

Apostila de Exercícios do Curso Preparatório para a OBI

Edison Thomaz
Hilda Alves
Luciano Gonda
Priscila Martins
Ricardo Santos

Capítulo 1

Lista de Exercícios - Lógica

Esta lista de exercícios é composta por exercícios teóricos que abordam raciocínio lógico.

1. Três músicos, João, Antônio e Francisco, tocam harpa, violino e piano. Contudo, não se sabe quem toca o quê. Sabe-se que o Antônio não é o pianista. Mas o pianista ensaia sozinho na Terça. O João ensaia com o Violoncelista às Quintas. Quem toca o quê?
2. Quantas rainhas são necessárias e em que posições devem estar localizadas para que todas as casas desocupadas de um tabuleiro de xadrez fiquem atacadas por uma das rainhas?
3. Temos 32 golfistas que vão jogar individualmente um torneio em N semanas. Pretende-se que cada jogador só jogue uma vez um contra o outro. Sabendo que ocorre apenas 1 jogo por semana, qual o número mínimo de semanas antes de haver repetições?
4. No antigo Egito, havia um prisioneiro numa cela com duas saídas, cada uma delas com um guarda. Cada saída dava para um corredor diferente em que um dava para o campo e, portanto, para a liberdade e o outro para um fosso de crocodilos. Só os guardas sabiam qual a saída certa, mas um deles dizia sempre a verdade e outro mentia sempre. O prisioneiro não sabia nem qual a saída certa nem qual o guarda verdadeiro. Qual a pergunta (e uma só pergunta) que o prisioneiro deveria fazer a um dos guardas ao acaso, para saber qual a porta certa?
5. Três pessoas se registram em um hotel. Elas pagam R\$30,00 ao gerente e vão para seus quartos. O gerente nota que a diária é de R\$25,00 e entrega R\$5,00 ao mensageiro do hotel para ele devolver aos hóspedes. No caminho, o mensageiro conclui que seria complicado dividir R\$5,00 entre 3 pessoas, então ele embolsa R\$2,00 e entrega R\$1,00 para cada pessoa. Agora, cada um pagou R\$10,00 e pegou R\$1,00 de volta. Então, cada um pagou R\$9,00, totalizando R\$27,00. O mensageiro tem R\$2, totalizando R\$29. Onde está o outro R\$1,00?

6. São dados N azulejos de dimensões 10cm x 10cm. Com eles, você deve montar um conjunto de quadrados (com espessura de um azulejo) de modo a utilizar *TODOS* os azulejos dados. Inicialmente você deve montar o maior quadrado possível com os azulejos dados; então, com os azulejos que sobraram, você deve montar o maior quadrado possível, e assim sucessivamente. Por exemplo, se forem dados 31 azulejos, o conjunto montado terá quatro quadrados, conforme ilustra a Figura 1.1:

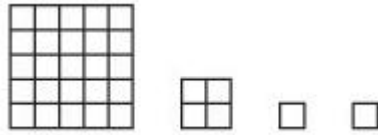


Figura 1.1: *Conjunto de quatro quadrados montados a partir de 31 azulejos.*

- Qual o número de quadrados do conjunto montado se forem dados 347 azulejos?
Qual o número de quadrados do conjunto montado se forem dados 148 azulejos?
Qual o número de quadrados do conjunto montado se forem dados 75 azulejos?
Qual o número de quadrados do conjunto montado se forem dados 40 azulejos?
7. Matemáticos gostam de usar sequências de números inteiros que seguem várias leis de formação interessantes. Por exemplo, $0, 1, 4, 9, 16, \dots$ é a sequência dos quadrados dos números consecutivos $0, 1, 2, 3, 4, \dots$. Nos problemas a seguir serão dadas algumas sequências de números e você deverá descobrir o número que falta, indicado por uma interrogação.
- (a) 4, 9, 25, 49, 121, ? , 289, 361 ...
(b) 19, 28, 37, ?, 55, 64, ...
(c) 2, 6, ?, 20, 30, 42, 56, 72, ...
8. Uma pessoa resolve contar usando a mão esquerda da seguinte maneira: Ela começa com 1 no dedão, 2 no dedo indicador, 3 no médio, 4 no anelar, 5 no mínimo e depois inverteu a ordem, contando 6 no anelar, 7 no médio, 8 no indicador, 9 no dedão, 10 novamente no dedo indicador e assim por diante. Em qual dedo a pessoa parou se contou até:
- (a) 222
(b) 534
(c) 1001
9. Temos 8 bolas de aparência idêntica, mas uma delas pesa mais do que as outras 7, que por sua vez têm o mesmo peso. Temos também uma balança de pratos simples. Como podemos identificar qual é a bola mais pesada efetuando o mínimo de medições possíveis?

10. Em uma certa comunidade, todos os nativos sempre mentem e os visitantes sempre falam a verdade. Um estrangeiro encontra-se com três pessoas e pergunta ao primeiro se ele é um nativo e recebe uma resposta. O segundo informa então que o primeiro falou que ele é um nativo, mas o terceiro afirma que o primeiro é um nativo. Quantas dessas pessoas são nativos?

Capítulo 2

Lista de Exercícios 2 - Operadores Aritméticos

Esta lista de exercícios é composta por exercícios teóricos e práticos que abordam os conceitos de tipos de variáveis e a utilização de operadores aritméticos da linguagem C.

1. O que é uma variável?
2. Quais variáveis a seguir são aceitas pelo compilador C?
 - n1
 - 1h
 - ijk_2
 - 3_jdk
 - if
 - Ac5
 - var inteira
 - papel-braco
3. Indique a função dos operadores aritméticos a seguir:
 - (a) *:
 - (b) /:
 - (c) %:
4. Indique o valor de r após cada expressão abaixo: (dados: $int\ a = 6$; $float\ b = 5.5$; $int\ c = 4$; $float\ d = 4.0$)
 - (a) $r = 5 + a$;
 - (b) $r = a * a$;

- (c) $r = a / b$;
- (d) $r = a \% c$;
- (e) $r = a*b/c$;
- (f) $r = a*(b/c)$;
- (g) $r = a*b+(5/a)*(c*b)$;

5. Construa um programa na linguagem C que receba como entrada dois números, faça a soma e a média desses dois números e exiba somente a média.
6. Construa um programa na linguagem C que receba um número n e imprima o valor correspondente ao seu quadrado (n^2).
7. Construa um programa na linguagem C que receba como entrada o peso e a altura de uma pessoa e calcule o seu IMC. O IMC é dado pela fórmula:

$$IMC = \frac{peso}{altura^2}$$

8. Construa um programa na linguagem C que receba como entrada 2 valores inteiros e armazene em uma variável a e b , depois troque os valores de a com b e imprima-os na tela.
9. Construa um programa que leia uma quantidade em horas, minutos, segundos e informe a quantidade total de segundos equivalente.
10. Construa um programa que leia uma Velocidade(V) em m/s e converta para km/h .
11. Construa um programa que leia a quantidade de quilômetros rodados e a quantidade gasta de combustível em litros por um carro e informe a média de combustível gasto em cada quilômetro rodado.
12. Construa um programa que calcule a quantidade de dinheiro gasto por um fumante com cigarros durante n anos. Para isso, é necessário ler a quantidade de cigarros que o fumante fuma por dia, a quantidade de anos que ele fuma e o preço médio de uma carteira de cigarros. (OBS: 1-Cada carteira de cigarros contém 20 cigarros. 2-Cada ano têm 365 dias.)
13. Construa um programa que receba o valor de uma temperatura em graus Celsius(C) e calcule a sua temperatura correspondente em graus Fahrenheit(F).

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

14. Construa um programa que receba o preço de custo de um produto e calcule o preço final do mesmo sabendo:

- O preço final é calculado através da soma do preço de custo, o valor dos impostos e o lucro esperado.
 - O valor dos impostos é de 45% do valor do preço de custo.
 - O lucro esperado é de 50% do valor do preço de custo.
15. Construa um programa que calcule o gasto de uma viagem de carro de uma cidade a outra, sabendo:
- O carro utilizado roda 15 KM com 1 litro de gasolina.
 - O preço da gasolina é de R\$2,60.
 - O valor de cada pedágio é de R\$8,00.

Seu programa deve receber a distância e a quantidade de pedágios entre as cidades.

16. Construa um programa que recebe como entrada a altura(h) e o sexo da pessoa e calcule seu peso ideal, utilizando as seguintes expressões:
- homens: $72 * h - 58$
 - mulheres: $62.1 * h - 44.7$
17. Construa um programa que recebe como entrada a quantidade de latão em Kg e informe ao usuário a quantidade de cobre e zinco presente no mesmo, sabendo que o latão é feito de 70% de cobre e de 30% de zinco.
18. Construa um programa que receba o número de lados N de um polígono convexo, calcule o número de diagonais diferentes (nd) deste polígono pela fórmula:

$$nd = \frac{n(n-3)}{2}$$

19. Construa um programa que receba o primeiro termo (a_1) de uma Progressão Aritmética (PA), sua razão (r), um número N e, a seguir, calcule e imprima o N -ésimo termo da PA pela fórmula:

$$a_n = a_1 + (n - 1)r$$

20. Construa um programa que calcule o valor acumulado de uma poupança programada. Seu programa deverá receber o valor da constante de aplicação mensal(P), a taxa (i) e número de meses (N). Utilize a fórmula:

$$valor_acumulado = P * \frac{(1+i)^N - 1}{i}$$

21. Construa um programa que receba dois pontos $P1(x_1, y_1)$ e $P2(x_2, y_2)$ e calcule e imprima a distância entre esses dois pontos pela seguinte fórmula:

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

22. Construa um programa que receba as dimensões de uma cozinha (comprimento, altura e largura) e calcule quantas caixas de azulejos são necessárias para azulejar todas as paredes exceto o teto (considere que não será descontada a área ocupada por janelas e portas). Cada caixa de azulejo possui $1,5m^2$.
23. Um sistema de equações lineares da forma:

$$Ax + By = C$$

$$Dx + Ey = F$$

Pode ser resolvido utilizando as seguintes fórmulas:

$$x = \frac{CE - BF}{AE - BD} \text{ e } y = \frac{AF - CD}{AE - BD}$$

Construa um programa que receba o valor dos coeficientes (A, B, C, D, E e F) e calcule os valores de x e y . Existe algum caso em que o algoritmo não funciona?

24. Construa um programa que receba um valor X de dinheiro e calcule o número mínimo de notas necessárias para se obter a quantia X de dinheiro (Considere que existem apenas notas de R\$50,00, R\$20,00, R\$10,00, R\$5,00, R\$2,00 e R\$1,00). **Exemplo:** Para R\$98,00 são necessárias:
- 1 Nota de R\$50,00
 - 2 Notas de R\$20,00
 - 1 Nota de R\$5,00
 - 1 Nota de R\$2,00
 - 1 Nota de R\$1,00

Capítulo 3

Lista de Exercícios - Estruturas Condicionais

Esta lista de exercícios é composta por exercícios teóricos e práticos que abordam os conceitos de estruturas condicionais e operadores lógicos da linguagem C.

1. Determine os resultados lógicos das expressões mencionadas, assinalando se são verdadeiras ou falsas. Considere para as respostas os seguintes valores: $X = 1$, $A = 3$, $B = 5$, $C = 8$ e $D = 7$.
 - (a) ☐ $(X \geq 2)$
 - (b) ☐ **não** $(X > 3)$
 - (c) ☐ $(X < 1)$ e $(B \geq D)$
 - (d) ☐ $(X < 1)$ e **não** $(B > D)$
 - (e) ☐ $(D < 0)$ ou $(C > 5)$
 - (f) ☐ **não** $(D < 0)$ e $(C > 5)$
 - (g) ☐ **não** $(D > 3)$ ou **não** $(B < 7)$
 - (h) ☐ **não** $(X > 3)$ ou $(C < 7)$
 - (i) ☐ $(A > B)$ ou **não** $(C > B)$
 - (j) ☐ $(A > B)$ ou $(C > B)$
2. Construa um programa que leia um número N e informe se o mesmo é par ou ímpar.
3. Construa um programa que leia duas notas de um aluno, calcule a média e informe se ele foi **Aprovado** ou **Reprovado** sabendo que a média do colégio é 6,0.
4. Construa um programa que leia um símbolo de operação do usuário (+, -, / ou *) e dois números reais. O programa deve retornar o resultado da operação recebida sobre esses dois números.

5. Construa um programa que leia três números inteiros e os imprima em ordem crescente. Primeiro, verifique se os números são diferentes.
6. Construa um programa que leia o nome, o sexo e o estado civil de uma pessoa. Caso sexo seja “F” e o estado civil seja “CASADA”, solicitar o tempo de casada (anos).
7. Construa um programa que leia 2 números a e b , informe se a é múltiplo de b , ou se b é múltiplo de a , ou se nenhuma das opções anteriores acontece.
8. Construa um programa que leia um número inteiro e verifique se o número correspondente é um mês válido no calendário e escrever o nome do mês. Senão, escrever uma mensagem ‘*Mês Inválido*’.
9. Construa um programa que leia 3 valores (considere que não serão informados valores iguais) e calcule a soma dos 2 maiores.
10. Construa um programa que leia o valor da largura e do comprimento de uma figura geométrica, informe se esta é um quadrado ou um retângulo.
11. Uma empresa dá um prêmio aos funcionários que cumprem ou ultrapassam determinado valor de vendas de produtos. A cada funcionário foi estabelecido um valor a ser alcançado. Faça um algoritmo para ler o valor fixado e o valor de vendas de um funcionário e imprimir a mensagem “Ganhou!” se o funcionário tiver conseguido o prêmio ou “Não ganhou!” se não tiver conseguido.
12. São dados graus para as notas de um exame conforme Tabela 3.1:

Nota	Grau
0 - 39	F
40 - 49	E
50 - 59	D
60 - 69	C
70 - 79	B
> 80	A

Tabela 3.1: Grau da nota do aluno.

