

### Problema: CAIXAS

*Joãozinho e sua família acabaram de se mudar. Antes da mudança, ele colocou todos os seus livros dentro de  $N$  caixas numeradas. Para facilitar a retirada dos livros, ele fez um inventário, indicando em qual caixa cada livro foi colocado, e o guardou na caixa de número 1.*

Chegando no seu novo quarto, ele viu que seus pais guardaram as caixas em  $P$  pilhas, arrumadas em fila, com cada pilha encostada na pilha seguinte. Joãozinho é um garoto muito sistemático; por isso, antes de abrir qualquer outra caixa, ele quer recuperar seu inventário.

Joãozinho também é um garoto muito desajeitado; para tirar uma caixa de uma pilha, ele precisa que a caixa esteja no topo da pilha e que ao menos um de seus lados, não importa qual, esteja livre (isto é, não tenham nenhuma caixa adjacente).

Para isso, Joãozinho precisa desempilhar algumas das caixas. Como o quarto dele é bem grande, ele sempre tem espaço para colocar as caixas retiradas em outro lugar, sem mexer nas pilhas que os pais dele montaram.

Para minimizar seu esforço, Joãozinho quer que você escreva um algoritmo que, dadas as posições das caixas nas pilhas, determine quantas caixas Joãozinho precisa desempilhar para recuperar seu inventário.

#### Dados

Número de Caixas e de Pilhas  $N$  e  $P$ , onde  $(1 \leq P \leq N \leq 1.000)$ . As caixas são numeradas sequencialmente de 1 a  $N$ .

Cada uma das  $P$  linhas seguintes descreve uma pilha. Cada linha contém um inteiro  $Q_i$ , indicando quantas caixas há na pilha  $i$ , seguido de um espaço em branco, seguido de uma lista de  $Q_i$  números, que são os identificadores das caixas.

Todas as pilhas contêm pelo menos uma caixa, e todas as caixas aparecem exatamente uma vez na entrada. As caixas em cada pilha são listadas em ordem, da base até o topo da pilha. Todas as caixas têm o mesmo formato.

Seu algoritmo deve determinar o número mínimo de caixas, além da caixa 1, que Joãozinho precisa desempilhar para recuperar o seu inventário.

### Problema: DNA

O DNA é uma molécula envolvida na transmissão de caracteres hereditários e na produção de proteínas, que são os principais constituintes de seres vivos. O DNA é formado pelas bases nitrogenadas adenina (A), guanina (G), citosina (C) e timina (T).

A identificação da sequência de bases que constitui uma determinada parte do DNA pode ajudar a descoberta da cura de doenças que atacam seres vivos. Teoricamente, a identificação do DNA pode também permitir a recriação de espécies extintas, como na estória do escritor americano Michael Crichton.

O professor de Biologia de sua escola, prof. Estevão Espilbergo, conseguiu amostras de células de uma espécie de mosquito extinto a milhares de anos, e pretende, ambiciosamente, recriar o animal a partir de seu DNA. Para isso, conseguiu que um laboratório de genômica fizesse a identificação das bases das células. No entanto, pelo estado precário das células obtidas, o resultado não foi dos melhores. O professor Estevão recebeu do laboratório duas sequências, com a informação de que essas sequências contêm, provavelmente, muitos "buracos", ou seja, entre uma base e outra corretamente detectadas podem existir bases não detectadas.

O prof. Estevão então decidiu combinar as duas sequências para formar uma sequência única, e precisa de sua ajuda.

### **Tarefa**

Sua tarefa é escrever um algoritmo que determine a menor sequência que contenha, como subsequências, as duas sequências obtidas pelo laboratório. Dizemos que uma sequência S1 é subsequência de uma outra sequência S2 se acrescentando-se alguns elementos a S1 obtém-se S2. Por exemplo, ACGT é uma subsequência de ATCGAAT, pois basta inserir um T após o A e dois A's após o G.

### **DADOS**

Dadas duas sequências S1 e S2 compostas por caracteres A, C, G e T, com tamanhos m e n respectivamente, determine a sequência de tamanho mínimo que tenha como subsequências S1 e S2.

