é gerado pela adição dos dois termos anteriores. Se iniciarmos com 1 e 2, os primeiros 10 termos seriam: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89. Considerando os termos da sequência de Fibonacci cujos valores não excedem 4 milhões, encontre a soma de todos os termos pares.
37. 2520 é o menor número que pode ser dividido por qualquer um dos números de 1 a 10 sem que sobre nenhum resto. Encontre o menor número positivo com essa propriedade que seja divisível por qualquer valor de 1 a 20.
38. Os seis primeiros números primos são: 2, 3, 5, 7, 11 e 13. Neste caso, 13 é o 6º número primo. Qual é o 10.001º primo?

39. Os fatores primos de 13195 são 5, 7, 13 e 29. Qual é o maior fator primo do número 600851475143?
40. A famosa conjectura de Goldbach diz que todo inteiro par maior que 2 é a soma de dois números primos. Testes extensivos foram feitos sem contudo ser encontrado um contra-exemplo. Escreva um programa mostrando que a afirmação é verdadeira para todo número par entre 200 e 1500. O programa deve imprimir cada número e seus correspondentes primos. Utilize a função definida nos exercícios anteriores.
41. Faça um algoritmo que leia o nome e as 4 notas bimestrais de 5 alunos, calculando e apresentando a média de cada um.
42. Faça um algoritmo que leia 30 números inteiros quaisquer. Ao final o algoritmo deve mostrar a quantidade de números inteiros positivos, negativos e zeros digitados.
43. Faça um algoritmo que leia um número inteiro positivo N, some todos os números inteiros entre 1 e N, e mostre o resultado obtido.
44. Considerando os candidatos A e B, faça um algoritmo que aceite os votos de 10 pessoas e apresente o candidato vencedor e a quantidade de votos que ele recebeu.
45. Faça um algoritmo que leia 20 caracteres. Ao final o algoritmo deve mostrar a quantidade lida de cada vogal e a quantidade de vogais lidas.
46. Para computar o vencedor de uma eleição deve ser feito um algoritmo. Há 3 candidatos, e os votos dos eleitores foram codificados da seguinte forma:
- 1, 2 ou 3: votos para os respectivos candidatos
 - 0: voto em branco
 - Qualquer valor maior que 3: voto nulo
- Escrever um algoritmo que leia o voto de 50 eleitores e forneça o número do vencedor da eleição (suponha que não pode haver empates) e as quantidades de votos brancos e nulos.
47. Faça um algoritmo que some a idade de 40 pessoas.
48. Imprimir todas as tabuadas de multiplicar de 1 até 10.
49. Faça um algoritmo que leia um número inteiro positivo N, calcule e apresente o valor de H, onde

$$H = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots \frac{1}{N}$$

50. Faça um algoritmo que leia um número inteiro entre 1 e 9 e mostre a tabuada de multiplicação do mesmo.
51. Faça um algoritmo que leia um número N e escreva na tela todos os números pares de 0 a N e todos os números ímpares de N a 0.
52. Faça um algoritmo que leia um número inteiro N e mostre todos os números primos entre 0 e N.
53. Faça um algoritmo que leia um número inicial e um número final e mostre na tela todos os números inteiros entre eles. Se o número final for menor que o inicial, o algoritmo deve apresentar uma mensagem de erro.
54. Faça um algoritmo que leia um número e divida-o por dois (sucessivamente) até que o resultado seja menor que 1. Mostre o resultado da última divisão efetuada.
55. Faça um algoritmo que leia uma lista de números terminada pelo número 9999 e mostre cada número lido. Ao final, o algoritmo deve mostrar a média aritmética de todos os números da lista.
56. Faça um algoritmo que leia um número e, em seguida, leia uma lista de números até achar um igual ao primeiro número lido. Mostre todos os números lidos.
57. Faça um algoritmo que leia N e uma lista de N números e mostre a soma de todos os números da lista.
58. Faça um algoritmo que leia um número N e uma lista de N números inteiros positivos e mostre o maior número da lista.
59. Faça um algoritmo que leia uma lista de letras terminada pela letra “Z”. Ao final o algoritmo deve mostrar a quantidade lida de cada vogal
60. Faça um algoritmo que leia nomes de cidades e suas respectivas populações. O processamento deve parar quando o nome da cidade for “fim”. Ao final, deve ser mostrado: quantidade de cidades digitadas, quantidade de vezes que foi digitada a cidade “Campos”, a maior população informada.
61. Entrar com 20 números e imprimir a soma dos positivos e o total de números negativos.
62. Criar um algoritmo que leia um número da entrada e imprima a multiplicação por 3 e por 5 dos números no intervalo de 1 ao número. Exemplo:
Número lido: 2 Saída: para 1, $1*3 = 3$, $1*5 = 5$ para 2, $2*3 = 6$, $2*5=10$

63. A série RICCI difere da série de FIBONACCI porque os dois primeiros termos são fornecidos pelo usuário. Os demais termos são gerados da mesma forma que a série FIBONACCI. Criar um algoritmo que imprima os N=10 primeiros termos da série de RICCI e a soma dos termos impressos, sabendo-se que para existir esta série serão necessários pelo menos três termos. Os termos da sequência de Fibonacci são construídos como no exemplo (a partir do primeiro): 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...
64. Criar um algoritmo que calcule o produto de dois números lidos, através de somas sucessivas. Suponha que os números lidos sejam positivos. Por exemplo, $3 * 4 = 4 + 4 + 4$
65. A série de FETUCCINE é gerada da seguinte forma: os dois primeiros termos são fornecidos pelo usuário; a partir daí, os termos são gerados com a soma ou subtração dos dois termos anteriores, ou seja:

$$A_i = A_{i-1} + A_{i-2} \text{ para } i \text{ ímpar}$$
$$A_i = A_{i-1} - A_{i-2} \text{ para } i \text{ par}$$

Desenvolva um algoritmo que imprima os 10 primeiros termos da série de FETUCCINE.

66. Uma empresa de telefonia deseja calcular as contas telefônicas de seus assinantes através do computador. A cobrança de seus serviços é feita da seguinte maneira:

(a) Tarifa básica

- telefone residencial (código 1): R\$ 23,00
- telefone comercial (código 2): R\$ 30,00

(b) Serviço local

- R\$ 0,10 por pulso excedente (acima de 90 pulsos)

(c) Serviço despertador

- R\$ 0,47 por vez

Na entrada de dados teremos:

- código do assinante;
- tipo de telefone (comercial ou residencial);
- Número de pulsos registrados para chamadas locais e número de serviços de despertador prestados.

Desenvolva um algoritmo que leia os dados de um conjunto de assinantes (o código do assinante igual a zero encerra a entrada de dados), calcule e imprima:

- para cada assinante, o total de sua conta;
- valor da maior conta e o código do assinante que a pagou;
- valor médio arrecadado por assinante no mês.

67. Faça um algoritmo que some a idade de várias pessoas. Para finalizar a entrada de dados, entre com uma idade igual a zero.

68. Faça um algoritmo que peça para digitar S,C,V,D, F. Quando a opção F for detectada o algoritmo deve mostrar o total de solteiros, casados, divorciados e viúvos que passaram pelo sistema.

69. Faça um algoritmo que leia um número indeterminado de notas (entre 0 e 10) e ao final mostre a média, a maior nota, a menor nota e o número de alunos. Quando a nota digitada for -1 deve-se encerrar a entrada de dados. Valores fora da faixa apesar de aceitos não serão computados para cálculo.

70. Faça um algoritmo que repetidamente mostra na tela duas opções: "1 - Fim" e "2 - Calcular raiz" e lê do teclado a opção desejada pelo usuário. Se a opção for 1, o algoritmo termina. Se a opção for 2, o algoritmo lê um número real e, se o número for positivo ou zero, calcula e mostra sua raiz quadrada, se negativo, mostra uma mensagem de erro. E se a opção for inválida (nem 1 nem 2), é mostrada uma mensagem apropriada. Quando a opção não é a de terminar, o algoritmo volta para mostrar novamente as opções e ler a opção do usuário.

71. Solicitar nome, preço unitário e quantidade de todos os produtos de uma loja. Para cada produto, mostrar seu subtotal. Ao final, deve-se mostrar a quantidade de produtos processados e o total geral. A condição de parada será uma pergunta feita para cada produto: CONTINUA? S/N.

72. Uma escola tem 5 turmas e cada turma tem N alunos. Desenvolva um algoritmo que imprima, por turma, o total de alunos com média superior a 7 e a média geral da escola.

73. Crie um algoritmo que informe a quantidade total de calorias de uma refeição a partir da escolha do usuário que deverá informar o prato, a sobremesa e bebida (veja a tabela a seguir). Suponha que o algoritmo é desenvolvido para 50 usuários.

PRATO	SOBREMESA	BEBIDA
Vegetariano 180 cal	Abacaxi 75 cal	Chá 20 cal
Peixe 230 cal	Sorvete diet 110 cal	Suco de laranja 70 cal
Frango 250 cal	Mousse diet 170 cal	Suco de melão 100 cal
Carne 350 cal	Mousse chocolate 200 cal	Refrigerante diet 65 cal

74. Desenvolva um algoritmo que a partir da idade e peso de 10 pacientes calcule a dosagem de determinado medicamento e imprima a receita informando quantas gotas do medicamento o paciente deve tomar por dose. Considere que o medicamento em questão possui 500mg/mL, e que cada mL corresponde a 20 gotas.