Construa um programa que leia a nota de um aluno e imprima o Grau.

13. Construa um programa que leia três valores para os lados de um triângulo, considerando lados como: A , B e C . O programa deverá verificar se os lados fornecidos formam realmente um triângulo. Se esta condição for verdadeira, deverá ser indicado qual tipo de triângulo foi formado: isósceles, escaleno ou equilátero.
14. Construa um programa que leia um conjunto de 4 valores i, a, b, c , onde i é um valor inteiro e positivo e a, b, c , são quaisquer valores reais. Exibir na tela os valores; a , b e c de acordo com as seguintes condições:

- $i = 1$ escrever os três valores a, b, c em ordem crescente.
 - $i = 2$ escrever os três valores a, b, c em ordem decrescente.
 - $i = 3$ escrever os três valores a, b, c de forma que o maior entre a, b, c fique dentre os dois.
15. Construa um programa que leia a data de nascimento (dd/mm/aa) de uma pessoa e a data atual, calcule e informe a idade da pessoa.
16. Construa um programa que leia o salário-base de um funcionário, calcule e mostre o salário a receber, sabendo-se que esse funcionário tem gratificação de 5% sobre o salário base e paga imposto de 7% sobre o total.
17. As maçãs custam R\$ 1,30 cada se forem compradas menos de uma dúzia, e R\$ 1,00 se forem compradas pelo menos 12. Construa um programa que leia o número de maçãs compradas, calcule e escreva o custo total da compra.
18. Construa um programa que leia a idade de um nadador e 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
 - adulto = maiores de 18 anos
19. Construa um programa que leia o número de identificação do aluno, as 3 notas obtidas pelo aluno nas 3 provas e a média dos exercícios que fazem parte da avaliação. Calcular a média de aproveitamento, usando a fórmula:

$$MA = (Nota1 + Nota2 * 2 + Nota3 * 3 + ME) / 7$$

A atribuição de conceitos obedece a Tabela 3.2: Seu programa deve informar o número

Média de Aproveitamento	Conceito
$\geq 9,0$	A
$\geq 7,5$ e $< 9,0$	B
$\geq 6,0$ e $< 7,5$	C
$\geq 4,0$ e $< 6,0$	D
$< 4,0$	E

Tabela 3.2: Média de aproveitamento.

do aluno, suas notas, a média dos exercícios, a média de aproveitamento, conceito correspondente e a mensagem: APROVADO se o conceito for A,B ou C e REPROVADO se o conceito for D ou E.

20. Construa um programa que leia a hora de início de um jogo e a hora do final do jogo (considerando horas e minutos inteiros) e calcule a duração do jogo em horas, sabendo-se que o tempo máximo de duração do jogo é de 24 horas e que o jogo pode iniciar em um dia e terminar no dia seguinte.
21. A Secretaria de Meio Ambiente controla o índice de poluição e mantém 3 grupos de indústrias que são altamente poluentes do meio ambiente. O índice de poluição aceitável varia de 0,05 até 0,29. Se o índice sobe para 0,3 as indústrias do 1º grupo são intimadas a suspenderem suas atividades, se o índice crescer para 0,4 as indústrias do 1º e 2º grupo são intimadas a suspenderem suas atividades, se o índice atingir 0,5 todos os grupos devem ser notificados a paralisarem suas atividades. Faça um programa que leia o índice de poluição medido e emita a notificação adequada aos diferentes grupos de empresas.
22. Nos últimos dias, o governo brasileiro resolveu aumentar a taxa de juros anual para 26%. Isso fez com que a associação de donas de casa de Campo Grande-MS ficasse preocupada com o preço de produtos utilizados no dia-a-dia. Dessa forma, a associação dividiu seus integrantes em diversas equipes para fazer consulta de preços. Dona Ofélia, uma das associadas, ficou responsável pela consulta de preços de arroz, óleo de soja e cenoura nos supermercados Xilimlim e Tabajara. Como possui um conhecimento do uso do computador, ela decidiu utilizá-lo para realizar os cálculos dos preços. Sua tarefa é ajudá-la e desenvolver um programa que leia os preços dos três produtos em ambos os supermercados. Em seguida, o programa deve determinar qual a soma dos preços de todos os produtos em cada um dos supermercados. Por fim, seu programa deve informar qual o melhor supermercado para comprar os três produtos juntos.
23. Construa um programa que leia três números representando o dia, o mês e o ano de uma data (o ano deve ter 4 dígitos). Em seguida, seu programa deve verificar se a data é válida. Obs.: Lembre-se que um ano é bissexto quando é múltiplo de 4, mas não é múltiplo de 100, com exceção dos anos múltiplos de 400.
24. Uma empresa paga a seus funcionários R\$ 1,00 de comissão para cada produto vendido, entretanto, se forem vendidos mais de 250 produtos, o valor aumenta para R\$ 1,50. Se a quantidade for superior a 500 produtos, o valor da comissão sobe para R\$ 2,00. Construa um programa que leia o nome de um funcionário e a quantidade de produtos que ele vendeu. Exiba o nome do funcionário e o total de comissão que ele vai receber.
25. Uma cidade é classificada de acordo com o índice de poluição da seguinte forma: índice de poluição menor que 35 como agradável; de 35 até 60 desagradável, e acima de 60 perigoso. Joãozinho é extremamente alérgico à poluição, por isso, precisa se mudar para uma cidade menos poluída possível. Além disso, ele também não gosta de cidades

muito populosas. Construa um programa que leia o nível de poluição de uma cidade e quantidade de habitantes. Após a leitura, classifique a cidade da seguinte forma:

- Boa: somente se o nível de poluição for menor que 35 e a cidade tiver menos que 20.000 habitantes.
- Razoável: se o nível de poluição for agradável ou desagradável e o número de habitantes for menor que 20.000.
- Ruim: se o nível for desagradável ou ruim e o número de habitantes for maior que 40.000.
- Péssima: se o nível de poluição for maior que 60 e o número de habitantes for maior que 100.000.

26. Uma empresa concederá um aumento de salário aos seus funcionários, variável de acordo com o cargo, conforme a Tabela 3.3. Construa um programa que leia o salário e o cargo de um funcionário e calcule o novo salário. Se o cargo do funcionário não estiver na tabela, ele deverá, então, receber 40% de aumento. Mostre o salário antigo, o novo salário e a diferença.

Código	Cargo	Percentual
101	Gerente	30%
102	Engenheiro	20%
103	Técnico	10%

Tabela 3.3: Porcentagem de aumento de salário por cargo.

27. Um banco concederá um crédito especial aos seus clientes, variável com o saldo médio no último ano. Construa um programa em C que leia o saldo médio de um cliente e calcule o valor do crédito de acordo com a Tabela 3.4. Mostre uma mensagem informando o saldo médio e o valor do crédito.

Saldo Médio	Crédito
até R\$200,99	-
R\$201,00 a R\$400,99	20% do valor do saldo médio
R\$401,00 a R\$600,00	30% do valor do saldo médio
acima de R\$600,00	40% do valor do saldo médio

Tabela 3.4: Porcentagem de crédito oferecido.

Capítulo 4

Lista de Exercícios - Estruturas de Repetição

4 Esta lista de exercícios é composta por exercícios teóricos e práticos que abordam os conceitos de estruturas de repetição da linguagem C.

1. Construa um programa que leia dois números inteiros I e F , gere e imprima os números ímpares dentro do intervalo $[I, F]$. Primeiro verifique qual é o maior.
2. Construa um programa que leia n números inteiros até que o usuário digite 0. Em seguida, seu programa deve informar a média dos números lidos.
3. Construa um programa que leia um número n e informe se ele é primo ou não.
4. José tem 1,50 m e cresce 2 centímetros por ano. Pedro tem 1,10 m e cresce 3 centímetros por ano. Construa um programa que calcule em quantos anos Pedro será maior que José.
5. Construa um programa que leia dois números inteiros e informe se um número é permutação do outro. **Exemplo:** 432 é permutação de 234, 1121 é permutação de 1211.
6. Construa um programa que leia um número inteiro e exiba o maior número primo que seja menor do que o número digitado.
7. Construa um programa que imprima a tabuada dos números de 1 a 10.
8. Construa um programa que leia um número n , em seguida leia uma sequência de n números inteiros e verifique se a sequência está ordenada Crescentemente, Decrescentemente ou está desordenada. **Exemplo:**

A sequência 1,4,5,7,12,54,78 está ordenada crescentemente.

9. Construa um programa que leia um número inteiro n e imprima o valor da seguinte soma:

$$\frac{1}{n} + \frac{2}{n-1} + \frac{3}{n-2} + \dots \frac{n}{1}$$

10. Um Pet Shop deseja calcular o custo de criação de porquinhos da Índia. O custo é encontrado pela fórmula:

$$Custo = (H * 0,8)/2 + 10$$

onde H é o número de porquinhos da Índia e $Custo$ é custo em dólar para se criar os porquinhos. Construir um programa que leia sucessivos valores de H e calcule os respectivos custos. O programa deve parar quando um valor negativo de H for fornecido.

11. A UCDB realizou uma pesquisa sobre algumas características físicas de seus alunos, e necessita de um programa que tabule e informe:
- qual a maior idade entre os alunos;
 - qual a menor idade entre alunos;
 - o total de alunos do sexo masculino;
 - o total de alunos do sexo feminino.

As respostas foram codificadas da seguinte maneira:

Idade: valor numérico indicando o número de anos de vida;

Sexo: “M” para masculino e “F” para feminino;

Obs.: O programa dever terminar quando o usuário digitar ”s”(sim) em uma pergunta se deseja continuar ou não.

12. A empresa Unidos & Cansados Ltda., realizou uma pesquisa sobre a aceitação de seu produto alimentício e necessita de um programa que tabule e informe:
- quantas mulheres maiores de 40 anos indicaram o produto como bom;
 - quantas mulheres maiores de 50 anos indicaram o produto como ruim;
 - quantos homens indicaram o produto como péssimo;
 - total de homens que participaram da pesquisa.
 - total de mulheres participaram da pesquisa.

As respostas foram codificadas da seguinte maneira:

Idade: valor numérico indicando o numero de anos de vida;

Sexo e “M” para homens e “F” para mulheres;

Opinião com relação ao produto: 1-péssimo 2-ruim 3-regular 4-bom 5-ótimo.

Obs.: O programa dever terminar quando o usuário digitar “s” (sim).

13. Construa um programa que leia um número n natural e verifique se n é triangular ou não. Dizemos que um número natural é triangular se ele é produto de três números naturais consecutivos. **Exemplo:**

120 é triangular, pois $4 \cdot 5 \cdot 6 = 120$.

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

$$n = 1 \dots 1$$

$$n = 2 \dots 1$$

$$n = 3 \dots 2$$

$$F(n) = F(n-1) + F(n-2)$$

Construa um programa que leia um número inteiro n e informe $F(n)$.

15. Construa um programa que leia dois números inteiros x e n , calcule x^n usando apenas a operação de soma e informe o resultado ao usuário.
16. Construa um programa que leia dois números inteiros positivos e determine o Máximo Divisor Comum entre eles utilizando o algoritmo de Euclides. Exemplo Figura 4.1:

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

Figura 4.1: *Algoritmo de euclides.*

17. Construa um programa que dados n e uma sequência de n números inteiros, determina o comprimento de um segmento crescente de comprimento máximo. **Exemplos:**

Na sequência 5, 10, 3, **2, 4, 7, 9**, 8, 5 o comprimento do segmento crescente máximo é **4**. Na sequência 10, 8, 7, 5, 2 o comprimento de um segmento crescente máximo é 1.

18. Construa um programa que dados n e uma sequência de n números inteiros, determina quantos segmentos de números iguais consecutivos compõem essa sequência. **Exemplo:**

A sequência 5, 2, 2, 4, 4, 4, 4, 1, 1 é formada por 4 segmentos de números iguais.

19. Sabe-se que um número da forma n^3 é igual a soma de n ímpares consecutivos. **Exemplo:**

$$1^3 = 1,$$

$$2^3 = 3 + 5,$$

$$3^3 = 7 + 9 + 11,$$

$$4^3 = 13 + 15 + 17 + 19,$$

...

Construa um programa que dado n , determine os ímpares consecutivos cuja soma é igual a n^3 para n assumindo valores de 1 a n .

20. Para realizar a totalização dos votos de uma eleição para um cargo majoritário com 3 candidatos, leia os votos de cada seção até que seja digitado o número da seção 0 (zero). Para cada seção são informados o número de votos do candidato A, B, C, o número de votos brancos e nulos. Então determine:

- (a) O número de votantes
- (b) O total de votos de cada candidato
- (c) O total de votos brancos e de votos nulos
- (d) O total de votos válidos
- (e) O candidato com maior votação
- (f) Se a eleição foi válida. Para isso o total de votos brancos mais votos nulos deve ser menor que o total de votos válidos
- (g) Se haverá segundo turno. Para não haver segundo turno basta que o total de votos do candidato vencedor seja maior que 50% dos votos válidos.