### **Exemplo de Entrada**

AAATTT  
GAATCT

### **Saída**

GAAATCTT

### *Problema: APOSTAS*

Manuel quer ficar rico rápido e sem muito esforço, então ele decidiu fazer carreira apostando. Inicialmente, ele planeja estudar os ganhos e as perdas de jogadores, de modo que ele possa identificar padrões de vitórias consecutivas e elaborar uma estratégia que seja sempre vencedora. Contudo, Manuel, tão esperto como ele acha que é, não sabe como programar computadores, de modo que ele contratou você para escrever programas que irão auxiliá-lo a elaborar a estratégia dele.

Sua primeira tarefa é escrever um algoritmo que identifique o máximo ganho possível de uma sequência de apostas. Uma aposta é uma quantia de dinheiro e é ou vencedora (e isto é registrado como um valor positivo), ou perdedora (e isto é registrado como um valor negativo). Entrada

Dada uma sequência de com N apostas construa um algoritmo que determine o máximo ganho dentro da sequência de apostas. No caso da sequência não obter ganho escreva "Sequência de derrotas."

Exemplo: Na sequência de apostas : 12, -4, -10, 4 , 9 o ganho máximo é 13

### **Problema: PALÍNDROMO**

Uma cadeia de caracteres é chamada de palíndromo se sequência de caracteres da esquerda para a direita é igual à sequência de caracteres da direita para a esquerda (uma outra definição é que o primeiro caractere da cadeia deve ser igual ao último caractere, o segundo caractere seja igual ao penúltimo caractere, o terceiro caractere seja igual ao antepenúltimo caractere, e assim por diante). Por exemplo, as cadeias de caracteres 'mim', 'axxa' e 'ananaganana' são exemplos de palíndromos.

Se uma cadeia não é palíndromo, ela pode ser dividida em cadeias menores que são palíndromos. Por exemplo, a cadeia 'aaxyx' pode ser dividida de quatro maneiras distintas, todas elas contendo apenas cadeias palíndromos: {'aa', 'xyx'}, {'aa', 'x', 'y', 'x'}, {'a', 'a', 'xyx'} e {'a', 'a', 'x', 'y', 'x'}.

Escreva um algoritmo que, dada uma cadeia de caracteres  $C$ , de tamanho  $n$ , determine qual o menor número de partes em que uma cadeia deve ser dividida de forma que todas as partes sejam palíndromos.

Exemplo para a cadeia  $C=bbabcbbaab$  o número mínimo é 5

Problema: Comércio de Camelos

Por volta de 800 A.C., El Mamum, Califa de Bagdá, foi apresentado à fórmula  $1+2*3*4+5$ , que teve sua origem na contabilidade financeira de uma transação de camelos. A fórmula estava sem parênteses e era ambígua. Sendo assim, ele decidiu pedir aos servos para achar um método de encontrar qual interpretação da fórmula é mais vantajosa para ele, dependendo se ele está comprando ou vendendo os camelos.

Você foi contratado por El Mamum para escrever um algoritmo que determine as interpretações máxima e mínima possíveis de uma expressão sem parênteses.

### Dados

Dada uma expressão composta de  $N$  números separados pelos operadores de soma  $+$  e de produto  $*$ . Escreva um algoritmo que determine as interpretações com valores máximos e mínimos.

Problema: Árvore Genealógica

A empresa Genesis faz vários serviços dedicados a famílias, principalmente para descobrir parentes e parentescos perdidos. Para mostrar a eficiência de seus serviços, a Genesis quer, para cada árvore genealógica que ela tem, achar o par de parentes com parentesco mais distante. Para isso, eles definiram uma "função-parentesco" definida da seguinte forma:

1.  $\text{grau}(A, B) = 0$ , se  $A = B$
2.  $\text{grau}(A, B) = 1 + \min\{\text{grau}(\text{pai}(A), B), \text{grau}(A, \text{pai}(B))\}$ , se  $A \neq B$

### Dados

Dadas  $N$  relações pai-filho, construa um algoritmo que, dada uma árvore genealógica, determine o nome dos dois parentes com maior grau de parentesco, e o valor do grau de parentesco entre eles.

### Exemplo

Para a lista de relações pai-filho  $\{a-b; b-c; a-d; d-e; e-f\}$  o resultado esperado é  $c, f$  5.