- Adultos e adolescentes desde 12 anos, inclusive, se tiverem peso igual ou acima de 60 quilos devem tomar 1000 mg; com peso abaixo de 60 quilos devem tomar 875 mg.
- Para crianças e adolescentes abaixo de 12 anos a dosagem é calculada pelo peso corpóreo conforme a tabela a seguir:

5 Kg a 9 kg	125 mg
9.1 Kg a 16 Kg	250 mg
16.1 Kg a 24 Kg	375 mg
24.1 Kg a 30 Kg	500 mg
Acima de 30 Kg	750 mg

75. Sabe-se que um número da forma n é igual a soma de n ímpares consecutivos.

Exemplo: $13 = 1$, $23 = 3 + 5$, $33 = 7 + 9 + 11$, $43 = 13 + 15 + 17 + 19$, ...

Dado M , determine os ímpares consecutivos cuja soma é igual a n^3 para n assumindo valor de 1 a M . Utilize o conceito de rotina.

76. Dado um número natural na base binária, escreva uma rotina que o transforme para a base decimal. Faça um programa, com reprocessamento, que utilize esta rotina.

77. Dado um número natural na base decimal, escreva uma rotina que o transforme para a base binária. Faça um programa, com reprocessamento, que utilize esta rotina.

78. Qualquer número natural de quatro algarismos pode ser dividido em duas dezenas formadas pelos seus dois primeiros e dois últimos dígitos.

Exemplo: 1278: 12 e 78.

Escreva um programa que imprima todos os milhares (4 algarismos) cuja raiz quadrada seja a soma das dezenas formadas pela divisão acima. Utilize o conceito de rotina.

Exemplo: A raiz de $9801 = 99 = 98 + 01$.

Portanto, 9801 é um dos números a ser impresso.

79. Dado N e dois números naturais i e j diferentes de 0, escreva uma rotina que imprima em ordem crescente os N primeiros naturais que são múltiplos de i ou de j ou de ambos. Faça um programa, com reprocessamento, que utilize esta rotina.

Exemplo: Para $N = 6$, $i = 2$ e $j = 3$ a saída deverá ser: 0,2,3,4,6,8

80. Um número i é congruente módulo m a j se $i \text{ Mod } m = j \text{ Mod } m$. Dados N , j e m naturais não nulos, imprima os N primeiros naturais congruentes a j módulo m . Utilize o conceito de rotina.
81. Dados três números naturais, escreva uma rotina que verifique se eles formam os lados de um triângulo retângulo. Faça um programa, com reprocessamento, que utilize esta rotina.
82. Escreva uma rotina que, dado um número, retorne seu valor por extenso. Faça um programa, com reprocessamento, que utilize esta rotina.
83. Dados uma frase e uma palavra, escreva uma rotina que verifique o número de vezes que a palavra ocorre na frase. Faça um programa, com reprocessamento, que utilize esta rotina.

Exemplo: Na frase ANA E MARIANA GOSTAM DE BANANA, a palavra ANA ocorre 4 vezes.

84. Escreva uma rotina que calcule a distância entre dois pontos. Escreva também, o modo de chamada desta rotina.
85. Dado um número inteiro positivo N , escreva uma rotina que imprima todos os inteiros entre 1 e N que são comprimentos da hipotenusa de um triângulo retângulo com catetos inteiros. Faça um programa, com reprocessamento, que utilize esta rotina.
86. São dados dois números inteiros positivos p e q , sendo que o número de dígitos de p é menor ou igual ao número de dígitos de q . Faça uma rotina que verifique se p é um subnúmero de q . Escreva um programa, com reprocessamento, que utilize esta rotina.

Exemplos:

(a) $p = 23$, $q = 452367$, p é subnúmero de q .

(b) $p = 23$, $q = 452167$, p não é subnúmero de q .

87. Faça um programa, utilizando o conceito de rotina, para resolver o seguinte problema:

São dadas as coordenadas reais x e y de um ponto, um número natural N , e as coordenadas reais de N pontos ($1 \leq N \leq 100$). Deseja-se calcular e imprimir, sem repetição, os raios das circunferências centradas no ponto (x,y) que passam por pelo menos um dos N pontos dados.

Exemplo: $(x, y) = (1, 1)$; $N = 5$ pontos: $(-1, 1.2), (1.5, 2), (0, -2), (0, 0.5), (4, 2)$
Neste caso há três circunferências de raios: 1.12, 2.01e3.162.

Observações:

(a) Distância entre os pontos (a, b) e (c, d) é $\sqrt{(a - c)^2 + (b - d)^2}$.

(b) Dois pontos estão na mesma circunferência se estão à mesma distância do centro.

88. (SPOJ ALADES - 5474. Alarme Despertador) Daniela é enfermeira em um grande hospital, e tem os horários de trabalho muito variáveis. Para piorar, ela tem sono pesado, e uma grande dificuldade para acordar com relógios despertadores.

Recentemente ela ganhou de presente um relógio digital, com alarme com vários tons, e tem esperança que isso resolva o seu problema. No entanto, ela anda muito cansada e quer aproveitar cada momento de descanso. Por isso, carrega seu relógio digital despertador para todos os lugares, e sempre que tem um tempo de descanso procura dormir, programando o alarme despertador para a hora em que tem que acordar. No entanto, com tanta ansiedade para dormir, acaba tendo dificuldades para adormecer e aproveitar o descanso.

Um problema que a tem atormentado na hora de dormir é saber quantos minutos ela teria de sono se adormecesse imediatamente e acordasse somente quando o despertador tocasse. Mas ela realmente não é muito boa com números, e pediu sua ajuda para escrever um programa que, dada a hora corrente e a hora do alarme, determine o número de minutos que ela poderia dormir.

Entrada

A entrada contém vários casos de teste. Cada caso de teste é descrito em uma linha, contendo quatro números inteiros $H1$, $M1$, $H2$ e $M2$, com $H1:M1$ representando a hora e minuto atuais, e $H2:M2$ representando a hora e minuto para os quais o alarme despertador foi programado ($0 \leq H1 \leq 23, 0 \leq M1 \leq 59, 0 \leq H2 \leq 23, 0 \leq M2 \leq 59$).

O final da entrada é indicado por uma linha que contém apenas quatro zeros, separados por espaços em branco.

Os dados devem ser lidos da entrada padrão.

Saída

Para cada caso de teste da entrada seu programa deve imprimir uma linha, cada uma contendo um número inteiro, indicando o número de minutos que Daniela tem para dormir.

O resultado de seu programa deve ser escrito na saída padrão.

Exemplo

```
Entrada:
1 5 3 5
23 59 0 34
21 33 21 10
0 0 0 0
```

```
Saída:
120
35
1417
```

89. (SPOJ BAFO - 1386. Bafo) Álbuns de figurinhas – sejam de times de futebol, princesas ou super-heróis – têm marcado gerações de crianças e adolescentes. Conseguir completar um álbum é uma tarefa muitas vezes árdua, envolvendo negociações com colegas para a troca de figurinhas. Mas a existência das figurinhas propicia uma outra brincadeira, que foi muito popular entre crianças no século passado: o jogo de bater figurinhas (o famoso “Bafo”). O jogo é muito simples, mas divertido (e muito competitivo). No início de uma partida, cada criança coloca em uma pilha um certo número de figurinhas. Uma partida é composta de rodadas; a cada rodada as crianças batem com a mão sobre a pilha de figurinhas, tentando virá-las com o vácuo formado pelo movimento da mão. As crianças jogam em turnos, até que a pilha de figurinhas esteja vazia. Ganha a partida a criança que conseguir virar mais figurinhas.

Aldo e Beto estão jogando bafo com todas as suas figurinhas e pediram sua ajuda para calcular quem é o vencedor.

Tarefa

Você deve escrever um programa que, dada a quantidade de figurinhas que Aldo e Beto viraram em cada rodada, determine qual dos dois é o vencedor.

Entrada

A entrada é composta de vários casos de teste, cada um correspondendo a uma partida entre Aldo e Beto. A primeira linha de um caso de teste contém um número inteiro R que indica quantas rodadas ocorreram na partida. Cada uma das R linhas seguintes contém dois inteiros, A e B , que correspondem, respectivamente, ao número de figurinhas que Aldo e Beto conseguiram virar naquela rodada. Em todos os casos de teste há um único vencedor (ou seja, não ocorre empate). O final da entrada é indicado por $R = 0$.

Saída

Para cada caso de teste da entrada, seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do caso de teste, no formato “Teste n ”, onde n é numerado seqüencialmente a partir de 1. A segunda linha deve conter o nome do vencedor (Aldo ou Beto). A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo

Entrada:

```
2
1 5
2 3
3
0 0
4 7
10 0
0
```

Saída:

```
Teste 1
Beto
```

```
Teste 2
Aldo
```

Restrições

- $1 \leq R \leq 1000$ ($R = 0$ apenas para indicar o final da entrada)
- $0 \leq A \leq 100$
- $0 \leq B \leq 100$

90. (SPOJ BIT - 812. Bits Trocados) As Ilhas Weblands formam um reino independente nos mares do Pacífico. Como é um reino recente, a sociedade é muito influenciada pela informática. A moeda oficial é o Bit; existem notas de B\$ 50,00, B\$10,00, B\$5,00 e B\$1,00. Você foi contratado(a) para ajudar na programação dos caixas automáticos de um grande banco das Ilhas Weblands.