21. Construa um programa que leia um número inteiro positivo n , verifica e informa se o mesmo é perfeito ou não. Dizemos que n é perfeito se for igual à soma de seus divisores positivos diferentes de n . **Exemplo:**

$$28 \text{ é perfeito, pois } 1 + 2 + 4 + 7 + 14 = 28$$

22. Construa um programa que leia um número inteiro na base decimal e determine o seu equivalente na base binária. O resultado será um número inteiro que representa um número binário.
23. Construa um programa que leia um número inteiro positivo n e imprime o mesmo na ordem inversa. **Exemplo:**

Dado 26578 a saída deverá ser 87562.

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

1297: 12 e 97.

5314: 53 e 14.

Construa um programa que imprima todos os números de quatro algarismos cuja raiz quadrada seja a soma das dezenas formadas pela divisão acima. **Exemplo:**

$$\sqrt{9801} = 99 = 98 + 01.$$

Portanto, 9801 é um dos números a ser impresso.

25. Construa um programa que leia um número natural, verifica e informa se o mesmo é palíndromo ou não. Dizemos que um número natural n com pelo menos 2 algarismos é palíndromo se o primeiro algarismo de n é igual ao seu último algarismo; o segundo algarismo de n é igual ao penúltimo algarismo; e assim sucessivamente. **Exemplos:**
- 567765 é palíndromo;
32423 é palíndromo;
567675 não é palíndromo.

Capítulo 5

Lista de Exercícios - Vetores

Esta lista de exercícios é composta por exercícios teóricos e práticos que abordam os conceitos de manipulação de vetores da Linguagem C.

1. Indique quais das variáveis a seguir representa um vetor de números reais:
 - (a) `() int A;`
 - (b) `() float A[5];`
 - (c) `() char A[5];`
2. Faça um programa que leia 10 números e os armazene em um vetor, em seguida, imprima-os.
3. Faça um programa que leia 5 números inteiros, armazene-os em um vetor e em seguida conte quantos elementos são negativos e informe ao usuário. Por fim, imprima todos os elementos do vetor.
4. Faça um programa que leia dois vetores de 12 elementos e em seguida calcule a soma dos vetores, elemento a elemento, e armazene em um terceiro vetor.
5. Leia um vetor de 10 elementos e em seguida encontre a posição do elemento x (dado pelo usuário) no vetor. Caso o elemento não exista no vetor informe ao usuário.
6. Escreva um programa que faz a leitura do vetor `A[20]` de inteiros apenas com valores distintos, ou seja, valores diferentes. Caso o usuário insira um inteiro já existente, o programa deve alertar a duplicidade.
7. Leia um vetor com 20 elementos. A seguir, troque o primeiro elemento com o último, o segundo com o penúltimo, etc, até o décimo com o primeiro.
8. Faça um programa que leia um vetor de n elementos inteiros, calcule e mostre:
 - (a) A quantidade de números pares;

- (b) Quais os números pares;
 - (c) A quantidade de números ímpares;
 - (d) Quais os números ímpares.
9. Dado um vetor de n elementos, faça um programa que leia um número inteiro x e remova-o do vetor. Se houver mais de uma ocorrência, seu programa deve remover todas as ocorrências. Se o elemento não estiver presente, imprima uma mensagem informando que o elemento não existe.
 10. Dado um vetor de n elementos inteiros, calcule a média dos elementos.
 11. Incremente o exercício anterior de forma que seu programa imprima em uma linha os elementos que estão abaixo da média e em outra os que estão acima da média.
 12. Escreva um algoritmo que leia o vetor $K[20]$ de inteiros e escreva 2 vetores V e Q .
 V deve conter os elementos distintos de K .
 Q deve conter a quantidade de vezes que cada valor de V ocorre em K .
 13. Escreva um algoritmo que leia um vetor $X[20]$ de reais e retorne o seu maior e o seu menor valor.
 14. Idem ao exercício anterior. Imprima a posição do maior e do menor elemento do vetor X .
 15. Escreva um algoritmo que leia um vetor $X[30]$ e retorne um vetor K , cujos elementos são os valores de X multiplicados pelo seu índice.
 16. Uma escola deseja saber se existem alunos cursando, simultaneamente, as disciplinas de Algoritmos e Lógica de Programação. Coloque os números das matrículas dos alunos que cursam Algoritmos em um vetor, no máximo de quinze alunos. Coloque os números das matrículas dos alunos que cursam Linguagem de Programação em outro vetor, no máximo dez alunos. Mostre o número da matrícula dos alunos que estão matriculados nas duas disciplinas.
 17. Faça um programa que receba o nome de n produtos e seus respectivos preços, calcule e mostre:
 - (a) A quantidade de produtos com preço inferior a R\$ 50,00.
 - (b) O nome dos produtos com preço entre R\$ 50,00 e R\$ 100,00.
 - (c) A média dos preços dos produtos com preços superior a R\$ 100,00.
 18. Faça a leitura de 15 números e armazene-os em um vetor. Depois gere dois vetores: um com os números em ordem crescente e outro com os números em ordem decrescente. Por fim, mostre o conteúdo desses dois vetores.

19. Um vendedor precisa de um programa que leia o valor das vendas mensais e mostre quais as três maiores vendas e os dias que as mesmas foram realizadas. Imprima o valor das vendas ordenadas de forma crescente pela data.
20. Faça um programa que leia dois vetores de 10 elementos numéricos cada um e mostre um vetor resultante da intercalação desses dois vetores.

Exemplo:

V1: 1, 9, 8, 7, 6, 5, 3, 4, 2, 0;

V2: 0, 1, 2, 3, 5, 6, 4, 7, 9, 8;

Resultante: 1, 0, 9, 1, 8, 2, 7, 3, 6, 5, 5, 6, 3, 4, 4, 7, 2, 9, 0, 8;

21. Suponha que a Tabela 5.1 represente a memória do computador no momento da execução de um algoritmo.

teste [4]	4
teste [3]	6
teste [2]	8
teste [1]	9
teste [0]	11
i	1
j	10

Tabela 5.1: Exemplo de valores na memória do computador

Baseado na Tabela 5.1, responda:

- (a) O que apareceria na tela se a seguinte instrução fosse executada?

```
i = i + 2;
cout << "O valor que está na posição " << i << "é: " << teste[i];
cout << "\nValor a ser impresso: " << j+1;
```

- (b) Considerando a memória apresentada na Tabela 5.1, como ficaria a memória após a execução do seguinte trecho de código?

```
i = 0;
while(i < 5)
{
    teste[i] = teste[i] * j;
    i = i + 1;
}
```

- (c) Considerando a memória apresentada na Tabela 1.1, como ficaria a memória após a execução do seguinte trecho de código?

```
for(i = 0; i < 5; i++)
{
    teste[i] = teste[i] * j;
    i = i + 1;
}
```

22. Professor Mario anda muito a tarefa e por isso está sem tempo para corrigir provas. Você é um de seus alunos. Como o professor sabe que você possui conhecimento em programação de computadores, pediu sua ajuda. Ele quer que você faça um programa para corrigir provas de múltipla escolha. Cada prova tem 10 questões e cada questão vale 1 ponto. O primeiro conjunto de dados a ser lido é o gabarito da prova. Os outros dados serão os números dos alunos e suas respectivas respostas. Existem 5 alunos matriculados. Calcule e mostre:

- (a) O número e a nota de cada aluno.
- (b) A percentagem de aprovação, sabendo-se que a nota mínima é 6,0.

23. Em uma fábrica trabalham homens e mulheres divididos em três classes:

- Trabalhadores que fazem até 30 peças por mês – classe 1;
- Trabalhadores que fazem de 31 a 35 peças por mês – classe 2;
- Trabalhadores que fazem mais de 35 peças por mês – classe 3;

A classe 1 recebe salário mínimo (R\$465,00). A classe 2 recebe salário mínimo mais 3% do salário mínimo por cada peça acima das 31 peças iniciais. A classe 3 recebe salário mínimo mais 5% do salário mínimo por cada peça fabricada acima das 35 peças iniciais. A fábrica possui 15 operários. Faça um programa que leia, para cada operário: o seu número (inteiro), o número de peças fabricadas no mês e seu sexo (1 para masculino ou 2 para feminino). Os dados devem ser armazenados em 3 vetores: `vet_num_op`, `vet_num_pecas` e `vet_sexo` respectivamente. O programa deve calcular os salários dos funcionários, armazená-los em um quarto vetor (`vet_salarios`) e exibir um relatório que contenha o número do operário, a quantidade de peças fabricadas no mês e o seu salário. O programa deve mostrar também o total da folha de pagamento da fábrica.

24. Faça um programa que leia o preço de compra e o preço de venda de cem mercadorias. O algoritmo deverá imprimir quantas mercadorias proporcionam:

- (a) Lucro menor do que 10%.
- (b) Lucro entre 10% e 20%.

(c) Lucro maior que 20%.

25. Faça um programa que leia dois conjuntos de números inteiros, tendo cada um 10 e 20 elementos. Depois imprima os elementos que não são comuns aos dois conjuntos.
26. Dados dois vetores, um contendo a quantidade e o outro o preço de 20 produtos, elabore um algoritmo que calcule e exiba o faturamento que é igual a quantidade \times preço. Calcule e exiba também o faturamento total que é o somatório de todos os faturamentos, a média dos faturamentos e quantos faturamentos estão abaixo da média.
27. Um jogador viciado de cassino deseja fazer um levantamento estatístico simples sobre uma roleta. Para isso, ele fez n lançamentos nesta roleta. Sabendo que uma roleta contém 37 números (de 0 a 36), calcular a frequência de cada número desta roleta nos n lançamentos realizados.

Capítulo 6

Lista de Exercícios - Vetores de Strings

Esta lista de exercícios é composta por exercícios teóricos e práticos que abordam os conceitos de manipulação de vetores de string da Linguagem C.

1. Faça uma pesquisa sobre as funções da biblioteca `string.h`. Informe o nome, os argumentos de entrada (nome e tipo) e o tipo de retorno. Além disso, faça um breve resumo sobre a funcionalidade de cada função.
2. Qual valor é retornado na chamada da seguinte função:
`strcmp("Casa", "carro")`.
3. Qual o tamanho do menor vetor que devemos declarar para armazenar uma cadeia de caracteres com 10 letras?
4. Faça um programa que leia uma palavra informada pelo usuário em seguida seu programa deve imprimir o número de caracteres que há na palavra.
5. Faça um programa que receba uma frase e conte as vogais presentes nesta frase apresentando a seguinte saída:

Exemplo:

Frase: "Na próxima quarta-feira é feriado."

a : ***** (6)

e : *** (3)

i : *** (3)

o : ** (2)

u : * (1)

6. Faça um programa para concatenar duas strings que devem ser informadas pelo usuário. A segunda string deve ser concatenada na primeira. Seu programa deve imprimir

também o tamanho de cada string. (OBS: Seu programa deve realizar a tarefa acima citada sem o uso de funções da biblioteca string.h)

7. Fazer um programa para ler uma frase e ver quantas vezes um determinado caracter aparece na frase. Esse caracter deve ser informado pelo usuário.
8. Dadas duas cadeias (uma contendo uma frase e outra contendo uma palavra), determine o número de vezes que a palavra ocorre na frase. Para exemplificar considere a palavra “ANA” e a frase: ANA e MARIANA GOSTAM DE BANANA, a palavra ANA ocorre 4 vezes.
9. Fazer um programa para ler uma string e um caracter. Sempre que o caracter lido aparecer na frase ele deve ser substituído por asterisco.
Exemplo:
Frase: o dia esta nublado
Caracter: d
Resultado: o *ia esta nubla*o
10. Fazer um programa para ler uma frase e contar quantas palavras existem na frase.
11. Ao ser fornecido uma frase, uma posição na frase e uma palavra, insira a palavra na posição indicada sem perder o conteúdo da frase.
Exemplo:
Frase: o dia uente
Posição: 6
Palavra: esta q
Resultado: o dia esta quente
12. Ao serem fornecidas duas cadeias, gerar e exibir a intercalação das palavras contidas nas cadeias em uma terceira cadeia.
Exemplo:
Frase1: Em de espeto de
Frase2: casa ferreiro é pau
Frase3: Em casa de ferreiro espeto é de pau
13. Fazer um programa para contar quantas consoantes existem em um frase informada pelo usuário.
14. Faça um programa que receba uma string e a imprima-a escrita de trás pra frente.
15. Faça um programa que leia uma string e diga se ela é palíndrome. Uma string é palíndrome quando pode ser lida tanto de trás pra frente quanto de frente para trás e possui exatamente a mesma sequência de caracteres. Ex.: ASA, SUBI NO ONIBUS, ARARA. Desconsidere os espaços.

16. Faça um programa que conte quantas letras maiúsculas existem numa string informada pelo usuário.
17. Faça um programa que leia duas cadeias de caracteres (string) e determine:
 - (a) Se as mesmas são iguais ou não.
 - (b) Caso as palavras sejam diferentes, qual delas tem maior comprimento (não esquecer a possibilidade de existirem palavras diferentes de mesmo tamanho).
 - (c) Verifique se a segunda palavra é uma substring da primeira ou vice-versa.

Exemplo de entrada:

Palavra 1= casamento

Palavra 2 = casa

Exemplo de saída:

As palavras são diferentes;

A primeira palavra é maior que a segunda.