Problema: Saldo de Gols

Hipólito é um torcedor fanático. Coleciona flâmulas, bandeiras, recortes de jornal, figurinhas de jogadores, camisetas e tudo o mais que se refira a seu time preferido. Quando ganhou um computador de presente em uma festa, resolveu montar um banco de dados com os resultados de todos os jogos de seu time ocorridos desde a sua fundação, em 1911. Depois de inseridos os dados, Hipólito começou a ficar curioso sobre estatísticas de desempenho do time. Por exemplo, ele deseja saber qual foi o período em que o seu time acumulou o maior saldo de gols. Como Hipólito tem o computador há muito pouco tempo, não sabe programar muito bem, e precisa de sua ajuda.

## Dados

A partir de uma lista R com os resultados( $R_i$ ) de N jogos do time (primeira partida: 3 x 0, segunda partida: 1 x 2, terceira partida: 0 x 5 ...). Sua tarefa é escrever um algoritmo que determine em qual período o time conseguiu acumular o maior saldo de gols, e qual foi esse saldo. Um período é definido pelos números de sequência de duas partidas, I e J, onde  $I \leq J$ . O saldo de gols acumulado entre I e J é dado pela soma dos gols marcados pelo time em todas as partidas realizadas entre I e J (incluindo as mesmas) menos a soma dos gols marcados pelos times adversários no período. Se houver mais de um período com o mesmo saldo de gols, escolha o maior período (ou seja, o período em que  $J - I$  é maior). Se ainda assim houver mais de uma solução possível, escolha qualquer uma delas como resposta.

**Exemplo:** Para a lista  $R = \{2X2, 0X5, 6X2, 1X4, 0X0, 5X1, 1x5, 6x2, 0x5\}$ , onde cada placar corresponde aos gols do meu time X time adversário, o resultado esperado é o período entre os jogos 3-8 com saldo +5.

## Problema: Material de Construção

Digamos que você deseja transportar alguns itens de materiais de construção com o intuito de levá-los da loja ao local da construção. Para tanto, conseguiu emprestado um caminhão que comporta 20 m<sup>3</sup> de volume como capacidade máxima. Como o veículo deverá ser devolvido o mais rápido possível, poderá ser realizada apenas uma viagem. Assim, devem ser escolhidos os itens de forma a otimizar o conteúdo transportado pelo caminhão ao fazer o frete.

Dada uma lista de materiais de construção informando o volume a ser transportado o valor do benefício proporcionado pela entrega desse material, construa um algoritmo de determine o benefício máximo que carga pode alcançar e determine a lista de materiais a serem transportados que acarretam esse benefício máximo.

Exemplo de Entrada:

Ítem	Quantidade m3	Benefício
Areia	12	8
Pedra	12	7
Cimento	3	13
Cal	4	11,5
Madeira	8	10
Ferro	2	10

## Problema: Desenvolvimento de Software

Você é gerente de uma empresa de desenvolvimento de software de médio porte que precisa desenvolver um sistema grande que pode ser dividido em módulos. Cada módulo possui um prazo para ser implementado de forma que o desenvolvimento como um todo não ultrapasse a data final prevista pelos analistas e não acarrete em penalizações para a empresa. Você, como gerenciador, deve determinar o mínimo de pessoas para desenvolver tais módulos de forma a otimizar o custo de pessoal (quanto menos pessoas, menor o custo). Observe que uma pessoa só pode dedicar-se a um módulo por vez. A partir de uma lista com as datas de início e término das implementações de cada módulo, construa um algoritmo que determine o número mínimo de pessoas para desenvolver tais módulos.

Exemplo de Entrada:

Início	1	1	1	3	5	4	2	4	7	15	8	14	15	18	23
Fim	5	7	3	6	12	12	8	9	11	19	12	17	18	23	27

Problema: Update

Com o surgimento de novas aplicações para computadores cada vez mais pesadas, notou-se a necessidade de atualizar o hardware em uso para satisfazer as exigências das aplicações emergentes. Com o feriado surgiu a oportunidade de deslocar-se a Pedro Juan Caballero para efetuar as compras das peças para a atualização, visto a grande diferença de preços encontrada no mercado externo e no país vizinho. Porém, como é de interesse que a compra seja declarada ao fisco com o intuito de legalizar a compra, achou-se necessário a declaração dos bens adquiridos. Para tanto, o valor máximo que a compra pode atingir é de U\$ 300,00, já que a entrada no Brasil foi feita via terrestre.