Tarefa

Os caixas eletrônicos das Ilhas Weblands operam com todos os tipos de notas disponíveis, mantendo um estoque de cédulas para cada valor (B\$ 50,00, B\$10,00, B\$5,00 e B\$1,00). Os clientes do banco utilizam os caixas eletrônicos para efetuar retiradas de um certo número inteiro de Bits.

Sua tarefa é escrever um programa que, dado o valor de Bits desejado pelo cliente, determine o número de cada uma das notas necessário para totalizar esse valor, de modo a minimizar a quantidade de cédulas entregues. Por exemplo, se o cliente deseja retirar B\$50,00, basta entregar uma única nota de cinquenta Bits. Se o cliente deseja retirar B\$72,00, é necessário entregar uma nota de B\$50,00, duas de B\$10,00 e duas de B\$1,00.

Entrada

A entrada é composta de vários conjuntos de teste. Cada conjunto de teste é composto por uma única linha, que contém um número inteiro positivo V , que indica o valor solicitado pelo

cliente. O final da entrada é indicado por $V = 0$.

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste n”, onde n é numerado a partir de 1. Na segunda linha devem aparecer quatro inteiros I, J, K e L que representam o resultado encontrado pelo seu programa: I indica o número de cédulas de B\$50,00, J indica o número de cédulas de B\$10,00, K indica o número de cédulas de B\$5,00 e L indica o número de cédulas de B\$1,00. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo

Entrada:

1
72
0

Saída:

Teste 1
0 0 0 1

Teste 2
1 2 0 2

Restrições

- $0 \leq V \leq 10000$ ($V = 0$ apenas para indicar o fim da entrada)

91. (SPOJ CALCU11 - 11005. Calculadora) Solicitando Boas Contas (SBC) é uma organização de inspeção de calculadoras. Todos os fabricantes procuram ter o selo de qualidade da SBC, que faz com que os clientes comprem o produto sem preocupação com contas erradas.

Você está encarregado de testar máquinas que fazem apenas operações de multiplicação e divisão. Além disso, o termo a ser digitado em cada operação (que dividirá ou multiplicará o número atualmente exibido no visor) só pode conter um único dígito.

A calculadora exibe o número 1 quando ligada. Depois disso, o usuário pode digitar um número com um único dígito e escolher se esse número deve multiplicar ou dividir o número exibido anteriormente; o resultado da operação escolhida é então exibido na calculadora. Pode-se repetir esse processo indefinidamente.

Apesar de só podermos entrar com números inteiros de um dígito, o visor da calculadora permite exibir números com múltiplos dígitos e até mesmo números fracionários.

Dada uma sequência de operações que foram realizadas nessa calculadora logo depois de ligada, sua tarefa é conferir o resultado exibido.

Entrada

A primeira e única linha da entrada contém um inteiro N . Cada uma das próximas N linhas contém um dígito e um caractere '*' ou '/', que representam uma operação realizada na calculadora.

Saída

Seu programa deve imprimir uma única linha contendo o resultado que deve ser exibido pela calculadora ao final das operações.

Restrições

- $1 \leq N \leq 100000$.
- Os números informados são inteiros entre 1 e 9.
- O resultado final da conta é um número inteiro entre 1 e 2^{30} .

Exemplos

Entrada

```
3
2 *
1 *
3 *
```

Saída

```
6
```

O usuário deseja calcular o resultado da seguinte expressão: $1/2 * 1 * 3$. Note que a primeira ocorrência do número 1 vem do fato da calculadora mostrar inicialmente 1 ao invés de 0.

Entrada

```
3
2 /
3 /
6 *
```

Saída

```
1
```

Neste exemplo, o usuário deseja calcular o resultado da seguinte expressão: $((1/2)/3) * 6$.

Entrada

11

9 *

9 *

9 *

9 *

9 *

9 *

9 *

9 *

9 *

9 *

9 /

Saída

387420489

92. (SPOJ VIVO - 1388. Vivo ou Morto) Toda criança certamente já brincou de “vivo ou morto”. A brincadeira é dirigida por um “chefe” (um adulto), que comanda dois ou mais participantes (crianças). A brincadeira é composta de rodadas. No início, os participantes são organizados pelo chefe em fila única. A cada rodada o chefe grita “vivo” ou “morto” e todos os participantes tentam seguir sua ordem, levantando-se ao ouvir a palavra “vivo” ou abaixando-se ao ouvir a palavra “morto”. Um participante que não segue a ordem do chefe é eliminado, deixando o seu lugar na fila. Os participantes remanescentes agrupam-se novamente em fila única, preenchendo as posições dos participantes eliminados, mas mantendo suas posições relativas. O jogo continua até que uma rodada seja composta por exatamente um participante. Tal participante é dito o vencedor do jogo.

Por exemplo, considere que a brincadeira inicie com cinco participantes, identificados por números inteiros de 1 a 5, e que o chefe organize a fila na ordem $3 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 5$. Se na primeira rodada forem eliminados os participantes 2 e 4, a fila da segunda rodada será formada por $3 \rightarrow 1 \rightarrow 5$; se na segunda rodada for eliminado o participante 1, a fila da terceira rodada será formada por $3 \rightarrow 5$. Se na terceira rodada o participante 3 for eliminado, o vencedor da brincadeira será o participante 5.

Tarefa

Sua tarefa é escrever um programa que determine o vencedor de uma partida de “vivo ou morto”, a partir da informação das ordens dadas pelo chefe e das ações executadas pelos participantes em cada rodada.

Entrada

A entrada é constituída de vários casos de teste, cada um representando uma partida. A primeira linha de um caso de teste contém dois números inteiros P e R indicando respectivamente a quantidade inicial de participantes ($2 \leq P \leq 100$) e quantidade de rodadas da partida ($1 \leq R \leq 100$). Os participantes são identificados por números de 1 a P . A segunda linha de um caso de teste descreve a fila organizada pelo chefe, contendo P números inteiros distintos x_1, x_2, \dots, x_P , onde x_1 representa o identificador do participante no primeiro lugar na fila, x_2 representa o identificador do participante no segundo lugar na fila, e assim por diante ($1 \leq x_i \leq P$). Cada uma das R linhas seguintes representa uma rodada, contendo um número

inteiro inteiro N indicando o número de participantes da rodada ($2 \leq N \leq P$), um número inteiro inteiro J representando a ordem dada pelo chefe ($0 \leq J \leq 1$) e N números inteiros A_i representando a ação do participante colocado na i -ésima posição na fila ($0 \leq A_i \leq 1$). Ordens e ações “vivo” são representadas pelo valor 1, ordens e ações “morto” pelo valor zero. Cada partida tem exatamente um vencedor, determinado somente na última rodada fornecida no caso de teste correspondente. O final da entrada é indicado por $P = R = 0$.

Saída

Para cada caso de teste seu programa deve produzir três linhas. A primeira identifica o conjunto de teste no formato “Teste n ”, onde n é numerado a partir de 1. A segunda linha deve conter o identificador do vencedor. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo

Entrada:

```
2 2
2 1
2 1 1 1
2 1 1 0
5 4
3 2 1 4 5
5 1 1 1 1 1 1
5 0 0 1 0 1 0
3 0 0 1 0
2 1 0 1
0 0
```

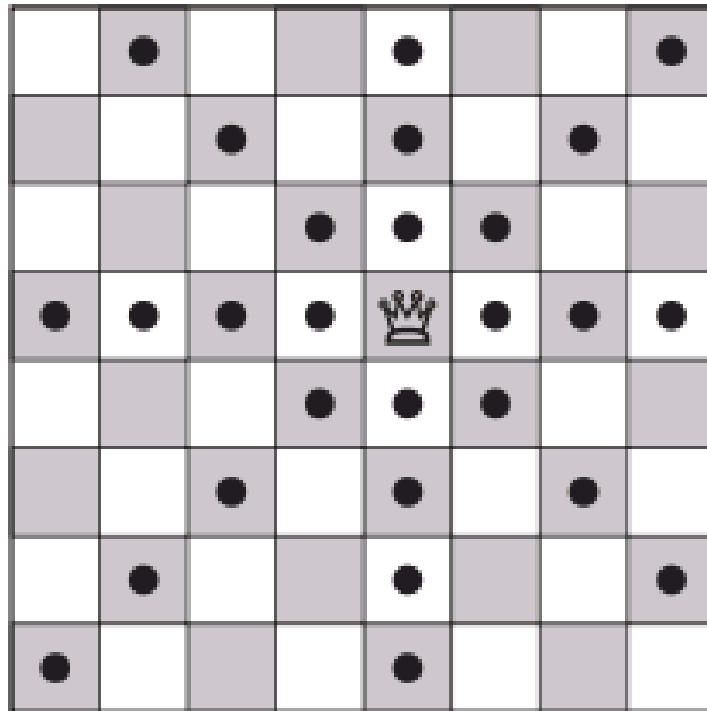
Saída:

```
Teste 1
2
Teste 2
5
```

Restrições

- $2 \leq P \leq 100$ ($P = 0$ apenas para indicar o fim da entrada)
- $1 \leq R \leq 100$ ($R = 0$ apenas para indicar o fim da entrada)
- $1 \leq x_i \leq P$, para $1 \leq i \leq P$
- $2 \leq N \leq P$
- $0 \leq J \leq 1$
- $0 \leq A_i \leq 1$, para $1 \leq i \leq N$

93. (SPOJ DAMA - 3240. Dama) O jogo de xadrez possui várias peças com movimentos curiosos: uma delas é a dama, que pode se mover qualquer quantidade de casas na mesma linha, na mesma coluna, ou em uma das duas diagonais, conforme exemplifica a figura abaixo:



O grande mestre de xadrez Kary Gasparov inventou um novo tipo de problema de xadrez: dada a posição de uma dama em um tabuleiro de xadrez vazio (ou seja, um tabuleiro 8 8, com 64 casas), de quantos movimentos, no mínimo, ela precisa para chegar em outra casa do tabuleiro?