A segunda palavra é uma substring da primeira

18. Faça um programa que leia uma cadeia de caracteres (string) e tenha como resultado uma terceira cadeia, correspondente à concatenação das duas cadeias. Não é permitido o uso de nenhuma função da biblioteca string.h. A operação de concatenação das cadeias ENGENHARIA e ALGORITMOS é ENGENHARIAALGORITMOS.
19. Faça um programa que dado um nome completo retorne a abreviatura deste nome. Não se devem abreviar as preposições como: do, de, etc. A abreviatura deve vir separada por pontos. Ex: Paulo Jose de Almeida Prado. Abreviatura: P.J.A.P.
20. Faça um programa para cadastro e diálogo de login. O programa deve:
 - (a) Cadastrar um nome de usuário via teclado. O nome de usuário tem, no máximo, 8 caracteres, sendo válidos somente os caracteres numéricos e as letras maiúsculas ou minúsculas. Somente os caracteres válidos devem ser exibido durante a digitação do nome de usuário.
 - (b) Cadastrar uma senha do usuário via teclado. Esta segue as mesmas regras do nome de usuário, com a diferença de que são exibidos somente asteriscos à medida que a senha é digitada.
 - (c) Receber um novo nome de usuário e uma nova senha, utilizando os mesmos procedimentos descritos nos itens a e b.
 - (d) Comparar o nome de usuário cadastrado com o recebido posteriormente e a senha cadastrada com a senha recebida. Caso sejam idênticos, informar “OK”, do contrário informar “Acesso negado”.

21. Faça um programa para receber uma string informada pelo usuário de no máximo 50 caracteres e fazer uma estatística dos caracteres digitados. Por exemplo, para a string “O EXERCICIO E FACIL”, a estatística mostrada será 'O' = 2, 'E' = 3, 'X' = 1, 'R' = 1, 'C' = 3, 'I' = 3, 'F' = 1, 'A' = 1, 'L' = 1. (OBS: Pode ser desconsiderados os acentos das palavras).
22. Observe atentamente o código abaixo. Explique por que as strings st1 e st2 devem ter no máximo 10 caracteres se os respectivos vetores foram declarados com 11 posições. Explique também o funcionamento do código.

```
#include <string.h>
#include <iostream.h>
main( )
{
    char st1[11], st2[11], st3[21], ch;
    int i, j;

    cout << "\n Digite a primeira string com até 10 caracteres ";
    gets(st1);

    cout << "\n Digite a segunda string com até 10 caracteres ";
    gets(st2);

    for (i = 0; st1[i]; i++)
        st3[i] = st1[i];

    for (j = 0; st2[j]; j++)
        st3[i+j] = st2[j];

    j = j+i;

    for (i = j; i ; i--)
    {
        ch = st3[i-1];
        cout << ch;
    }
    cout << "\n\n";
}
```

23. Faça um programa que receba uma string e faça com que a primeira letra de cada palavra fique em maiúsculo. Para isso, basta subtrair 32 do elemento que deseja alterar para maiúsculo.

```
chrNome[0] = chrNome[0] - 32;
```

Exemplo:

Entrada: lab. de linguagem de programação.

Saída: Lab. De Linguagem De Programação.

24. Escreva um programa em C que recebe 5 cadeias de caracteres de tamanho máximo 50, e as imprima na tela em ordem alfabética.

Capítulo 7

Lista de Exercícios - Matrizes

Esta lista de exercícios é composta por exercícios teóricos e práticos que abordam os conceitos de manipulação de matrizes da Linguagem C.

1. Qual das alternativas a seguir corresponde à declaração de uma matriz:
(a) `() int M;`
(b) `() char X[5];`
(c) `() double VET[10][10];`
(d) `() float MATRIZ[10];`
2. Declarada a matriz $M_{2 \times 3}$, quais os valores que podem ser armazenados na matriz:
`int M[2][3];`
(a) `() 1.2; 4.7; 5.6; 20.4; 25; 89.12;`
(b) `() 12; 7; 1; 3; 9; 6;`
3. Faça um programa que receba uma matriz $M_{5 \times 5}$ e retorne a soma dos seus elementos.
4. Escrever um algoritmo para armazenar valores numéricos em uma matriz $M_{5 \times 6}$. A seguir, calcular a média dos valores pares contidos na matriz e escrever seu conteúdo.
5. Escrever um algoritmo para ler uma matriz $A_{7 \times 4}$ contendo valores numéricos distintos (seu programa deve validar essa condição). Depois encontre o menor valor contido na matriz e sua posição.
6. Gerar a matriz transposta de uma matriz $B_{5 \times 5}$ dada pelo usuário (a transposta é obtida permutando-se as linhas e as colunas de uma matriz).
7. Dado um valor numérico x e uma matriz $A_{3 \times 4}$ elabore um algoritmo que calcule e exiba uma outra matriz B que deverá conter cada elemento da matriz A dividido pelo valor numérico x .

8. Escreva um programa que leia uma matriz de números reais de 4 linhas e 4 colunas. Em seguida o programa deve mostrar:

- (a) Os elementos da diagonal principal;
- (b) Os elementos da matriz triangular inferior;
- (c) Os elementos da matriz triangular superior;

9. Escreva um algoritmo que lê uma matriz $M_{5 \times 5}$ e calcula as somas:

- (a) Da linha 4 de M .
- (b) Da coluna 2 de M .
- (c) Da diagonal principal.
- (d) Da diagonal secundária.
- (e) O produto dos elementos das colunas ímpares.

Mostre ao usuário todos os resultados das operações listadas acima.

10. Faça um programa que receba:

- (a) As notas de 15 alunos em cinco provas diferentes e armazene-as em uma matriz de ordem 15×5 ;
- (b) Armazene os nomes dos 15 alunos e armazene-os em um vetor.
- (c) Mostre para cada aluno, o nome, a média aritmética das cinco provas e a situação (Aprovado, Reprovado ou Exame) e a média da classe.

11. Faça um programa que leia duas matrizes quadradas de ordem 3 de números inteiros; Calcule a soma das duas matrizes e armazene em uma terceira matriz. Em seguida mostre a terceira matriz da seguinte maneira:

Na primeira vez os elementos devem ser mostrados em sua ordem natural.

Na segunda, os elementos devem aparecer em ordem crescente.

Veja os exemplos a seguir:

1ª Impressão:

14	7	89
16	21	4
39	63	32

2ª Impressão:

4	7	14
16	21	32
39	63	63

12. Escrever um algoritmo que lê uma matriz $M_{5 \times 5}$ e cria 2 vetores $SL[5]$, $SC[5]$ que contenham respectivamente as somas das linhas e das colunas de M . Imprima a matriz e os vetores criados.

13. Escrever um programa que leia duas matrizes de ordem 4×6 e cria:

- (a) Uma matriz $M1$ que seja a soma das duas matrizes;

(b) Uma matriz $M2$ que seja a diferença das duas matrizes;

Depois das operações mostre na tela as matrizes obtidas;

14. Dado uma matriz de ordem 3, faça um programa que verifique se a matriz é simétrica ($A_{ij} = A_{ji}$).
15. Dado uma matriz com números reais de ordem 4, faça um programa que leia estes valores e ao final da leitura, imprima os seguintes relatórios:
- (a) A soma de cada coluna da matriz;
 - (b) Os valores que são menores que a média dos valores;
16. Dada uma matriz real A com m linhas e n colunas e um vetor real V com n elementos, determinar o produto de A por V .
17. Dada uma matriz quadrada P , de ordem 4:
- (a) Some os quadrados dos elementos da coluna 1;
 - (b) Some os elementos da linha 2;
 - (c) Some os elementos ímpares da matriz;
 - (d) Some os elementos das linhas pares da matriz;
 - (e) Some os elementos de índice ímpar da linha 3.
18. Dizemos que uma matriz inteira $A_{n \times n}$ 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.
Exemplo: A matriz abaixo é de permutação:

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Observe que:

$$\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Não é permutação.

Dada uma matriz $A_{n \times n}$ verifique se A é ou não uma permutação.

19. Dada uma matriz $A_{m \times n}$, imprimir o número de linhas e o número de colunas nulas da matriz.

Exemplo: $m = 4$ e $n = 4$

$$\begin{pmatrix} 1 & 0 & 2 & 3 \\ 4 & 0 & 5 & 6 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Saída: tem 2 linhas nulas e 1 coluna nula.

20. Dizemos que uma matriz quadrada inteira é um quadrado mágico se a soma dos elementos de cada linha, a soma dos elementos de cada coluna e a soma dos elementos das diagonais principal e secundária são todas iguais.

Exemplo: A matriz

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

Saída: A matriz é um quadrado mágico.

Dada uma matriz quadrada $A_{n \times n}$ verificar se A é um quadrado mágico.

21. Os elementos A_{ij} de uma matriz de inteiros $A_{n \times m}$ representam os custo de transporte da cidade i para a cidade j . Dado n itinerários, cada um com k cidades, calcule o custo total para cada itinerário.

Exemplo:

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

O custo do itinerário 0 3 1 3 3 2 1 0 é

$$A_{03} + A_{31} + A_{13} + A_{33} + A_{32} + A_{21} + A_{10} = 3 + 1 + 400 + 5 + 2 + 1 + 5 = 417$$

22. Faça um programa que leia elementos de uma matriz quadrada de ordem n e armazene o elemento somente se A_{ij} é maior que todos os elementos anteriores, caso seja menor imprima um aviso na tela e peça outro número ao usuário.
23. Faça um programa que leia uma matriz de ordem 3×3 e imprima a sua resultante. Suponha duas matrizes $A_{a \times b}$ e $B_{m \times n}$.

- $A * B$ só é possível se e somente se $b = m$. A matriz resultante seria do tipo $R_{b \times n}$.
- $B * A$ só é possível se e somente se $n = a$. A matriz resultante seria do tipo $R_{a \times n}$.

Exemplo:

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

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

$A * B$ é possível.

$B * A$ não é possível. A matriz resultante é do tipo $R_{2 \times 2}$.

24. Na teoria de Sistemas define-se elemento mínimax de uma matriz, o menor elemento da linha em que se encontra o maior elemento da matriz. Faça um programa que leia uma matriz $A_{10 \times 10}$ e determina o elemento mínimax desta matriz, imprima a linha e coluna que o elemento se encontra, o elemento e o maior elemento da matriz.

25. Dadas duas matrizes inteiras A e B de ordem 4×3 , fazer um programa que gere uma matriz booleana C , tal que o elemento C_{ij} seja V se os elementos nas posições respectivas das matrizes A e B forem iguais, e F caso contrário. Exibir as matrizes A , B e C .

Exemplo:

$$A = \begin{bmatrix} 2 & 4 & 6 \\ 1 & 5 & 9 \\ 3 & 7 & 2 \\ 4 & 6 & 8 \end{bmatrix}$$

$$B = \begin{bmatrix} 2 & 5 & 8 \\ 1 & 9 & 7 \\ 3 & 7 & 1 \\ 4 & 5 & 8 \end{bmatrix}$$

$$C = \begin{bmatrix} V & F & F \\ V & F & F \\ V & V & F \\ V & F & V \end{bmatrix}$$

Capítulo 8

Lista de Exercícios - Structs

Esta lista de exercícios é composta por exercícios teóricos e práticos que abordam os conceitos de manipulação de structs da Linguagem C.

1. Assinale na alternativa que corresponde à forma correta de uma estrutura:

(a) ()

```
struct
{
    int a;
    char b[5];
}
```

(b) ()

```
struct 123P
{
    int a;
    char b[5];
}
```

(c) ()

```
struct P123
{
    int a;
    char b[5];
};
```

2. Escreva um programa que preencha uma variável do tipo struct e depois exiba na tela. A estrutura deverá conter campos para: nome, endereço, idade, telefone, data. Sendo que data deverá ser uma estrutura com os campos: dia, mês e ano.
3. Imagine a seguinte situação: você precisa armazenar na memória os dados referentes a 60 alunos nome, série e suas notas ao longo do ano (4 bimestres) em 4 disciplinas, inglês, francês, matemática e português. Crie uma estrutura para resolver o problema acima. Por fim, imprima o nomes do melhor aluno de cada matéria, ou seja, os alunos com maiores notas nas respectivas matérias.
4. Faça um programa que leia o nome e o sexo de n pessoas. Crie uma estrutura que armazene todas essas informações. Esta estrutura deve ter um campo para data de nascimento, a geração da data de nascimento deve ser feita aleatoriamente através do trecho de código abaixo:

```
D.Mes = 1 + (rand() \% 12);  
D.Ano = 1950 + (rand() \% 49);  
D.Dia = 1 + (rand() \% 30);
```

O programa deve conter um menu com as seguintes opções:

- Inserir informações.
- Listar todos os nomes e respectivas idades.
- Listar os nomes das pessoas mais velhas do que uma certa idade fornecida.*

*Na opção 3, após entrar neste campo, perguntar a idade para listar os nomes;

5. Uma Loja de móveis deseja cadastrar seus clientes, para isso você foi contratado para fazer um programa que auxilie nessa tarefa. Faça um programa que leia o nome, idade, renda, endereço, cidade, estado e CEP de um cliente e informe seu crédito na loja (50% da renda). O programa deve finalizar quando for digitado “FIM” no nome do cliente .
6. O Banco Satolepense S/A necessita de um programa capaz de ler e armazenar os dados de n clientes “Vips”. Os dados são: nome, sobrenome, data de abertura da conta, valor do depósito mais recente e saldo. Suponha que os dados referente a um cliente são lidos somente uma vez, no início da execução. Após a leitura, o programa deve imprimir:
 - (a) Todos os dados do cliente mais antigo (ou seja, cliente com data de abertura de conta mais antiga), juntamente com a mensagem “cliente mais antigo”.
 - (b) Lista dos clientes (no formato: nome, sobrenome, saldo) com saldo maior que um dado valor, este valor deve ser fornecido pelo usuário, juntamente com a mensagem “clientes com saldo maior que *valor*”.

Defina uma variável do tipo struct para armazenar os dados dos clientes Vips.