DADOS:

Dada uma lista de peças, seus preços e benefícios espero, construa um algoritmo que determine o benefício máximo com o valor limite informado, e a lista de peças a serem compradas.

Exemplo de entrada:

Item	Valor(U\$)	Benefício
Processador	98	14
Placa-Mãe	101	13
HD	73	12
Placa de Vídeo	65	9
Pende de Memória	20	10
kit(teclado,mouse e gabinete)	40	12,5
Drive Blue-Ray	20	10
Monitor	63	11

Problema: Guerra

Guerra, um evento digno apenas de aparecer na literatura, filmes ou talvez programação de concursos, veio a atingir o império Noglôniano, que está enfrentando o império vizinho de Quadradônia. Protocolos de guerra combinados por ambas as partes indicam que a guerra será travada em sucessivas batalhas, em cada uma das quais um soldado diferente de cada império vai enfrentar outro, de modo que cada soldado irá participar em exatamente uma batalha. O império que ganhar mais batalhas ganha a guerra. Cada império tem um exército formado por S soldados, e cada soldado tem uma certa habilidade de combate. Em cada batalha entre dois soldados, aquele com maior habilidade de combate ganha a batalha. Se ambos os soldados têm as mesmas habilidades de combate, a batalha é declarada como empate e tecnicamente nenhum lado é vitorioso. Os espões de Noglônia tiveram que interceptar informações secretas relativas às habilidades de combate de cada soldado do exército de Quadradônia, por isso a rainha de Noglônia requer a sua assistência, a fim de calcular o número máximo de batalhas que podem ganhar durante a guerra se os seus soldados forem enviados na ordem apropriada.

DADOS:

Dada uma lista com  $Q$  números inteiros, onde cada número  $Q_i$  representa as habilidades de combate dos diferentes soldados do exército de Quadradônia, na ordem em que as batalhas irão acontecer. Uma segunda lista com  $N$  números inteiros, onde cada número  $N_i$  representa as habilidades de combate dos diferentes soldados do exército de Noglônia, em uma ordem arbitrária.

Construa um algoritmo que determine o número máximo de batalhas que Noglônia pode ganhar durante a guerra.

Exemplo de Entrada  $Q=\{2, 1, 1\}$  e  $N=\{1, 1, 2\}$

Resultado esperado 1

Problema: Dentista

Os dentistas são extremamente meticolosos em seu trabalho, tendo que agir com muita precisão em todas as suas atividades. Pedro é um dentista meticoloso como todos os outros. Infelizmente sua secretária não é muito organizada e, com o intuito de ajudar sempre os pacientes, aceita que eles marquem consultas no horário que quiserem, sem se preocupar com os demais horários marcados, ocasionando vários conflitos de horários que muito incomodaram Pedro e os pacientes. Por exemplo, se uma consulta começar às 9 horas e durar 2 horas, nenhuma outra consulta deveria ser marcada para iniciar às 10 horas.

Ao perceber que sua agenda tinha conflito de horários, Pedro pediu sua ajuda para descobrir a maior quantidade de consultas que podem ser atendidas sem sobreposição.

Você deve escrever um algoritmo que, dados os horários de início e término das consultas agendadas pela secretária, responda a quantidade máxima de consultas que podem ser atendidas sem sobreposição.

Problema: Organizado Pacotes

Uma empresa de mineração extrai térbio, um metal raro usado para a construção de ímãs leves, a partir de areia de rio. Eles mineram um grande rio em  $N$  pontos de mineração, cada um deles identificado por sua distância a partir da origem do rio. Em cada ponto de mineração, uma pequena pilha ou amontoado de minério mineral altamente valorizado é extraído do rio.

Para recolher o minério mineral, a empresa reagrupa os  $N$  amontoados produzidos em um menor número de  $K$  pilhas ou montes maiores, cada um localizado num dos pontos de extração inicial. Os montes recém-formados são então recolhidos por caminhões.

Para reagrupar os  $N$  montes eles usam uma barca, o que na prática pode levar qualquer quantidade de minério mineral por ser bem larga. A barcaça começa na origem do rio e somente pode viajar rio abaixo, de modo que o amontoado de mineral produzido em um ponto  $X$  de mineração pode ser levado para um ponto  $Y$  de mineração somente se  $Y > X$ . Cada monte é movimentado completamente para outro ponto de mineração, ou não se move. O custo de mover um monte com peso  $W$  a partir de um ponto  $X$  de mineração para um ponto  $Y$  de mineração é  $W(Y - X)$ . O custo

total do agrupamento é a soma dos custos de cada movimento de um monte. Nota-se que um monte que não é movido não tem influência sobre o custo total.