Kary achou a solução para alguns desses problemas, mas teve dificuldade com outros, e por isso pediu que você escrevesse um programa que resolve esse tipo de problema.

Entrada

A entrada contém vários casos de teste. A primeira e única linha de cada caso de teste contém quatro inteiros $X1$, $Y1$, $X2$ e $Y2$ ($1 \leq X1, Y1, X2, Y2 \leq 8$). A dama começa na casa de coordenadas $(X1, Y1)$, e a casa de destino é a casa de coordenadas $(X2, Y2)$. No tabuleiro, as colunas são numeradas da esquerda para a direita de 1 a 8 e as linhas de cima para baixo também de 1 a 8. As coordenadas de uma casa na linha X e coluna Y são (X, Y) .

O final da entrada é indicado por uma linha contendo quatro zeros.

Saída

Para cada caso de teste da entrada seu programa deve imprimir uma única linha na saída, contendo um número inteiro, indicando o menor número de movimentos necessários para a dama

chegar em sua casa de destino.

Exemplo

Entrada

4 4 6 2

3 5 3 5

5 5 4 3

0 0 0 0

Saída

1

0

2

94. (SPOJ DECADARI - 1820. O Relógio Decadário) Nos primeiros anos após a Revolução Francesa, intelectuais foram encarregados de erradicar da sociedade certas tradições e superstições da era negra da realeza. Algumas destas contribuições alcançaram um amplo sucesso, tais como o sistema métrico. Outras caíram em (quase) completo esquecimento, tais como o sistema de tempo decimal, inventado pelo matemático Gilbert Romme. O sistema de tempo decimal divide o dia em 10 horas decimais, que são divididas em 100 minutos decimais, que por sua vez são divididos em 100 segundos decimais.

Você foi encarregado pela mundialmente famosa fabricante de relógios "Splatch" para incluir mais uma característica inútil na próxima linha de produtos: um mostrador de tempo decimal. Sua primeira tarefa é implementar um programa que converte um tempo no formato tradicional para um tempo decimal com precisão de um centésimo de segundo.

Entrada

A entrada consiste de uma sequência de linhas, cada uma contendo exatamente um tempo tradicional, no formato HHMMSSCC, onde $0 \leq HH \leq 23$, $0 \leq MM \leq 59$, $0 \leq SS \leq 59$ e $0 \leq CC \leq 99$. A entrada é terminada com um fim de arquivo (EOF).

Saída

Para cada tempo tradicional, a saída deverá mostrar uma linha com o horário decimal correspondente, arredondado por truncamento, no formato HMMSSCC, onde $0 \leq H \leq 9$, $0 \leq MM \leq 99$, $0 \leq SS \leq 99$ e $0 \leq CC \leq 99$.

Exemplo

Entrada:

00000000

23595999

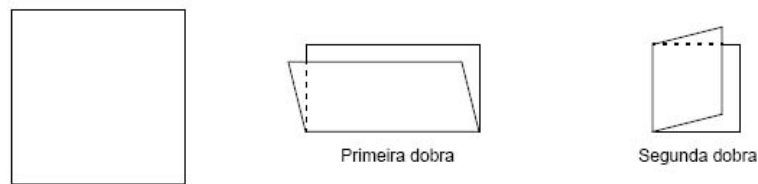
12000000

14273467

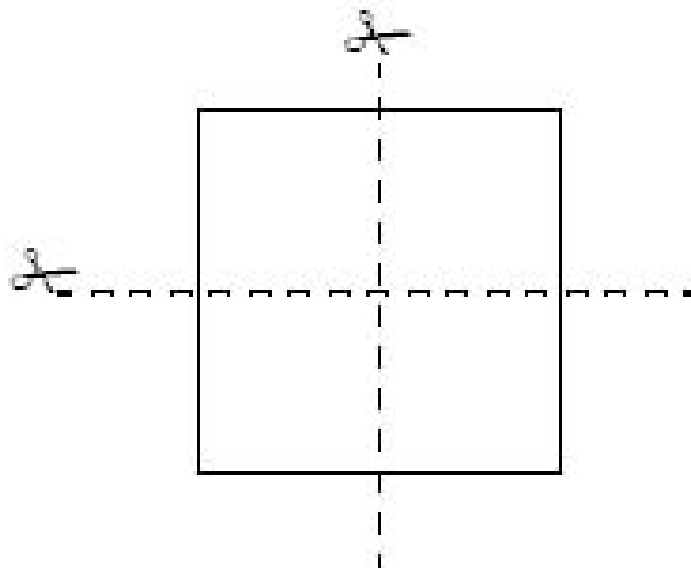
02475901

Saída:
0000000
9999998
5000000
6024846
1166552

95. (SPOJ DOBRA - 817. Dobradura) Zezinho tem aulas de Iniciação Artística em sua escola, e recentemente aprendeu a fazer dobraduras em papel. Ele ficou fascinado com as inúmeras possibilidades de se dobrar uma simples folha de papel. Como Zezinho gosta muito de matemática, resolveu inventar um quebra-cabeça envolvendo dobraduras. Zezinho definiu uma operação de dobradura D que consiste em dobrar duas vezes uma folha de papel quadrada de forma a conseguir um quadrado com $1/4$ do tamanho original, conforme ilustrado na figura.



Depois de repetir N vezes esta operação de dobradura D sobre o papel, Zezinho cortou o quadrado resultante com um corte vertical e um corte horizontal, conforme a figura abaixo.



Zezinho lançou então um desafio aos seus colegas: quem adivinha quantos pedaços de papel foram produzidos?

Entrada

A entrada é composta de vários conjuntos de teste. Cada conjunto de teste é composto de uma única linha, contendo um número inteiro N que indica o número de vezes que a operação de dobradura D foi aplicada. O final da entrada é indicado por $N \leq -1$.

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste n ”, onde n é numerado a partir de 1. A segunda linha deve conter o número de pedaços de papel obtidos depois de cortar a dobradura, calculado pelo seu programa. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo

Entrada:

1
0
-1

Saída:

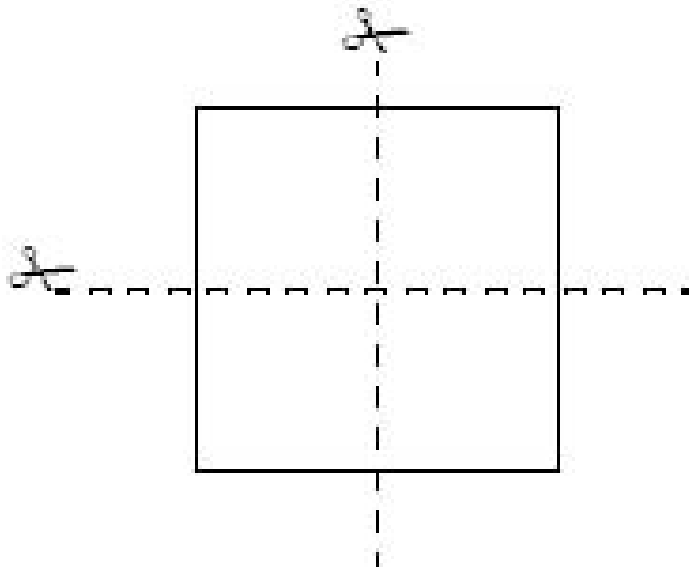
Teste 1
9

Teste 2
4

Restrições

- $-1 \leq N \leq 15$ ($N \leq -1$ apenas para indicar o fim da entrada)
96. (SPOJ ELEVADOR - 8302. Elevador) A FCC (Fábrica de Cilindros de Carbono) fabrica vários tipos de cilindros de carbono. A FCC está instalada no décimo andar de um prédio, e utiliza os vários elevadores do prédio para transportar os cilindros. Por questão de segurança, os cilindros devem ser transportados na posição vertical; como são pesados, no máximo dois cilindros podem ser transportados em uma única viagem de elevador. Os elevadores têm formato de paralelepípedo e sempre têm altura maior que a altura dos cilindros.