7. Uma determinada fábrica paga seus funcionários por horas trabalhadas e faz esse controle através de um sistema de cartão de ponto eletrônico. Ao passar o crachá no aparelho pela segunda vez no mesmo dia, na saída do expediente, o sistema grava um registro com o número da matrícula do funcionário e a quantidade de horas trabalhadas naquele dia. Os registros vão sendo acumulados dia a dia, ao longo de um período. O valor atual pago por hora é R\$13,66 e os números de matrículas variam de 1 a 1000. O tipo abaixo define o registro que deve ser produzido pelo aparelho:

```
struct REG_PONTO
{
    int matricula;
    float horas;
};
```

O programa deve ter um menu contendo:

- Ponto.
 - Ver as horas trabalhadas por um dado funcionário.
 - Ver o valor que um dado funcionário tem a receber no momento.
 - Sair.
8. Faça um programa que cadastre os dados da folha de pagamento. A ficha do funcionário contém matrícula, nome, salário, data de admissão. Leia os dados de n funcionários e dê o relatório no final:
- Lista de funcionários cadastrados (matrícula, nome, salário e data admissão).
 - Total da folha (soma dos salários).
 - O salário médio.
 - O menor salário.
 - A matrícula, o nome e a data de admissão do funcionário que tem o maior salário.
9. Num determinado dia foi feita uma pesquisa de audiência de TV em várias casas de uma certa cidade. Para cada casa visitada, o entrevistador (munido de um notebook) questionava o número de aparelhos de televisão existentes e, para cada aparelho questionava o número de pessoas que estavam assistindo tal TV, solicitando também que se informasse qual emissora de televisão estava sendo assistido dentre as opções (Cultura, SBT, Globo, Record, MTV, Futura, RedeTV, Band ou nenhum). Se o TV estivesse desligado, então nenhuma pessoa estava utilizando tal aparelho e nada era anotado. Implementar uma versão do programa sendo usado pelo entrevistador de modo que:
- Sejam lidos um número indeterminado de dados, terminando a pesquisa quando o entrevistador escolher a opção para FIM.

- Seja calculada e exibida o número de lares, TVs e pessoas pesquisadas, indicando o número médio de TV por lar e o número médio de pessoas por TV.
- Determine o número de espectadores assistindo cada emissora e a porcentagem de audiência para cada emissora.

Identifique as emissoras com maior e menor audiência na pesquisa.

- Um cinema que possui capacidade de 100 lugares está sempre com ocupação total. Certo dia cada espectador respondeu a um questionário, no qual constava: sua idade, seu sexo e sua opinião em relação ao filme (que podia ser ótimo, bom, regular, ruim ou péssimo). Elabore um programa que, lendo estes dados, responda:
 - A quantidade de respostas de cada opinião.
 - A diferença percentual entre respostas regular e bom.
 - A média das idades das pessoas que responderam ruim.
 - A média das idades das mulheres que responderam ótimo.
 - A porcentagem de respostas péssimo e a maior idade que utilizou esta opção.
- Um colecionador deseja organizar sua coleção de CDs. Este colecionador precisa de alguém que faça um programa que o ajude nessa tarefa, você foi o escolhido. Crie uma estrutura que contenha as informações sobre um CD de música, sendo as informações:
 - Nome da banda.
 - Dia do lançamento do CD.
 - Mês do lançamento do CD.
 - Ano do lançamento do CD.
 - Valor do CD.
 - Número de membros da banda.
 - Produtora do CD.
 - Gênero.

Seu programa deve facilitar ao máximo a utilização do usuário, fornecendo um menu onde ele possa obter a informações desejadas facilmente. Abaixo segue um exemplo de como você pode formar sua estrutura.

```

Struct Banda
{
    char nome[10];
    int dia;
    int mês;
    int ano;
    float valor;
    int membros;
    char produtora[15];
};

```

12. Uma pesquisa sobre algumas características físicas da população de uma determinada região coletou os seguintes dados, referentes a cada habitante, para serem analisados:

- Sexo (masculino, feminino)
- Cor dos olhos (azuis, verdes, castanhos)
- Cor dos cabelos (louros, castanhos, pretos)
- Idade em anos.

Os dados de cada habitante serão fornecidos em linhas, sendo que um valor de idade negativo indica o fim da leitura. O programa a ser implementado deverá fornecer as seguintes informações:

- (a) O número de homens de olhos verdes, cabelos castanhos e idade entre 30 e 40 anos.
- (b) O número de mulheres de olhos azuis, cabelos louros e idade entre 20 e 30 anos.
- (c) A porcentagem de crianças de ambos os sexos, olhos castanhos, cabelos castanhos ou pretos e idade inferior a 14 anos.

13. Faça um programa que leia as notas, nomes e disciplinas de n acadêmicos de uma universidade. Esta universidade usa o seguinte sistema de médias:

- Média $\geq 7,5 \rightarrow$ Aprovado.
- Média $\leq 7,5$ e média $> 4,5 \rightarrow$ Exame.
- Média $\leq 4,5 \rightarrow$ Reprovado.

Seu programa deve calcular a média de cada acadêmico. Em seguida, seu programa deve informar o número de acadêmicos aprovados, de exame e reprovados e seus respectivos nomes. Deve mostrar ainda, a disciplina com o maior índice de aprovação, caso haja empate mostre as que possuem o mesmo índice.

14. Foi realizada uma pesquisa entre 500 habitantes de uma certa região. De cada habitante foram coletados os dados: idade, sexo, salário e número de filhos. Crie a estrutura de dados adequada para armazenar estas informações e armazene as informações digitadas pelo usuário na estrutura de dados criada. Calcule a média do salário dos habitantes, a média de filhos dos homens, qual o sexo da pessoa com o maior salário e a idade e o número de filhos da pessoa com o menor salário.

15. Faça um programa que tenha as seguintes especificações:

(a) Uma estrutura que contenha os campos:

- Marca.
- Ano.
- Número Série motor.
- Cor.
- Preço.
- Placa.

OBS. A placa deve conter 3 letras e 4 números

(b) Uma tabela de impostos

- Para carros entre 1970 e 1980 $\rightarrow 0,8\%$ sobre o valor do carro.
- Para carros entre 1980 e 1990 $\rightarrow 0,9\%$ sobre o valor do carro.
- Para carros entre 1990 e 2000 $\rightarrow 1\%$ sobre o valor do carro.
- Para carros acima de 2000 $\rightarrow 1,5\%$ sobre o valor do carro.

Seu programa deve receber as informações de n carros. Seu programa deve calcular o imposto para cada veículo e acrescentar ao preço do carro. Em seguida imprima as placas, a cor, número de série do motor e a marca dos cinco carros mais caros. Imprima também o ano do carro mais antigo e seu preço.

16. Um programador precisa fazer um programa que cadastre as fitas de vídeo (VHS) e os DVDs de uma locadora de filmes. Devido ao tamanho das prateleiras, a locadora pode ter no máximo 1000 itens que contêm as seguintes informações cada:

- Número de Série (0 a 999).
- Tipo (VHS ou DVD).
- Título do Filme.
- Tipo de Filme (Infantil, Adulto, romance, etc.).
- Situação: disponível, alugado, extraviado.

Considerando que a informações de cada filme será armazenada em um vetor indexado pelo Número de Série, faça um programa que simule o funcionamento desta locadora.

17. Deseja-se fazer a emissão da folha de pagamento de uma empresa. Para cada um dos funcionários são dadas as seguintes informações:

- Nome do funcionário.
- Salário base do funcionário.
- Horas extras diurnas (hed).
- Horas extras noturnas (hen).
- Número de dependentes.
- Número de faltas em horas.
- Valor de vales retirados no mês.

A empresa possui n ($n \leq 100$) funcionários, que devem ser lidos inicialmente. Após ler estas informações, imprima o nome, salário, valor em horas extras, salário família, salário bruto, valor de desconto do INSS, faltas no mês (em horas), valor dos vales, imposto de renda e o salário líquido sabendo que:

- Valor em horas extras = $(hed * \text{salário base} + hen * 1.2 * \text{salário base}) / 160$.
- Salário família = Número de dependentes * 0.05 * salário base.
- Salário bruto = salário base + salário família + valor em horas extras.
- INSS = $0.08 * \text{salário bruto}$.
- Valor em faltas = número de faltas * salário base / 160.
- Imposto de renda = $0.13 * \text{salário bruto}$.
- Salário líquido = salário bruto - INSS - valor em faltas - imposto de renda.

18. Uma empresa de vendas de produtos de informática deseja implantar um sistema informatizado para controle de estoque. Neste sentido, faça um programa que execute as seguintes tarefas:

(a) Cadastrar os n ($n \leq 1000$) produtos da loja, onde cada produto possui as seguintes informações:

- Código do produto.
- Quantidade disponível em estoque.
- Preço Unitário.
- Nome do Produto.
- Marca do Produto.

Seu programa deve cadastrar os produtos conforme solicitado acima.

(b) Ler um pedido de compras com as seguintes informações:

- Código do Produto Desejado.
- Quantidade Solicitada.
- Nome do Comprador.

Seu programa deve ler essas informações e verificar se os pedidos podem ser atendidos.

(c) Verificar se existe no Catálogo de Produtos o produto especificado no Pedido de Compra, executando as seguintes tarefas:

- Se o produto não existir, emitir uma mensagem informando para o usuário.
- Se o produto existir em quantidade suficiente, atualizar o estoque da venda e emitir uma mensagem com as informações da venda:
 - Nome do comprador.
 - Código do Produto.
 - Nome do Produto.
 - Quantidade Solicitada.
 - Preço Unitário.
 - Preço Total da Compra.
- Se o produto não existir em quantidade suficiente, emitir uma mensagem. O programa deve retornar -1 se o produto não existir, 0 se o produto existir, mas não em quantidade suficiente e 1 caso a venda possa ser realizada sem problemas.
- Atualizar o estoque de produtos, inserindo mais unidades dos produtos já cadastrados. Para isso é necessário ler:
 - Código do Produto.
 - Quantidade a estocar.

Se o Código do produto informado for inexistente, deve ser emitida uma mensagem para o usuário. Caso contrário, deve ser atualizada a quantidade de produtos do estoque.

(d) Seu programa deve ter o seguinte menu:

- 1 - Cadastrar um novo produto.
- 2 - Solicitar Produtos.
- 3 - Atualizar Estoque.
- 4 - Sair.

Capítulo 9

Lista de exercícios 9 - Olimpíada Brasileira de Informática

Esta lista de exercícios é composta por exercícios que abordam todo o conteúdo ministrado nas aulas do Curso preparatório para a Olimpíada Brasileira de Informática.

Auto Estrada

Arquivo fonte: *auto.c*, *auto.cc*, *auto.cpp*, *auto.pas*

Certas regiões resolveram o problema de tráfego intenso com a construção de auto estradas, que são estradas contendo em geral quatro ou mais pistas de rolagem em cada sentido, de forma que um número grande de carros possa passar sem que ocorram congestionamentos. O problema das auto estradas é que, junto com os carros temos um aumento considerável de ruídos nas imediações da pista, o que incomoda os moradores das regiões próximas. A GoTo engenharia, uma empresa do ramo de construção especializada em obras de estradas, encontrou uma solução engenhosa para o problema: instalar grandes painéis defletores de som de cada lado da auto estrada para tentar minimizar o ruído percebido pelos vizinhos. Os painéis são construídos em blocos contínuos de 10 metros lineares. A auto estrada também é dividida em blocos de 10 metros de extensão, sendo cada bloco descrito por um código, como definido abaixo:

- P - Pista, trecho em linha reta sem curvas ou saídas. Deve-se instalar um painel de cada lado da auto estrada.
- C - Curva, trecho em curva de 90 graus na auto estrada. Deve-se instalar dois painéis de concreto do lado externo da curva; o outro lado fica sem painel instalado.
- A - Acesso, trecho em linha reta no qual existe uma entrada ou uma saída partir de um dos lados da auto estrada (mas não do outro). Deve-se instalar um painel no lado onde não existe o acesso.

- D - Duplo acesso, trecho em linha reta no qual existem dois acessos (entradas ou saídas, em qualquer combinação possível, um de cada lado da rodovia. Nenhum painel deve ser instalado nesse bloco da auto estrada.

Tarefa

Apesar de ser uma empresa formada por engenheiros, nenhum dos funcionários da GoTo sabe programar, de forma que eles decidiram contrataram você como consultor independente. Você deve escrever um programa para, dado um mapa da auto estrada, determinar quantos painéis defletores são necessários para cobrir toda a extensão dessa auto estrada.

Entrada

A entrada é composta de vários casos de Teste, que deve ser lido do dispositivo de entrada padrão (normalmente o teclado). A primeira linha contém um inteiro C , indicando o comprimento da auto estrada, em blocos de 10 metros. A linha seguinte contém C caracteres, cada letra descrevendo um bloco de 10 metros da auto estrada, como definido acima. O final da entrada é indicado por $C = 0$.

Exemplo de entrada

```
5
DAPCD
8
AACCAAPP
0
```

Saída

Para cada Teste, você deverá imprimir um identificador “Teste n ”, onde n é enumerado a partir de 1. Na linha seguinte imprima um número inteiro, representando quantas unidades de painel são necessárias para cobrir toda a extensão da auto estrada. Após cada Teste imprima uma linha em branco.