Dados os valores de  $N$  e  $K$ , os  $N$  pontos de mineração, e o peso da pilha ou amontoado produzido de cada ponto de mineração, escreva um programa que calcule o custo total mínimo para reagrupar estes  $N$  montinhos iniciais em  $K$  pilhas ou montes maiores.

Dados:

Dados dois inteiros  $N$  e  $K$  os quais denotam respectivamente, o número de montes ou pilhas iniciais e o número desejado de montes após o reagrupamento ( $1 \leq K < N$ ). Cada um dos montes iniciais é acompanhado de dois números inteiros  $X$  e  $W$  indicando que o ponto  $X$  de mineração produziu um amontoado com peso de  $W$  ( $1 \leq X, W \leq N$ ). Os montes ou pilhas são dados estritamente em ordem ascendente, considerando os seus pontos de mineração.

Construa um algoritmo que determine o mínimo custo total para reagrupar os  $N$  amontoados iniciais em  $K$  montes maiores.

Exemplos:

Entrada 1:  $N=3$   $K=1$   $\{(20,1), (30,1), (40,1)\}$  Resultado esperado: 30

Entrada 2:  $N=3$   $K=1$   $\{(11,3), (12,2), (13,1)\}$  Resultado esperado: 8

Entrada 3:  $N=6$   $K=2$   $\{(10,15), (12,17), (16,18), (18,13), (30,10), (32,1)\}$  Resultado esperado: 278

Problema: Apanhando e Ganhando

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.

Juliano finalmente foi selecionado para participar do programa, e pediu que você escrevesse um algoritmo 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.

Problema: Soma Permutada Elegante

São disponibilizados a você  $n$  inteiros  $A_1 A_2 A_3 \dots A_n$ . Encontre uma permutação destes  $n$  inteiros de forma que a soma da diferença absoluta entre os elementos adjacentes seja maximizada.

Supondo  $n = 4$  e que os inteiros fornecidos sejam 4 2 1 5. A permutação 2 5 1 4 permite a máxima soma. Para esta permutação, teríamos soma =  $\text{abs}(2-5) + \text{abs}(5-1) + \text{abs}(1-4) = 3+4+3 = 10$ .

De todas as 24 permutações possíveis, você não vai obter qualquer soma cujo valor exceda 10. Nós iremos chamar este valor 10, de soma permutada elegante.

Crie um algoritmo que dada uma lista de inteiros não-negativos determine a Soma Permutada Elegante.

#### Problema: Festival de Escultura de Gelo

Todos os anos, artistas de todo o mundo se reúnem na cidade, onde fazem esculturas de gelo gigantescas. A cidade vira uma galeria de arte ao céu aberto, uma vez que as esculturas ficam expostas na rua por semanas, sem derreter. Afinal, a temperatura média no inverno de Harbin (época em que ocorrerá a final mundial do ICPC) é de -20 graus.

O primeiro passo para fazer a escultura é montar um grande bloco de gelo da dimensão pedida pelo artista. Os blocos são recortados das geleiras de Harbin em blocos de altura e profundidade padrão e vários comprimentos diferentes. O artista pode determinar qual o comprimento que ele deseja que tenha o seu bloco de gelo para que a escultura possa começar a ser esculpida.

Os comprimentos disponíveis dos blocos são  $\{a_1; a_2; \dots; a_N\}$  e o comprimento que o artista deseja é  $M$ . O bloco de comprimento 1 é muito usado, por este motivo ele sempre aparece na lista de blocos disponíveis.

#### Dados:

Dois inteiros  $N$  e  $M$  representando o número de tipos de blocos e o comprimento desejado pelo artista, respectivamente; e uma lista de inteiros  $a_1; a_2; \dots; a_N$ , representando o comprimento dos blocos. Sua tarefa é construir um algoritmo que determine o número mínimo de blocos tal que a soma de seus comprimentos seja  $M$ .

Exemplo:  $N=6$   $m=100$   $\{1,5,10,15,25,50\}$  Resultado esperado: 2