Para minimizar o número de viagens de elevador para transportar os cilindros, a FCC quer, sempre que possível, colocar dois cilindros no elevador. A figura abaixo ilustra, esquematicamente (vista superior), um caso em que isto é possível (a), e um caso em que isto não é possível (b):



Como existe uma quantidade muito grande de elevadores e de tipos de cilindros, a FCC quer que você escreva um programa que, dadas as dimensões do elevador e dos dois cilindros, determine se é possível colocar os dois cilindros no elevador.

Entrada

A entrada contém vários casos de teste. A primeira e única linha de cada caso de teste contém quatro números inteiros L , C , $R1$ e $R2$, separados por espaços em branco, indicando respectivamente a largura do elevador ($1 \leq L \leq 100$), o comprimento do elevador ($1 \leq C \leq 100$), e os raios dos cilindros ($1 \leq R1, R2 \leq 100$).

O último caso de teste é seguido por uma linha que contém quatro zeros separados por espaços em branco

Saída

Para cada caso de teste, o seu programa deve imprimir uma única linha com um único caractere: 'S' se for possível colocar os dois cilindros no elevador e 'N' caso contrário.

Exemplo

```
Entrada:
11 9 2 3
7 8 3 2
10 15 3 7
8 9 3 2
0 0 0 0

Saída:
S
N
```


97. (SPOJ ELEVADOR2 - 8778. Elevador) A Subindo Bem Confortavelmente (SBC) é uma empresa tradicional, com mais de 50 anos de experiência na fabricação de elevadores. Todos os projetos da SBC seguem as mais estritas normas de segurança, mas infelizmente uma série de acidentes com seus elevadores manchou a reputação da empresa.

Ao estudar os acidentes, os engenheiros da companhia concluíram que, em vários casos, o acidente foi causado pelo excesso de passageiros no elevador. Por isso, a SBC decidiu fiscalizar com mais rigor o uso de seus elevadores: foi instalado um sensor em cada porta que detecta a quantidade de pessoas que saem e entram em cada andar do elevador.

A SBC tem os registros do sensor de todo um dia de funcionamento do elevador (que sempre começa vazio). Eles sabem que as pessoas são educadas e sempre deixam todos os passageiros que irão sair em um andar saírem antes de outros passageiros entrarem no elevador, mas ainda assim eles têm tido dificuldade em decidir se a capacidade máxima do elevador foi excedida ou não.

Tarefa

Escreva um programa que, dada uma sequência de leituras do sensor e a capacidade máxima do elevador, determina se a capacidade máxima do elevador foi excedida em algum momento.

Entrada

A primeira linha da entrada contém dois inteiros N e C , indicando o número de leituras realizadas pelo sensor e a capacidade máxima do elevador, respectivamente ($1 \leq N \leq 1000$ e $1 \leq C \leq 1000$). As N linhas seguintes contém, cada uma, uma leitura do sensor. Cada uma dessas linhas contém dois inteiros S e E , indicando quantas pessoas saíram e quantas pessoas entraram naquele andar, respectivamente ($0 \leq S \leq 1000$ e $0 \leq E \leq 1000$).

Saída

Seu programa deve imprimir uma única linha contendo o caractere 'S', caso a capacidade do elevador tenha sido excedida em algum momento, ou o caractere 'N' caso contrário.

Exemplo

Entrada
5 10
0 5
2 7
3 3
5 2
7 0

Saída

N

Entrada

5 10

0 3

0 5

0 2

3 4

6 4

Saída

S

Entrada

6 4

0 5

3 5

4 5

1 0

1 1

1 1

Saída

S

98. (SPOJ FATORIA2 - 3774. Fatorial) Dado um inteiro N , determine quanto vale N fatorial (escreve-se $N!$). O fatorial de um número é o produto de todos os números entre 1 e N , inclusive. Por exemplo, $5! = 5 * 4 * 3 * 2 * 1 = 120$.

Entrada

A entrada é composta por uma única linha que contém o inteiro N .

Saída

A saída deve consistir de apenas uma linha contendo o valor de $N!$.

Restrições

- $0 \leq N \leq 12$

Exemplo

Entrada

2

Saída

2

Entrada

3

Saída








6

99. (SPOJ JINGLES - 8488. Composição de jingles) A. C. Marcos está dando os primeiros passos para ser um compositor de jingles. Ele está tendo alguns problemas, mas ao menos ele está criando melodias agradáveis e ritmos atrativos.

Na música, uma nota tem um tom (sua frequência, resultando em quão grave ou agudo é o som) e uma duração (por quanto tempo a nota soa). Neste problema, estamos interessados apenas na duração das notas.

Um jingle é dividido em uma sequência de compassos, e um compasso é formado de uma série de notas.

A duração de uma nota é indicada pela sua forma. Neste problema, iremos utilizar letras maiúsculas para indicar a duração de uma nota. A seguinte tabela lista todas as notas disponíveis:

Notas							
Identificador	W	H	Q	E	S	T	X
Duração	1	1/2	1/4	1/8	1/16	1/32	1/64

A duração de um compasso é a soma da duração de suas notas. Nos jingles de Marcos, cada compasso tem a mesma duração. Como Marcos é apenas um iniciante, seu famoso professor Johann Sebastian III o ensinou que a duração de um compasso deve ser sempre 1.

Por exemplo, Marcos escreveu uma composição contendo cinco compassos, dentre os quais quatro possuem a duração correta e um está errado. No exemplo abaixo, cada compasso é delimitado com barras e cada nota é representada como na tabela acima.

/HH/QQQQ/XXTXTEQH/W/HW/

Marcos gosta de computadores assim como de música. Ele quer que você escreva um programa que determine, para cada uma de suas composições, quantos compassos possuem a duração correta.

Entrada

A entrada contém vários casos de teste. Cada caso de teste é descrito em uma única linha contendo uma string cujo tamanho está entre 3 e 200 caracteres, inclusive, representando uma composição. Uma composição começa e termina com uma barra '/'. Compassos em uma composição são separados por uma barra '/'. Cada nota em um compasso é representada pela letra correspondente segundo a descrição acima. Você pode assumir que cada composição contém ao menos um compasso e que cada compasso contém ao menos uma nota. Todos os caracteres na entrada serão barras ou uma das sete letras maiúsculas usadas para representar as notas.

O último caso de teste é seguido por uma linha contendo um único asterisco.

Saída

Para cada caso de teste, seu programa deve imprimir uma única linha contendo um único inteiro, o número de compassos que possuem a duração correta.

Exemplo

Entrada:

/HH/QQQQ/XXXTXTEQH/W/HW/

/W/W/SQHES/

/WE/TEX/THES/

*

Saída:

4

3

0

100. (SPOJ JTACOGRA - 8701. Tacógrafo) Tacógrafos são dispositivos instalados em determinados tipos de veículos, que registram a velocidade, tempo e distância percorrida por tal veículo. É utilizada principalmente em veículos de transporte coletivo e de transporte de cargas, assim ajudando a evitar abusos de velocidade por parte dos motoristas.

A empresa SBC (Sociedade Brasileira dos Caminhoneiros) decidiu encomendar uma versão um pouco mais básica (e barata) para seus associados não precisarem gastar tanto na instalação desses aparelhos. Essas versões modificadas registram apenas os intervalos de tempo e as velocidades médias do caminhão naqueles intervalos.

Tarefa

Apesar das restrições dos aparelhos novos, a SBC quer poder saber qual foi a distância percorrida pelos caminhões. Você deverá escrever um programa que recebe uma série de intervalos de tempo com suas respectivas velocidades médias e calcula qual foi a distância total percorrida pelo caminhão de acordo com o tacógrafo.

Entrada

A primeira linha da entrada contém um inteiro N ($1 \leq N \leq 1000$) representando a quantidade de intervalos de tempo registrados no tacógrafo. As N linhas seguintes descrevem os intervalos de tempo. Cada uma dessas linhas possui dois inteiros T e V ($1 \leq T \leq 100, 0 \leq V \leq 120$), que representam, respectivamente o tempo decorrido (em horas) e a velocidade média (em quilômetros por hora) no intervalo de tempo.

Saída

Seu programa deve imprimir uma única linha, contendo um único número inteiro representando a distância total percorrida, em quilômetros.

Exemplo

Entrada

```
3
10 0
55 12
75 120
```

Saída

```
9660
```

Entrada

```
10
45 46
46 101
7 2
95 104
12 107
78 29
10 26
52 86
13 79
1 107
```

Saída

```
26022
```

Entrada

```
8
37 24
68 69
28 26
79 8
36 0
50 71
13 68
87 113
```

101. (SPOJ LOTICA - 8305. Leitura ótica) O professor João decidiu aplicar somente provas de múltipla escolha, para facilitar a correção. Em cada prova, cada questão terá cinco alternativas (A, B, C, D e E), e o professor vai distribuir uma folha de resposta para cada aluno. Ao final da prova, as folhas de resposta serão escaneadas e processadas digitalmente para se obter a nota de cada aluno. Inicialmente, ele pediu ajuda a um sobrinho, que sabe programar muito bem, para escrever um programa para extrair as alternativas marcadas pelos alunos nas folhas de resposta. O sobrinho escreveu uma boa parte do software, mas não pode terminá-lo, pois precisava treinar para a Maratona de Programação.