Exemplo da saída

```
Teste 1
5

Teste 2
```

Restrições

 $(1 \leq C \leq 1000000)$ $(C = 0, \text{ somente para indicar o final da entrada })$

Fliperama

Arquivo fonte: *fliperama.c*, *fliperama.cc*, *fliperama.cpp*, *fliperama.pas*

Bebe-bebe é um jogo muito popular de fliperama. E, como a maioria dos jogos de fliperama, ele deve mostrar as maiores pontuações. Para esse fim, a companhia Otori te contratou.

Tarefa

Escreva um programa que, dada a lista de todas as pontuações dos jogos de Bebe-bebe, mostra os melhores placares em ordem decrescente.

Entrada

A entrada é composta de vários casos de Teste. A primeira linha consiste de dois inteiros N e M , dizendo quantas partidas foram jogadas de Bebe-bebe e quantas linhas cabem no mostrador de melhores rankings. As N linhas seguintes contêm cada uma um inteiro indicando a pontuação obtida em cada jogo. O final da entrada é indicado por $N = M = 0$.

Exemplo de entrada

```
7 4
100
200
200
150
30
524
942
2 1
4000
2000
0 0
```

Saída

Para cada Teste, você deverá imprimir um identificador “Teste n ”, onde n é enumerado a partir de 1. Nas M linhas seguintes deve imprimir as M maiores pontuações em ordem decrescente. Após cada Teste imprima uma linha em branco.

Exemplo de saída

Teste 1

942

524

200

200

Teste 2

4000

Restrições

$$1 \leq N \leq 10000$$

$$1 \leq M \leq 500$$

$$M \leq N$$

($M = N = 0$ somente para indicar o final da entrada)

Fatorial

Arquivo fonte: *fatorial.c*, *fatorial.cc*, *fatorial.cpp*, *fatorial.pas*

Joãozinho é um garoto esperto da sexta série. Ele gosta muito de matemática e descobriu que sua professora é muito preguiçosa. Nas provas da matéria a professora pede que as crianças circulem a resposta com um quadrado colorido, e que façam o primeiro dígito diferente de zero (da direita para esquerda) do número especialmente grande com caneta. Joãozinho desconfiou que a professora olhava apenas para aquele dígito para corrigir a questão. A turma aprendeu a calcular o fatorial de um número, e isso será cobrado na próxima prova. Joãozinho está convencido de que não precisa escrever de fato o número correto, desde que o primeiro dígito (olhando da direita para esquerda) seja o correto.

Tarefa

Sua tarefa neste problema é ajudar Joãozinho a calcular para um número inteiro N da entrada, o primeiro dígito (da direita para esquerda) de $N!$ que seja diferente de zero.

Entrada

A entrada é composta de vários conjuntos de Teste. A primeira linha de cada Teste consiste um inteiro N . O final da entrada é indicado por $N = 0$.

Exemplo de entrada

```
5
4
0
```

Saída

Para cada Teste, você deverá imprimir um identificador “Teste n ”, onde n é enumerado a partir de 1. Na linha seguinte imprima o primeiro dígito (da direita para esquerda) diferente de zero. Após cada Teste imprima uma linha em branco.

Exemplo de saída

```
Teste 1
2

Teste 2
```

Restrições

$1 \leq N \leq 20$ ($N = 0$ apenas para indicar o fim da entrada)

Quem vai ser reprovado

Arquivo fonte: *reprovado.c*, *reprovado.cc*, *reprovado.cpp*, *reprovado.pas*

Prof. Wallywow da Universidade da Columbia Britânica está muito preocupado com a queda do nível de atenção de seus estudantes. Ele já tentou várias técnicas mundialmente conhecidas para incentivar os alunos a prestar atenção nas suas aulas e fazer as tarefas que ele passa para a turma: deu nota para os alunos mais participativos, ofereceu chocolates aos alunos, levou seu karaokê e cantava nas aulas etc. Como tais medidas não levaram a uma melhora no comparecimento às aulas (a idéia do karaokê, inclusive, mostrou-se bastante infeliz... na segunda aula com karaokê a turma reduziu-se a um aluno – que tinha problemas auditivos) ele teve uma brilhante idéia: faria uma competição entre os alunos. Prof. Wallywow passou um conjunto de problemas aos alunos, e deu um mês para que eles os resolvessem. No final do mês os alunos mandaram o número de problemas resolvidos corretamente. A promessa do brilhante didata era reprovar sumariamente o último colocado da competição. Os alunos seriam ordenados conforme o número de problemas resolvidos, com empates resolvidos de acordo com a ordem alfabética dos nomes (não há homônimos na turma). Isso fez com que alunos com nomes iniciados nas últimas letras do alfabeto se esforçassem muito nas tarefas, e não compartilhassem suas soluções com colegas (especialmente aqueles cujos nomes começassem com letras anteriores).

Tarefa

Sua tarefa neste problema é escrever um programa que lê os resultados dos alunos do Prof. Wallywow e imprime o nome do aluno reprovado.

Entrada

A entrada é composta de vários conjuntos de Teste. A primeira linha de cada Teste consiste em um inteiro N indicando o número de alunos na competição. Cada uma das N linhas seguintes contém o nome do aluno e o número de problemas resolvidos por ele. O nome consiste em uma seqüência de letras minúsculas [a-z] com no máximo 20 letras e cada time resolve entre 0 à 10 problemas. O final da entrada é indicado por $N = 0$.

Exemplo de entrada

```
4
cardonha 9
infelizreprovado 3
marcel 9
infelizaprovado 3
```

3
alex 5
abel 5
lucas 10
2
flavio 3
junior 4
0

Saída

Para cada Teste, você deverá imprimir um identificador “Teste n”, onde n é numerado a partir de 1. Na linha seguinte imprima o nome do aluno reprovado. Após cada Teste imprima uma linha em branco.

Exemplo de saída

Teste 1
infelizreprovado

Teste 2
alex

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

Restrições

$1 \leq N \leq 100$ ($N = 0$ apenas para indicar o fim da entrada)

Ele está impedido

Arquivo fonte: *impedido.c*, *impedido.cc*, *impedido.cpp*, *impedido.pas*

A Rede do Hemisfério é a maior rede de televisão de Tumbolia, um pequeno país situado a leste da América do Sul. O esporte mais popular em Tumbolia, obviamente, é o futebol; muitos jogos são transmitidos toda semana em Tumbolia. A Rede do Hemisfério recebe muitos pedidos para repassar lances polêmicos; normalmente esses acontecem quando um jogador é dito impedido pelo juiz. Um jogador atacante está impedido se ele está mais próximo da linha do gol do oponente do que o penúltimo adversário. Um jogador não está impedido se:

- ele está na mesma linha que o penúltimo adversário ou
- ele está na mesma linha que os dois últimos adversários.

Tarefa

Através do uso de tecnologia de computação gráfica, a Rede do Hemisfério consegue tirar uma foto do campo e determinar as distâncias dos jogadores até a linha do gol do time defensor, mas eles ainda precisam de um programa que, dadas essas distâncias, decida se um jogador está impedido.

Entrada

A entrada é composta por vários conjuntos de Teste. A primeira linha de cada caso de teste contém dois inteiros A e D separados por um espaço indicando, respectivamente, o número de jogadores atacantes e defensores envolvidos na jogada. A próxima linha contém A inteiros B_i separados por um espaço, indicando as distâncias dos jogadores atacantes até a linha do gol. A próxima linha contém D inteiros C_i separados por um espaço, indicando as distâncias dos defensores até a linha do gol. O final da entrada é dado por $A = D = 0$.

Exemplo de entrada

```
2 3
500 700
700 500 500
2 2
200 400
200 1000
3 4
530 510 490
```

480 470 50 310

0 0

Saída

Para cada Teste, você deverá imprimir um identificador “Teste n”, onde n é numerado a partir de 1. Na linha seguinte imprima o “S” se existe um jogador atacante impedido, e “N” caso contrário. Após cada Teste imprima uma linha em branco.

Exemplo de saída

Teste 1

N

Teste 2

S

Teste 3

N

Restrições

$$2 \leq A \leq D \leq 11$$

$$1 \leq B_i \leq 10000$$

$$1 \leq C_i \leq 10000$$

Recuperação

Arquivo fonte: *recupera.c*, *recupera.cc*, *recupera.cpp*, *recupera.pas*

A nossa grandiosa Professora Cris no último aquecimento ficou conhecida como a grande maquiavélica da FEC(Faculdade de Engenharia e Computação). A digníssima professora exigiu que os alunos formassem uma fila em ordem lexicográfica (pelo nome) com no máximo k permutações. Isto fez com que muitos alunos nem sequer entrassem na sala para fazer a prova. No entanto, nesta seletiva ela resolveu se redimir perante seus alunos, e resolveu aplicar um probleminha para recuperação.

Tarefa

Sua tarefa, mesmo não tendo sido reprovado, é dado uma sequência de N inteiros a_1, a_2, \dots, a_n , onde $-30 \leq a_j \leq 30$ para $j = 1, 2, \dots, n$, imprima, se existir, um inteiro a_k tal que $a_k = a_1 + a_2 + \dots + a_{k-1}$. Se houver mais de um inteiro que satisfaça esta condição, imprima o que aparece primeiro na sequência.

Entrada

A entrada é composta de vários conjuntos de Teste. A primeira linha de cada Teste consiste em um inteiro N indicando o número de inteiros da linha seguinte devem ser processados. O final da entrada é indicado por $N = 0$.

Exemplo de entrada

```
1
0
7
1 2 3 4 5 6 7
3
2 4 5
0
```

Saída

Para cada Teste, você deverá imprimir um identificador “Teste n ”, onde n é enumerado a partir de 1. Na linha seguinte imprima o inteiro que satisfaça a restrição descrita acima. Caso não exista tal inteiro imprima *nao achei*. Após cada Teste imprima uma linha em branco.

Exemplo de saída

Teste 1

0

Teste 2

3

Teste 3

nao achei

Restrições

$1 \leq N \leq 100$ ($N = 0$ somente para indicar o fim da entrada)

Dama

Arquivo fonte: *dama.c*, *dama.cc*, *dama.cpp*, *dama.pas*

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 9.1:

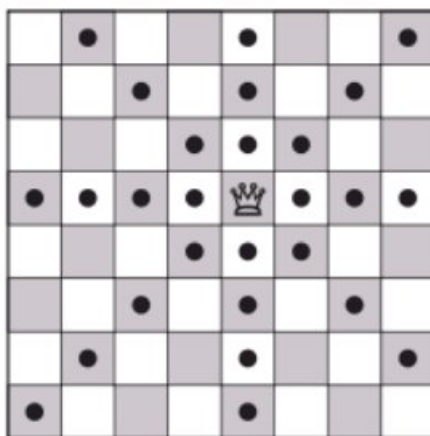


Figura 9.1: Tabuleiro de xadrez.

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 x 8, com 64 casas), de quantos movimentos, no mínimo, ela precisa para chegar em outra casa do tabuleiro?

Tarefa

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 é composta de vários conjuntos de Teste. A primeira e única linha de cada caso de teste contém quatro inteiros X_1, Y_1, X_2 e Y_2 . A dama começa na casa de coordenadas (X_1, Y_1) , e a casa de destino é a casa de coordenadas (X_2, Y_2) . 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 $X_1 = Y_1 = X_2 = Y_2 = 0$.

Exemplo de entrada

4 4 6 2

3 5 3 5
5 5 4 3
0 0 0 0

Saída

Para cada Teste, você deverá imprimir um identificador “Teste n”, onde n é enumerado a partir de 1. Na linha seguinte seu programa deve imprimir um número inteiro, indicando o menor número de movimentos necessários para a dama chegar em sua casa de destino. Após cada Teste imprima uma linha em branco.

Exemplo de saída

Teste 1

1

Teste 2

0

Teste 3

2

Restrições

$(1 \leq X_1 \leq Y_1 \leq X_2 \leq Y_2 \leq 8)$

$(X_1 = Y_1 = X_2 = Y_2 = 0, \text{ somente para indicar o final da entrada })$

Apagando e Ganhando

Arquivo fonte: *apaga.c*, *apaga.cc*, *apaga.cpp*, *apaga.pas*

Juliano é fã do programa de auditório Apagando e Ganhando, um programa no qual os participantes são selecionados através de um sorteio e recebem prêmios em dinheiro por participarem.

No programa, o apresentador escreve um número de N dígitos em uma lousa. O participante então deve apagar exatamente D dígitos do número que está na lousa; o número formado pelos dígitos que restaram é então o prêmio do participante.

Tarefa

Juliano finalmente foi selecionado para participar do programa, e pediu que você escrevesse um programa que, dados o número que o apresentador escreveu na lousa, e quantos dígitos Juliano tem que apagar, determina o valor do maior prêmio que Juliano pode ganhar.

Entrada

A entrada é composta por vários conjuntos de Teste. A primeira linha de cada caso de teste contém dois inteiros N e D , indicando a quantidade de dígitos do número que o apresentador escreveu na lousa e quantos dígitos devem ser apagados. A linha seguinte contém o número escrito pelo apresentador, que não contém zeros à esquerda. O final da entrada é indicado por $N = D = 0$.

Exemplo de entrada

```
4 2
3759
6 3
123123
7 4
1000000
0 0
```

Saída

Para cada Teste, você deverá imprimir um identificador “Teste n ”, onde n é enumerado a partir de 1. Na linha seguinte imprima o maior prêmio que Juliano pode ganhar. Após cada Teste imprima uma linha em branco.

Exemplo de saída

Teste1

79

Teste 2

323

Teste 3

100

Restrições

($1 \leq D < N \leq 100000$)

($N = D = 0$, somente para indicar o final da entrada)

Histórico de Comandos

Arquivo fonte: *fliperama.c*, *fliperama.cc*, *fliperama.cpp*, *fliperama.pas*

Uma interface por linha de comando (ILC) é um dos tipos de interface humano-computador mais antigos que existem. Uma ILC permite a interação com o software através de um interpretador de comandos, sendo normalmente acessível em um terminal (ou janela) de texto. A vantagem de um interpretador de comandos é que ele permite que o usuário opere o sistema usando apenas o teclado. Ainda hoje em dia, em que estamos acostumados com interfaces gráficas sofisticadas, muitos aplicativos e sistemas operacionais incluem algum tipo de interface por linha de comando, e muitos usuários ainda preferem usá-la para grande parte das tarefas. Um dos recursos mais úteis de um interpretador de comandos é o histórico de comandos. Quando um comando é digitado e executado, ele é colocado no histórico de comandos do terminal. O comando pode ser exibido novamente no terminal apertando a tecla '↑'; a tecla Enter executa o comando novamente quando o comando está sendo exibido no terminal. Todos os comandos executados são guardados no histórico: pressionar a tecla '↑' duas vezes exibe o penúltimo comando executado, pressioná-la três vezes exibe o antepenúltimo comando, e assim sucessivamente. Por exemplo, se o histórico inicial é (A, B, C, D), para repetir o comando C basta pressionar duas vezes a tecla '↑'. O histórico será então atualizado para (A, B, C, D, C). Nesse ponto, para repetir o comando A será necessário pressionar cinco vezes a tecla '↑'; o histórico será atualizado para (A, B, C, D, C, A). Nesse ponto, para repetir mais uma vez o comando A basta pressionar uma vez a tecla '↑'; o histórico será atualizado para (A, B, C, D, C, A, A). Leandro é administrador de sistemas e usa frequentemente o interpretador de comandos para gerenciar remotamente os servidores que administra. Em geral, ele precisa apenas repetir comandos que já havia digitado antes. Enquanto estava trabalhando em um servidor, ele teve uma curiosidade: quantas vezes ele precisa pressionar a tecla '↑' para executar uma determinada sequência de comandos? Ele sabe quais são as posições no histórico dos comandos que ele necessita executar, mas não sabe resolver esse problema. Por isso, pediu que você fizesse um programa que respondesse à pergunta dele.

Tarefa

Dada uma lista de comandos a serem executados calcular o número de vezes que Leandro precisa apertar a tecla '↑'.

Entrada

A entrada é composta de vários casos de teste. A primeira linha de cada caso de teste contém um número inteiro N , indicando o número de comandos que Leandro deseja executar. A segunda linha de um caso de teste contém N inteiros P_1, P_2, \dots, P_N , que indicam as posições

dos comandos no histórico no momento inicial, na ordem em que os comandos devem ser executados. Ou seja, o primeiro comando que deve ser executado está inicialmente na posição P_1 do histórico; depois deve ser executado o comando que está inicialmente na posição P_2 no histórico, e assim por diante, até P_N , que é a posição inicial do último comando que deve ser executado. Note que pode haver $P_i = P_j$. As posições são dadas em função do número de vezes que a tecla '↑' deve ser pressionada: um comando na posição 5 necessita que a tecla '↑' seja pressionada cinco vezes antes de aparecer no terminal (note que à medida que comandos vão sendo executados, a posição de um dado comando no histórico pode mudar). O final da entrada é indicado por $N = 0$.

Exemplo de entrada

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

Saída

Para cada caso de teste, seu programa deve imprimir um identificador do teste do tipo “Teste n”, onde n é numerado a partir de 1. A segunda linha deve conter o número de vezes que Leandro precisa pressionar a tecla '↑' para executar todos os comandos.

Exemplo de saída

```
Teste 1
13

Teste 2
16

Teste 3
25

Teste 4
```

9

(o exemplo de saída corresponde ao exemplo acima)

Restrições

$(1 \leq N \leq 1000)$

$(1 \leq P_i \leq 1000000)$

Rouba-Monte

Arquivo fonte: *rouba.c*, *rouba.cc*, *rouba.cpp*, *rouba.pas*

Um dos jogos de cartas mais divertidos para crianças pequenas, pela simplicidade, é Rouba- Monte. O jogo utiliza um ou mais baralhos normais e tem regras muito simples. Cartas são distingüidas apenas pelo valor (ás, dois, três, . . .), ou seja, os naipes das cartas não são considerados (por exemplo, ás de paus e ás de ouro têm o mesmo valor). Inicialmente as cartas são embaralhadas e colocadas em uma pilha na mesa de jogo, chamada de pilha de compra, com face voltada para baixo. Durante o jogo, cada jogador mantém um monte de cartas, com face voltada para cima; em um dado momento o monte de um jogador pode conter zero ou mais cartas. No início do jogo, todos os montes dos jogadores têm zero cartas. Ao lado da pilha de compras encontra-se uma área denominada de área de descarte, inicialmente vazia, e todas as cartas colocadas na área de descarte são colocadas lado a lado com a face para cima (ou seja, não são empilhadas). Os jogadores, dispostos em um círculo ao redor da mesa de jogo, jogam em sequência, em sentido horário. As jogadas prosseguem da seguinte forma:

- O jogador que tem a vez de jogar retira a carta de cima da pilha de compras e a mostra aos outros jogadores; vamos chamar essa carta de carta da vez.
- Se a carta da vez for igual a alguma carta presente na área de descarte, o jogador retira essa carta da área de descarte colocando-a, juntamente com a carta da vez, no topo de seu monte, com as faces voltada para cima, e continua a jogada (ou seja, retira outra carta da pilha de compras e repete o processo).
- Se a carta da vez for igual à carta de cima de um monte de um outro jogador, o jogador “rouba” esse monte, empilhando-o em seu próprio monte, coloca a carta da vez no topo do seu monte, face para cima, e continua a jogada.
- Se a carta da vez for igual à carta no topo de seu próprio monte, o jogador coloca a carta da vez no topo de seu próprio monte, com a face para cima, e continua a jogada.
- Se a carta da vez for diferente das cartas da área de descarte e das cartas nos topos dos montes, o jogador a coloca na área de descarte, face para cima, e a jogada se encerra (ou seja, o próximo jogador efetua a sua jogada). Note que esse é o único caso em que o jogador não continua a jogada.

O jogo termina quando não há mais cartas na pilha de compras. O jogador que tiver o maior monte (o monte contendo o maior número de cartas) ganha o jogo. Se houver empate, todos os jogadores com o monte contendo o maior número de cartas ganham o jogo.

Tarefa

Sua tarefa é fazer um programa que leia o número de cartas no baralho, o número de jogadores e as cartas na pilha de compras, em seguida o programa deve imprimir o número de cartas do ganhador e o número de identificação deste jogador, se houver mais de um ganhador imprimir em ordem crescente o número de todos os ganhadores.

Entrada

A entrada é composta de vários casos de teste. A primeira linha de um caso de teste contém dois inteiros N e J , representando respectivamente o número de cartas no baralho e o número de jogadores. As cartas do baralho são representadas por números inteiros de 1 a 13 e os jogadores são identificados por inteiros de 1 a J . O primeiro jogador a jogar é o de número 1, seguido do jogador de número 2, . . . , seguido pelo jogador de número J , seguido pelo jogador de número 1, seguido do jogador de número 2, e assim por diante enquanto houver cartas na pilha de compras. A segunda linha de um caso de teste contém N inteiros entre 1 e 13, separados por um espaço em branco, representando as cartas na pilha de compras. As cartas são retiradas da pilha de compras na ordem em que aparecem na entrada. O final da entrada é indicado por uma linha com $N = J = 0$.

Exemplo de entrada

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

Saída

Para cada caso de Teste seu programa deve imprimir um identificador “Teste n ”, onde n é enumerado a partir de 1. A segunda linha, contendo o número de cartas do monte do jogador ou jogadores que ganharam a partida, seguido de um espaço em branco, seguido do(s) identificador(es) dos jogadores que ganharam a partida. Se há mais de um jogador vencedor imprima os identificadores dos jogadores em ordem crescente, separados por um espaço em branco.

Exemplo de saída

Teste 1

0 1 2

Teste 2

5 1

Teste 3

3 2

Restrições

$(2 \leq N \leq 10.000)$

$(2 \leq J \leq 20 \text{ e } J \leq N)$

Ambulância

Arquivo fonte: *ambulancia.c*, *ambulancia.cc*, *ambulancia.cpp*, *ambulancia.pas*

O município de Águas Molhadas é formado por várias vilas interligadas por igarapés. Os igarapés funcionam como sistema viário, já que a maioria dos habitantes usa barcos como meio de transporte. Os igarapés têm uma forte corrente, o que obriga os barcos a transitarem todos no mesmo sentido em cada igarapé (na verdade, é proibido o trânsito de barcos na contra-mão). A prefeitura do município mantém um único Posto de Saúde e um barco-ambulância utilizado para transporte de doentes. O timoneiro do barco-ambulância está sempre com pressa, e tem medo de confundir-se no emaranhado de igarapés quando atende um chamado para buscar um doente em alguma vila. Por isso, ele deseja uma lista de todos os possíveis caminhos entre a vila onde se encontra o Posto de Saúde e cada outra vila do município, como mostra a Figura 9.2.

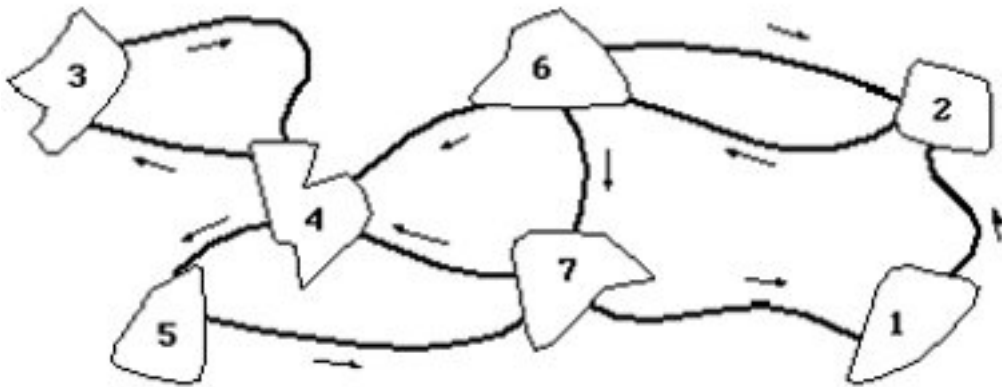


Figura 9.2: Sete vilas ligadas por onze igarapés com mão única.

Tarefa

Sua tarefa é escrever um programa que liste todos os caminhos que partem da vila onde se encontra o Posto de Saúde até cada outra vila do município, de forma que em cada caminho nenhuma vila é repetida. As vilas são numeradas de 1 a N, sendo que a vila onde se encontra o Posto de Saúde será sempre a de número 1.

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 a quantidade de vilas do município. As linhas seguintes contêm, cada uma, dois inteiros positivos X e Y que indicam que a vila X tem um igarapé que a liga diretamente com a vila Y, sem passar por outras vilas, com o trânsito fluindo na direção de X para Y. O final da lista de igarapés é indicado por X = 0 e Y = 0. O final da entrada é indicado por N = 0.

Exemplo de Entrada

```
2
1 2
2 1
0 0
7
1 2
2 6
7 4
7 1
4 5
4 3
3 4
6 4
6 7
6 2
5 7
0 0
0
```

Saída

Para cada conjunto de teste da entrada seu programa deve produzir um conjunto de saída com uma lista de caminhos. A primeira linha de um conjunto de saída deve conter o identificador do conjunto, no formato “Teste n”, onde n é numerado a partir de 1. As linhas seguintes devem conter os caminhos, ordenados lexicograficamente. Cada caminho é composto de uma sequência de identificadores de vilas separados por um ou mais espaços em branco. A última linha de um conjunto de saída deve ser deixada em branco.