Durante o processamento, a prova é escaneada usando tons de cinza entre 0 (preto total) e 255 (branco total). Após detectar os cinco retângulos correspondentes a cada uma das alternativas, ele calcula a média dos tons de cinza de cada pixel, retornando um valor inteiro correspondente àquela alternativa. Se o quadrado foi preenchido corretamente o valor da média é zero (preto total). Se o quadrado foi deixado em branco o valor da média é 255 (branco total). Assim, idealmente, se os valores de cada quadrado de uma questão são (255, 0, 255, 255, 255), sabemos que o aluno marcou a alternativa B para essa questão. No entanto, como as folhas são processadas individualmente, o valor médio de nível de cinza para o quadrado totalmente preenchido não é necessariamente 0 (pode ser maior); da mesma forma, o valor para o quadrado não preenchido não é necessariamente 255 (pode ser menor). O prof. João determinou que os quadrados seriam divididos em duas classes: aqueles com média menor ou igual a 127 serão considerados pretos e aqueles com média maior a 127 serão considerados brancos.

Obviamente, nem todas as questões das folhas de resposta são marcadas de maneira correta. Pode acontecer de um aluno se enganar e marcar mais de uma alternativa na mesma questão, ou não marcar nenhuma alternativa. Nesses casos, a resposta deve ser desconsiderada.

O professor João necessita agora de um voluntário para escrever um programa que, dados os valores dos cinco retângulos correspondentes às alternativas de uma questão determine qual a alternativa corretamente marcada, ou se a resposta à questão deve ser desconsiderada.

Entrada

A entrada contém vários casos de teste. A primeira linha de um caso de teste contém um número inteiro N indicando o número de questões da folha de respostas ($1 \leq N \leq 255$). Cada uma das N linhas seguintes descreve a resposta a uma questão e contém cinco números inteiros A, B, C, D e E , indicando os valores de nível de cinza médio para cada uma das alternativas da resposta ($0 \leq A, B, C, D, E \leq 255$).

O último caso de teste é seguido por uma linha que contém apenas um número zero.

Saída

Para cada caso de teste da entrada seu programa deve imprimir N linhas, cada linha correspondendo a uma questão. Se a resposta à questão foi corretamente preenchida na folha de resposta, a linha deve conter a alternativa marcada ('A', 'B', 'C', 'D' ou 'E'). Caso contrário, a linha deve

conter o caractere '*' (asterisco).

Exemplo

Entrada:

```
3
0 255 255 255 255
255 255 255 255 0
255 255 127 255 255
4
200 200 200 0 200
200 1 200 200 1
1 2 3 4 5
255 5 200 130 205
0
```

Saída:

```
A
E
C
D
*
*
B
```

102. (SPOJ MACACO - 814. Macaco-prego) O macaco-prego é um animal irrequieto e barulhento, merecedor também dos adjetivos desordeiro e despudorado. A sua cabeça, encimada por uma densa pelagem negra ou marrom-escura, semelhante a um gorro, torna seu aspecto inconfundível. Apesar de ser o macaco mais comum nas matas do país, uma de suas sub-espécies encontra-se seriamente ameaçada de extinção: o macacoprego- do-peito-amarelo, que se distingue das demais pela coloração amarelada do peito e da parte anterior dos braços.

Um grande esforço foi feito pelos primatologistas para aumentar a população dos macacos-pregodo- peito-amarelo. Sabe-se que eles se alimentam de plantas, das quais consomem preferencialmente frutos e brotos. Alimentam-se também de muitos animais, preferencialmente lesmas, lagartas e rãs, e preferem as florestas mais densas. Para determinar o melhor local do país para criar uma nova reserva ambiental para os macacos-prego-do-peito-amarelo, o governo fez um levantamento das regiões no país onde as condições preferidas desses animais ocorrem: regiões de floresta densa, regiões com frutos, regiões com muitos brotos, etc. Ajude a salvar os macacos-pregodo- peito-amarelo.

Tarefa

As regiões propícias para o macaco-prego-do-peito-amarelo foram determinadas como retângulos cujos lados são todos verticais ou horizontais. Sua tarefa é encontrar o local ideal para a reserva ambiental, definida como a interseção de todas as regiões dadas.



As regiões foram divididas de tal forma que uma região não tangencia qualquer outra região. Assim, a interseção entre quaisquer duas regiões ou é um retângulo ou é vazia.

Entrada

Seu programa deve ler vários conjuntos de teste. A primeira linha de um conjunto de teste contém um inteiro não negativo, N , que indica o número de regiões (o valor $N = 0$ indica o final da entrada). Seguem-se N linhas, cada uma contendo quatro números inteiros X , Y , U e V que descrevem uma região: o par X , Y representa a coordenada do canto superior esquerdo e o par U , V representa a coordenada do canto inferior direito de um retângulo.

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato "Teste n ", onde n é numerado a partir de 1. A segunda linha deve conter as coordenadas do retângulo de interseção encontrado pelo seu programa, no mesmo formato utilizado na entrada. Caso a interseção seja vazia, a segunda linha deve conter a expressão "nenhum". A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo

Entrada:

```
3
0 6 8 1
1 5 6 3
2 4 9 0
3
0 4 4 0
3 1 7 -3
6 4 10 0
0
```

Saída:

```
Teste 1
2 4 6 3
```

```
Teste 2
nenhum
```

Restrições

- $0 \leq N \leq 10000$ ($N = 0$ apenas para indicar o fim da entrada)
- $-10000 \leq X \leq 10000$
- $-10000 \leq Y \leq 10000$
- $-10000 \leq U \leq 10000$
- $-10000 \leq V \leq 10000$

103. (SPOJ MARAT09 - 11638. Maratona) A maratona é talvez a prova mais desgastante entre as modalidades olímpicas: são quarenta e dois mil, cento e noventa e cinco metros de percurso. Por isso, os organizadores sempre posicionam vários postos de água ao longo do trajeto da prova, onde copos de água são distribuídos aos competidores.

João Saci é um jovem atleta que tem boas chances de se tornar um maratonista de primeira linha. No entanto, João Saci descobriu que somente consegue terminar uma maratona se ingerir alguns copos de água durante o percurso. O Laboratório de Biomecânica da universidade local, através de experimentos, determinou que João Saci consegue percorrer exatamente mais dois mil metros após o instante em que ingere um copo de água. A distância que João Saci consegue percorrer após ingerir um copo de água é denominada de distância intermediária máxima. Assim, se a distância entre dois postos de água consecutivos no percurso da maratona for sempre menor ou igual do que a distância intermediária máxima de João Saci, ele consegue terminar a prova. Caso contrário ele não consegue terminar a prova.

O Laboratório de Biomecânica quer agora realizar estudos similares com outros maratonistas, que têm valor de distâncias intermediárias máximas distintas, e precisa de sua ajuda.

Tarefa

Sua tarefa é escrever um programa que, dada a posição dos postos de água ao longo do percurso, e a distância intermediária máxima de um atleta, determine se o atleta consegue ou não completar a prova.

Entrada

A entrada contém um único conjunto de testes, que deve ser lido do dispositivo de entrada padrão (normalmente o teclado).

A primeira linha da entrada contém dois números inteiros N e M , separados por um espaço em branco, indicando respectivamente o número de postos de água ($2 \leq N \leq 10000$) e a distância intermediária máxima de um atleta, em metros ($1 \leq M \leq 42195$). A segunda linha contém N números inteiros P_i , separados por um espaço em branco, representando a posição dos postos de água ao longo do trajeto da maratona. A posição de um posto de água é dada pela distância, em metros, do início do percurso até o posto de água ($0 \leq P_i \leq 42195$ para $1 \leq i \leq N$). O primeiro posto de água está sempre localizado no ponto de partida (ou seja, $P_1 = 0$) e todos os postos estão em posições distintas. Além disso, os postos de água são dados na ordem crescente

de sua distância ao início do percurso.

Note que a distância total da prova é a oficial para a maratona, ou seja, 42195 metros.

Saída

Seu programa deve imprimir, na saída padrão, uma única linha contendo o caractere 'S' se o atleta consegue terminar a prova, ou o caractere 'N' caso contrário.

Exemplos

Entrada

3 20000

0 20000 33333

Saída

S

Entrada

8 6000

0 6000 12000 18000 24000 32000 37000 40000

Saída

N

104. (SPOJ MATRIZ2 - 8782. Matrizes) O conglomerado indiano Tutu é um conjunto de empresas que atua nos mais diversos ramos da indústria, produzindo desde sapatos até aviões e foguetes. Por ser tão diversificada, precisa de grandes e rápidos sistemas para cálculos de contabilidade.

Um dos módulos mais importantes desse sistema é o de fornecimento de produtos, onde fica a base de dados de produtos e fornecedores. Um mesmo produto pode ser fornecido por vários fornecedores diferentes.

A outra grande matriz é a B, onde cada linha representa um dia do mês e cada coluna é um produto. O valor da matriz na linha m e coluna n representa a quantidade do produto n a ser adquirido no dia m.

Tal empresa tem uma política de fidelidade com seus fornecedores, e uma das práticas efetuadas pela empresa é, em um determinado dia, comprar todos os produtos necessários de um único fornecedor. Isto é, em um dia todos os produtos adquiridos serão comprados do fornecedor x, no outro dia do fornecedor y, e assim por diante

Para auxiliar a escolha de qual fornecedor será o escolhido no dia, foi gerada outra matriz C, que é o resultado da multiplicação das matrizes A B. Essa matriz diz o quanto será gasto pela empresa se adquirir todos os produtos de um determinado fornecedor em um determinado dia.

As matrizes A e B são quadradas (o número de linhas é igual ao número de colunas) e têm valores definidos pelas fórmulas

- $A_{ij} = (P * i + Q * j)(mod X)$

- $B_{ij} = (R * i + S * j)(mod Y)$

onde i é o índice da linha da matriz e j é o índice da coluna da matriz (todos os índices vão de 1 até N). Os inteiros P, Q, R, S, X e Y são parâmetros constantes, que definem as duas matrizes A e B .

Tarefa

Escreva um programa que, dados os parâmetros das matrizes A e B , e a posição de uma das entradas da matriz C , calcula o valor daquela entrada.

Entrada

A primeira linha da entrada contém um inteiro N , indicando as dimensões das matrizes A, B e C ($2 \leq N \leq 10^5$). A linha seguinte contém seis inteiros P, Q, R, S, X e Y , indicando os parâmetros das matrizes A e B ($2 \leq X, Y \leq 10^4; 0 \leq P, Q < X; 0 \leq R, S < Y$). Finalmente, a última linha da entrada contém dois inteiros I e J , indicando a linha e a coluna da matriz C a serem consultados ($1 \leq I, J \leq N$).

Saída

Seu programa deve imprimir uma única linha contendo o valor da matriz C na linha e coluna especificadas.

Exemplo

Entrada

```
3
4 3 2 3 5 6
2 2
```

Saída

```
18
```

Entrada

```
4
3 5 1 0 6 7
4 3
```

Saída

```
30
```

Entrada

```
2
```

```
2 2 0 1 3 2
2 1
```

Saída
2

105. (SPOJ METEORO - 1330. Meteoros) Em noites sem nuvens pode-se muitas vezes observar pontos brilhantes no céu que se deslocam com grande velocidade, e em poucos segundos desaparecem de vista: são as chamadas estrelas cadentes, ou meteoros. Meteoros são na verdade partículas de poeira de pequenas dimensões que, ao penetrar na atmosfera terrestre, queimam-se rapidamente (normalmente a uma altura entre 60 120 quilômetros). Se os meteoros são suficientemente grandes, podem não queimar-se completamente na atmosfera e dessa forma atingem a superfície terrestre: nesse caso são chamados de meteoritos.

Zé Felício é um fazendeiro que adora astronomia e descobriu um portal na Internet que fornece uma lista das posições onde caíram meteoritos. Com base nessa lista, e conhecendo a localização de sua fazenda, Zé Felício deseja saber quantos meteoritos caíram dentro de sua propriedade. Ele precisa de sua ajuda para escrever um programa de computador que faça essa verificação automaticamente.

Tarefa

São dados:

- uma lista de pontos no plano cartesiano, onde cada ponto corresponde à posição onde caiu um meteorito;
- as coordenadas de um retângulo que delimita uma fazenda.

As linhas que delimitam a fazenda são paralelas aos eixos cartesianos. Sua tarefa é escrever um programa que determine quantos meteoritos caíram dentro da fazenda (incluindo meteoritos que caíram exatamente sobre as linhas que delimitam a fazenda).

Entrada

Seu programa deve ler vários conjuntos de testes. A primeira linha de um conjunto de testes quatro números inteiros $X1$, $Y1$, $X2$ e $Y2$, onde $(X1, Y1)$ é a coordenada do canto superior esquerdo e $(X2, Y2)$ é a coordenada do canto inferior direito do retângulo que delimita a fazenda. A segunda linha contém um inteiro, N , que indica o número de meteoritos. Seguem-se N linhas, cada uma contendo dois números inteiros X e Y , correspondendo às coordenadas de cada meteorito. O final da entrada é indicado por $X1 = Y1 = X2 = Y2 = 0$.

Exemplo de entrada
2 4 5 1
2
1 2
3 3
2 4 3 2

```
3
1 1
2 2
3 3
0 0 0 0
```

Saída

Para cada conjunto de teste da entrada seu programa deve produzir três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste n”, onde n é numerado a partir de 1. A segunda linha deve conter o número de meteoritos que caíram dentro da fazenda. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de saída

Teste 1

1

Teste 2

2

(esta saída corresponde ao exemplo de entrada acima)

Restrições

- $0 \leq N \leq 10.000$
- $0 \leq X \leq 10.000$
- $0 \leq Y \leq 10.000$
- $0 \leq X1 < X2 \leq 10.000$
- $0 \leq Y2 < Y1 \leq 10.000$

106. (SPOJ MIOJO - 3826. Miojo) João é um fanático por miojos; ele os adora, e, como era de se esperar, ele levou vários pacotes quando foi acampar com seus colegas. Como João só gosta de miojos feitos com o tempo exato, ele se desesperou ao perceber que havia esquecido seu relógio em casa.

Por sorte, ele conseguiu, no caminho, comprar duas ampulhetas de durações diferentes. Por exemplo, se o miojo precisa de 3 minutos para ficar pronto, e João tiver uma ampulheta de 5 minutos e outra de 7, uma possível forma de cozinhar o miojo é:

- João começa virando as duas ampulhetas ao mesmo tempo.
- Quando a areia da ampulheta de 5 minutos se esgotar, João torna a virá-la.

- (c) João começa a preparar o miojo quando a areia da ampulheta de 7 minutos acabar.
- (d) João tira o miojo do fogo quando a ampulheta de 5 minutos acabar novamente.
- (e) Dessa forma, o miojo ficará 3 minutos no fogo (do minuto 7 ao minuto 10). Assim, apesar do miojo levar apenas três minutos para ser cozido, ele precisa de 10 minutos para ficar pronto.

Faça um programa que, dado o tempo de preparo do miojo, e os tempos das duas ampulhetas (ambos maiores que o tempo do miojo), determina o tempo mínimo necessário para o miojo ficar pronto. Você pode supor que sempre é possível cozinhar o miojo no tempo correto.

Entrada

A entrada contém um único caso de teste, composto por uma única linha, que contém três inteiros T , A e B , representando o tempo necessário para o preparo do miojo, o tempo da primeira ampulheta e o tempo da segunda ampulheta respectivamente.

Saída

Seu programa deve produzir uma única linha na saída, contendo o tempo mínimo para o preparo do miojo.

Restrições

- $0 \leq T \leq 10000$
- $T \nmid A, B \leq 40000$

Exemplo

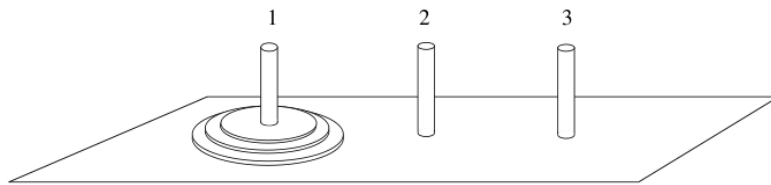
Entrada
3 5 7

Saída
10

Entrada
14 15 22

Saída
44

107. (SPOJ OBIHANOI - 842. Torres de Hanói) O quebra-cabeças Torres de Hanoi é muito antigo e conhecido, sendo constituído de um conjunto de N discos de tamanhos diferentes e três pinos verticais, nos quais os discos podem ser encaixados.



Cada pino pode conter uma pilha com qualquer número de discos, desde que cada disco não seja colocado acima de outro disco de menor tamanho. A configuração inicial consiste de todos os discos no pino 1. O objetivo do quebra-cabeças é mover todos os discos para um dos outros pinos, sempre obedecendo à restrição de não colocar um disco sobre outro menor.

Um algoritmo para resolver este problema é o seguinte.

```

procedimento Hanoi(N, Orig, Dest, Temp)
  se N = 1 então
    mover o menor disco do pino Orig para o pino Dest;
  senão
    Hanoi(N-1, Orig, Temp, Dest);
    mover o N-ésimo menor disco do pino Orig para o pino Dest;
    Hanoi(N-1, Temp, Dest, Orig);
  fim-se
fim

```

Tarefa

Sua tarefa é escrever um programa que determine quantos movimentos de trocar um disco de um pino para outro serão executados pelo algoritmo acima para resolver o quebra-cabeça.

Entrada

A entrada possui vários conjuntos de teste. Cada conjunto de teste é composto por uma única linha, que contém um único número inteiro N ($0 \leq N \leq 30$), indicando o número de discos. O final da entrada é indicado por 0.