Exemplo de Saída

```
Teste 1
1 2

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

1 2 6 4 3
1 2 6 4 5
1 2 6 4 5 7
1 2 6 7
1 2 6 7 4
1 2 6 7 4 3
1 2 6 7 4 5

Restrições

$0 \leq N \leq 100$ ($N = 0$ apenas para indicar o final da entrada)

$1 \leq Y \leq N$

$1 \leq X \leq N$

Caixas

Arquivo fonte: *caixa.c*, *caixa.cpp* ou *caixa.pas*

José é um fazendeiro que cultiva Banana e Abacates, ele percebeu que sempre que vende 1 caixa de Abacate também vende 1 caixa de Banana, nunca 2 caixas da mesma fruta. Para melhorar a rapidez de suas vendas, José quer organizar as caixas de Abacate de maneira que sempre fiquem à esquerda das caixas de Banana, porém, as caixas vazias não devem ser movidas para que mais tarde ele possa colocar Abacate ou Banana como preferir.

Tarefa

José pediu sua ajuda para fazer um programa que leia N caixas. O conteúdo de qualquer duas caixas vazias adjacentes não pode ser movido (trocas) com outras caixas escolhidas, preservando a sua posição inicial. Deve obter-se uma configuração onde todos os A's (caixas de abacate) são colocados à esquerda de todos os B's (caixas de banana), as posições das caixas vazias devem ser respeitadas.

Entrada

A entrada é composta por vários conjuntos de Teste. A primeira linha de cada caso de Teste é um inteiro N . A segunda linha terá os N elementos, onde duas caixas adjacentes vazias (representadas pelo 0), $(N/2) - 1$ símbolos "A" e $(N/2) - 1$ símbolos "B". O final da entrada é indicado por $N = 0$.

Exemplo de entrada

```
10
A A A A 0 0 B B B B
10
A A A A B B 0 0 B B
6
B B 0 0 B A A A
0
```

Saída

Para cada Teste, você deverá imprimir um identificador "Teste n", onde n é enumerado a partir de 1. Na linha seguinte imprima a configuração das caixas de maneira que satisfaça a necessidade de José. Após cada Teste imprima uma linha em branco.

Exemplo de saída

Teste 1

A B A B 0 0 A B A B

Teste 2

A B A B A B 0 0 A B

Teste 3

A B 0 0 A B A B

Restrições

$1 \leq N \leq 10$ (N sempre é par).

($N = 0$. Somente para indicar o final de entrada).

Popularidade

Arquivo fonte: *popularidade.c*, *popularidade.cc*, *popularidade.cpp*, *popularidade.pas*

Uma escola está promovendo uma eleição de popularidade, para determinar, naturalmente, quem é o aluno mais popular. Foi definido, então, que cada aluno deverá votar nos alunos de quem gosta. A quantidade de votos dados por cada aluno é variável, isto é, cada aluno pode votar em quantos alunos desejar, de acordo com suas preferências. O vencedor será aquele que receber mais votos, ou seja, aquele para o qual mais alunos indicaram que gostam dele. Para realizar a eleição, cada aluno receberá uma cédula eleitoral contendo os nomes de todos os alunos da escola (inclusive ele próprio), na qual deverá preencher os quadrados ao lado dos nomes dos alunos que gosta, utilizando caneta esferográfica azul ou preta. Após o término do período de votação, as cédulas serão colocadas numa máquina, a qual é capaz de informar quais quadrados foram preenchidos em cada cédula.

Tarefa

Você, estagiário da escola em questão, ficou responsável por apresentar um protótipo do sistema que recebe as informações da máquina e contabiliza os dados da eleição. Por se tratar de um protótipo, sua tarefa é apenas escrever um programa que, dadas as informações sobre simulações de preenchimento das cédulas, informe quantos votos recebeu o vencedor da eleição. Você pode assumir que os alunos da escola são participativos, de forma que todos compareceram à votação e cada um preencheu exatamente uma cédula eleitoral. Você pode assumir, ainda, que os alunos não sofrem por conflitos internos, de modo que cada aluno gosta de si mesmo e vota em si mesmo. Note, porém, que a relação “gostar de” não é simétrica, ou seja, se o aluno A gosta do aluno B, não necessariamente o aluno B gosta do aluno A.

Entrada

A entrada é composta por vários casos de teste, cada um correspondendo a uma simulação de eleição. A primeira linha de um caso de teste contém apenas um inteiro, N ($1 \leq N \leq 100$), indicando a quantidade de alunos da escola (identificados por inteiros de 1 a N) e, por consequência, a quantidade de cédulas preenchidas. A seguir há N linhas, cada uma correspondendo a uma cédula processada. Cada linha contém N inteiros, onde o j -ésimo inteiro da i -ésima linha é igual a 1, caso o j -ésimo quadrado da cédula i esteja preenchido (ou seja, o aluno de identificador i votou no aluno de identificador j); ou é igual a 0, caso contrário (o aluno de identificador i não votou no aluno de identificador j). A entrada termina quando $N = 0$.

Exemplo de entrada

```
3
1 0 1
0 1 1
1 0 1
5
1 1 1 0 0
1 1 0 1 1
1 0 1 0 1
0 1 0 1 0
0 1 1 1 1
3
1 0 0
0 1 0
0 0 1
0
```

Saída

Para cada Teste, você deverá imprimir um identificador “Teste n”, onde n é enumerado a partir de 1. Na linha seguinte seu programa deve imprimir apenas um inteiro, correspondente à quantidade de votos recebidos pelo vencedor da eleição.

Exemplo de saída

```
Teste 1
3

Teste 2
4

Teste 3
1
```

Sudoku

Arquivo fonte: *sudoku.c*, *sudoku.cc*, *sudoku.cpp*, *sudoku.pas*

O jogo de Sudoku espalhou-se rapidamente por todo o mundo, tornando-se hoje o passatempo mais popular em todo o planeta. Muitas pessoas, entretanto, preenchem a matriz de forma incorreta, desrespeitando as restrições do jogo. A matriz do jogo é uma matriz de inteiros 9×9 . Para ser uma solução do problema, cada linha e coluna deve conter todos os números de 1 a 9. Além disso, se dividirmos a matriz em 9 regiões 3×3 , cada uma destas regiões também deve conter os números de 1 a 9. O exemplo abaixo mostra uma matriz que é uma solução do problema.

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

Tarefa

Sua tarefa neste problema é escrever um programa que verifica se uma matriz preenchida é ou não uma solução para o problema.

Entrada

A entrada é composta por vários casos de teste. Na primeira linha é dado um número N de matrizes na entrada. Nas linhas seguintes são dadas as N matrizes. Cada matriz é dada em 9 linhas, em que cada linha contém 9 números inteiros.

Exemplo de entrada

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

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

Saída

Para cada Teste, você deverá imprimir um identificador “Teste n”, onde n é enumerado a partir de 1. Na segunda linha, seu programa deverá imprimir SIM se a matriz for a solução de um problema de Sudoku, e NAO caso contrário. Imprima uma linha em branco após cada Teste.

Exemplo de saída

```
Teste 1
SIM
```

```
Teste 2
NAO
```

Restrições

$1 \leq N \leq 20$.

Pós-Fixa

Arquivo fonte: *pos.c*, *pos.cc*, *pos.cpp*, *pos.pas*

Paulo é um menino muito esperto e curioso. Na tarde de ontem, ele estava resolvendo os exercícios de matemática da escola. Paulo observou que sempre as operações eram realizadas da esquerda para a direita, obedecendo as regras das operações. Ele resolveu, pesquisar na internet, se resolver os exercícios da direita para esquerda era válido, e num é que ele encontrou, É VÁLIDO e chama de notação pós-fixa. Ele desistiu então, fazer os exercícios aplicando a notação pós-fixa. Como essa notação era nova para Paulo, ele precisa de um programa que calcule o valor final da expressão para que ele possa saber se realizou corretamente os cálculos.

Tarefa

Sua tarefa é fazer um programa que dada a expressão que Paulo calculou informe o resultado, para que Paulo possa saber se acertou ou errou.

Entrada

A entrada é composta de vários casos de testes. A primeira e única linha de um caso de teste contém a expressão a ser calculada. O final da entrada deve ser dado pela expressão igual a 0.

Exemplo de entrada

```
9 + 1 * 5 * 3
0
```

Saída

Para cada Teste, você deverá imprimir um identificador “Teste n”, onde n é enumerado a partir de 1. A segunda linha de um caso deve ter apenas o resultado da expressão de entrada.

Exemplo de saída

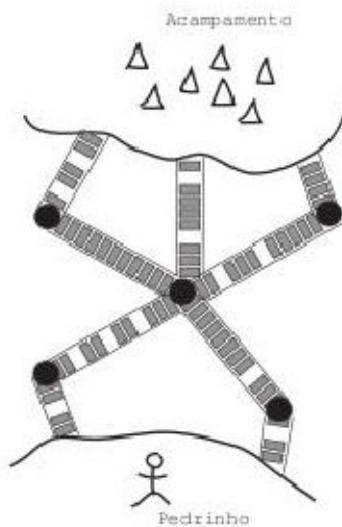
```
Teste 1
24
```

(O exemplo de saída corresponde ao exemplo de entrada).

Caminho das pontes

Arquivo fonte: *pontes.c*, *pontes.cc*, *pontes.cpp*, *pontes.pas*

Pedrinho é um rapaz muito aventureiro, que nas férias viaja pelo mundo em busca de lugares afastados e com bonitas vistas. Na sua viagem atual, Pedrinho está andando por uma escura floresta quando se depara com um perigoso desfiladeiro. Do outro lado do desfiladeiro ele sabe que existe um acampamento onde poderá descansar durante a noite para continuar suas aventuras no dia seguinte. Para chegar até o acampamento, ele terá que utilizar pontes que estão suspensas sobre o desfiladeiro. As pontes foram construídas interligando altos pilares cravados no fundo do desfiladeiro.



O piso das pontes é feita de tábuas de tamanhos iguais. Mas as pontes são velhas, e algumas tábuas caí ram. Felizmente, todas as tábuas que sobraram estão em perfeitas condições, ou seja, não existe o perigo de Pedrinho pisar em uma delas e a tábua cair. Além disso, em nenhuma das pontes duas tábuas consecutivas caí ram, de forma que os buracos deixados pelas tábuas que caíram podem ser pulados com segurança. No local onde Pedrinho se encontra existe uma placa mostrando as ligações entre as pontes e também quantas tábuas estão faltando em cada uma das pontes. Pedrinho está cansado e não há muita visibilidade durante a noite. Ele precisa, portanto, tomar muito cuidado para não cair em algum dos buracos. Pedrinho possui um laptop na mochila, mas só o usa para comunicar-se com os amigos. Ele liga sua internet via satélite, encontra você on-line, e pede sua ajuda.

Tarefa

Sua tarefa é escrever um programa que receba as informações sobre as pontes (as ligações entre elas e a quantidade de tábuas faltando em cada uma) e calcule qual é o menor número de buracos que Pedrinho precisa pular para chegar ao outro lado do desfiladeiro.

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 ($1 \leq N \leq 1.000, 2 \leq M \leq 10.000$) representando o número de pilares no desfiladeiro e o número de pontes, respectivamente. Cada uma das M linhas seguintes contém 3 inteiros S, T, B ($0 \leq S \leq N + 1, 0 \leq T \leq N + 1$ e $0 \leq B \leq 1.000$), indicando que existe uma ponte ligando os pilares S e T , e que possui B buracos. Não existe linha representando ponte com $S = T$. O valor de pilar 0 representa a borda do desfiladeiro onde Pedrinho está, e o valor de pilar $N + 1$ representa a borda do desfiladeiro onde está o acampamento. Não existem duas pontes distintas ligando o mesmo par de locais (pilares ou bordas do desfiladeiro). Você pode supor que sempre existir é um caminho de pontes entre o lado do desfiladeiro em que Pedrinho se encontra até o lado do desfiladeiro onde está o acampamento.

Exemplo de entrada

```
2 5
0 1 1
0 2 3
0 3 9
```

Saída

Seu programa deve imprimir, na saída padrão, um número inteiro representando a menor quantidade de buracos que Pedrinho terá que pular para conseguir chegar ao acampamento.

Exemplo de saída

```
3
(O exemplo de saída corresponde ao exemplo da entrada).
```

Dominó

Arquivo fonte: *domino.c*, *domino.cc*, *domino.cpp*, *domino.pas* Todos conhecem o jogo de dominós, em que peças com dois valores devem ser colocadas na mesa em seqüência, de tal forma que os valores de peças imediatamente vizinhas sejam iguais. O objetivo desta tarefa é determinar se é possível colocar todas as peças de um conjunto dado em uma formação válida.

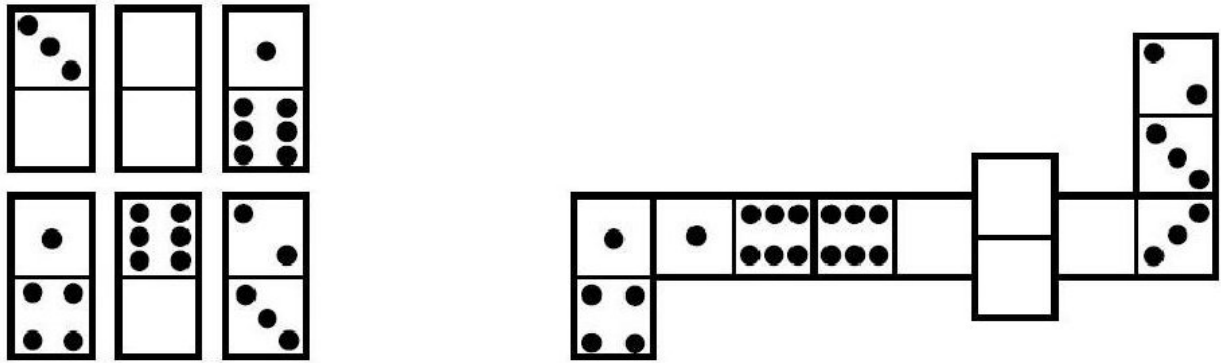


Figura 9.3: Conjunto de seis peças e uma formação utilizando todas as seis peças

Tarefa

É dado um conjunto de peças de dominó. Cada peça tem dois valores X e Y , com X e Y variando de 0 a 6 (X pode ser igual a Y). Sua tarefa é escrever um programa que determine se é possível organizar todas as peças recebidas em seqüência, obedecendo as regras do jogo de dominó.

Entrada

A entrada é composta de vários conjuntos de teste. A primeira linha de um conjunto de testes contém um número inteiro N que indica a quantidade de peças do conjunto. As N linhas seguintes contém, cada uma, a descrição de uma peça. Uma peça é descrita por dois inteiros X e Y ($0 \leq X \leq 6$ e $0 \leq Y \leq 6$) que representam os valores de cada lado da peça. O final da entrada é indicado por $N = 0$.

Exemplo de entrada

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

0 0
6
3 0
0 0
1 6
4 1
0 6
2 3
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 a expressão “sim ”se for possível organizar todas as peças em uma formação válida ou a expressão “não”(note a ausência de acento) caso contrário. 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
sim

Teste 2
nao

Teste 3
sim

(esta saída corresponde ao exemplo de entrada acima)

Restrições

$0 \leq N \leq 100$ ($N = 0$ apenas para indicar o final da entrada).