Saída

Para cada conjunto de teste, o seu programa deve escrever três linhas na saída. A primeira linha deve conter um identificador do conjunto de teste, no formato “Teste n”, onde n é numerado seqüencialmente a partir de 1. A segunda linha deve conter o número de movimentos que são executados pelo algoritmo dado para resolver o problema das Torres de Hanói com N discos. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

Exemplo

Entrada:
1

```

2
0

Saída
Teste 1
1

Teste 2
3

```

Restrições

- $0 \leq N \leq 30$ ($N = 0$ apenas para indicar o final da entrada)

108. (SPOJ PAR - 1363. Par ou ímpar) Muitas crianças gostam de decidir todas as disputas através do famoso jogo de Par ou Ímpar. Nesse jogo, um dos participantes escolhe Par e o outro Ímpar. Após a escolha, os dois jogadores mostram, simultaneamente, uma certa quantidade de dedos de uma das mãos. Se a soma dos dedos das mãos dos dois jogadores for par, vence o jogador que escolheu Par inicialmente, caso contrário vence o que escolheu Ímpar.

Tarefa

Dada uma seqüência de informações sobre partidas de Par ou Ímpar (nomes dos jogadores e números que os jogadores escolheram), você deve escrever um programa para indicar o vencedor de cada uma das partidas.

Entrada

A entrada é composta de vários conjuntos de testes. A primeira linha de um conjunto de testes contém um inteiro N , que indica o número de partidas de Par ou Ímpar que aconteceram. As duas linhas seguintes contém cada uma um nome de jogador. Um nome de jogador é uma cadeia de no mínimo um e no máximo dez letras (maiúsculas e minúsculas), sem espaços em branco. As N linhas seguintes contém cada uma dois inteiros A e B que representam o número de dedos que cada jogador mostrou em cada partida ($0 \leq A \leq 5$ e $0 \leq B \leq 5$). Em todas as partidas, o primeiro jogador sempre escolhe Par. O final da entrada é indicado por $N = 0$.

```

Exemplo de entrada
3
Pedro
Paulo
2 4
3 5
1 0
2
Claudio
Carlos
1 5

```


2 3
0

Saída

Para cada conjunto de teste da entrada, seu programa deve produzir a saída da seguinte forma. A primeira linha deve conter um identificador do conjunto de teste, no formato "Teste n", onde n é numerado seqüencialmente a partir de 1. As próximas N linhas devem indicar o nome do vencedor de cada partida. A próxima linha deve ser deixada em branco. A grafia mostrada no Exemplo de 3 Saída, abaixo, deve ser seguida rigorosamente.

Exemplo de Saída

Teste 1

Pedro

Pedro

Paulo

Teste 2

Claudio

Carlos

(esta saída corresponde ao exemplo de entrada acima)

Restrições

- $0 \leq N \leq 1000$ ($N = 0$ apenas para indicar o fim da entrada)
- $0 \leq A \leq 5$
- $0 \leq B \leq 5$
- $1 \leq \text{comprimento do nome de jogador} \leq 10$

109. (SPOJ PRIMO - 3828. Primo) Tarefa

Dado um inteiro N, verifique se N é primo.

Entrada

A entrada é composta por um único caso de teste, composto por uma única linha que contém o inteiro N.

Saída

Seu programa deve produzir uma única linha, contendo a palavra "sim", se N for primo, e "nao", caso contrário (note a ausência de acentuação).

Restrições

- $|N| < 2^{31}$

Exemplo

Entrada

7

Saída

sim

Entrada

10

Saída

nao

110. (SPOJ SOMA - 3830. Soma) Dada uma lista de N inteiros, encontre a soma de todos eles.

Entrada

A entrada é composta de um único caso de teste. A primeira linha contém um inteiro positivo N. As N linhas seguintes contém cada uma um inteiro X, representando os N números a serem somados.

Saída

Seu programa deve produzir uma única linha na saída, contendo a soma de todos os N inteiros.

Restrições

- $0 \leq N \leq 50$
- $|X| \leq 5000$

Exemplo

Entrada

2

2

3

Saída

5

Entrada

3

1

111. (SPOJ TELESCO2 - 8783. Telescópio) Telescópios são instrumentos que auxiliam a observação do céu, melhorando e aumentando o aspecto das estrelas, planetas e outros objetos brilhantes. Existem diversos tipos de telescópios, sendo os tipos mais comuns os de lentes objetivas (refratores) e os de espelhos (refletores).

A maneira como os telescópios melhoram a nossa percepção dos astros no céu é aumentando a quantidade de luz captada que chega aos nossos olhos. Toda luz que entra pelos nossos olhos entra por um orifício chamado pupila. Tal controla a quantidade de luz que entra nos olhos, aumentando o diâmetro quando o ambiente está escuro (e portanto precisamos obter mais luz para identificar os objetos) e diminuindo quando o ambiente está claro. Num ambiente muito escuro, a pupila pode atingir um diâmetro de 8 mm.

Cada objeto celeste (estrela, planeta, nebulosa, etc) emite uma quantidade de luz (fótons) que é homogeneamente distribuída quando chega na Terra. Por exemplo, a estrela A emite luz que pode ser captada a um fluxo de 40.000 fótons por segundo por milímetro quadrado. Isso é, a cada segundo, é possível captar 40.000 fótons provenientes da estrela A numa área de 1 mm². Ou seja, uma pupila de 10mm² de área captaria 400.000 fótons provenientes da estrela A por segundo.

Para que nosso cérebro consiga interpretar que existe um objeto ali, porém, ele precisa receber 40.000.000 fótons por segundo. Assim, podemos utilizar um telescópio com lente (ou espelho) de 100mm² de área, que vai captar a quantidade necessária de fótons provenientes da estrela A e encaminhá-los até nossa pupila, fazendo assim com que nosso cérebro perceba a presença da estrela ali.

Tarefa

Dada uma lista com estrelas no céu, o fluxo de fótons que cada uma delas emite, e área de abertura de um telescópio, dizer quantas estrelas serão perceptíveis usando tal telescópio.

Entrada

A primeira linha da entrada terá um inteiro A ($1 \leq A \leq 10.000$) representando a área de abertura do telescópio (em milímetros quadrados) a ser considerado. A segunda linha possui um inteiro N ($1 \leq N \leq 10.000$) representando o número de estrelas a serem estudadas. As N linhas seguintes terão, cada uma, um inteiro F ($1 \leq F \leq 20.000$) representando o fluxo de fótons que cada uma das N estrelas emitem (em fótons por segundo por milímetro quadrado).

Saída

Imprima um inteiro representando a quantidade de estrelas que serão percebidas ao se utilizar o telescópio em questão.

Exemplo

Entrada
10000
3
4000
3500
5100

Saída
2

Entrada
5869
3
3975
14234
8569

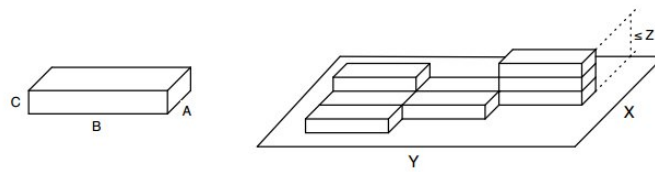
Saída
2

Entrada
2967
9
18650
16338
2400
17702
14619
13934
7979
16316
1053

Saída
6

112. (SPOJ TRANSP11 - 10866. Transporte) A Betalândia é um país que apenas recentemente se abriu para o comércio exterior e está preparando agora sua primeira grande exportação. A Sociedade Betalandesa de Comércio (SBC) ficou encarregada de conduzir a exportação e determinou que, seguindo os padrões internacionais, a carga será transportada em contêineres, que são, por sua vez, colocados em grandes navios para o transporte internacional.

Todos os contêineres betalandeses são idênticos, medindo A metros de largura, B metros de comprimento e C metros de altura. Um navio porta-contêineres pode ser visto como um retângulo horizontal de X metros de largura e Y metros de comprimento, sobre o qual os contêineres são colocados. Nenhuma parte de contêiner pode ficar para fora do navio. Além disso, para possibilitar a travessia de pontes, a altura máxima da carga no navio não pode ultrapassar Z metros.



Devido a limitações do guindaste utilizado, os contêineres só podem ser carregados alinhados com o navio. Ou seja, os contêineres só podem ser colocados sobre o navio de tal forma que a largura e o comprimento do contêiner estejam paralelos à largura e ao comprimento do navio, respectivamente.

A SBC está com problemas para saber qual a quantidade máxima de contêineres que podem ser colocados no navio e pede sua ajuda. Sua tarefa, neste problema, é determinar quantos contêineres podem ser carregados no navio respeitando as restrições acima.

Entrada

A entrada consiste de duas linhas. A primeira linha contém três inteiros A, B e C que representam as dimensões dos contêineres, enquanto a segunda linha contém outros três inteiros X, Y e Z que representam as dimensões do navio.

Saída

Seu programa deve imprimir apenas uma linha contendo um inteiro que indica a quantidade máxima de contêineres que o navio consegue transportar.

Restrições

- $1 \leq A, B, C, X, Y, Z \leq 10^6$
- É garantido que a maior resposta será menor ou igual a 10^6

Exemplos

Entrada

```
1 1 1
1 1 1
```

Saída

```
1
```

Entrada

```
1 2 5
9 6 11
```

Saída

54

Entrada

1 2 12

6 9 10

Saída

0