



Programação Competitiva — Primeiro Simulado

Profs. Guilherme Ramos, Rodrigo Bonifácio - 2014/2

| Matrícula | Nome |
|-----------|------|
| | |

Caderno de Problemas

Instruções

1. Este caderno contém 3 problemas, rotulados de A, B e C.
2. Em todos os problemas, a entrada do seu programa deve ser lida da *entrada padrão*. A saída deve ser escrita na *saída padrão*.
3. A avaliação é individual e tentativas de plágio entre os alunos não serão admitidas.
4. A duração do simulado é de 120 minutos.

Problema A

Botas Perdidas

Nome do arquivo fonte: *botas.c*, *botas.cpp* ou *botas.java*

A divisão de Suprimentos de Botas e Calçados do Exército comprou um grande número de pares de botas de vários tamanhos para seus soldados. No entanto, por uma falha de empacotamento da fábrica contratada, nem todas as caixas entregues continham um par de botas correto, com duas botas do mesmo tamanho, uma para cada pé. O sargento mandou que os recrutas retirassem todas as botas de todas as caixas para reembalá-las, desta vez corretamente.

Quando o sargento descobriu que você sabia programar, ele solicitou com a gentileza habitual que você escrevesse um programa que, dada a lista contendo a descrição de cada bota entregue, determina quantos pares corretos de botas poderão ser formados no total.

Entrada

A entrada é composta por diversos casos de teste. A primeira linha da entrada contém um inteiro N indicando o número de botas individuais entregues. Cada uma das N linhas seguintes descreve uma bota, contendo um número inteiro M e uma letra L , separados por um espaço em branco. M indica o número do tamanho da bota e L indica o pé da bota: $L = 'D'$ indica que a bota é para o pé direito, $L = 'E'$ indica que a bota é para o pé esquerdo.

Restrições:

- $2 \leq N \leq 10^4$
- N é par.
- $30 \leq M \leq 60$
- $L \in \{'D', 'E'\}$

Saída

Seu programa deve imprimir uma única linha contendo um único número inteiro indicando a quantidade total de pares corretos de botas que podem ser formados.

Exemplos

| Exemplo de Entrada | Saída esperada |
|--------------------|----------------|
| 4 | 2 |
| 40 D | 1 |
| 41 E | |
| 41 D | |
| 40 E | |
| 6 | |
| 38 E | |
| 39 E | |
| 40 D | |
| 38 D | |
| 40 D | |
| 37 E | |

Problema B

Copa do Mundo

Nome do arquivo fonte: *copa.c*, *copa.cpp* ou *copa.java*

Uma Copa do Mundo de futebol de botões está sendo realizada com times de todo o mundo. A classificação é baseada no número de pontos ganhos pelos times, e a distribuição de pontos é feita da forma usual. Ou seja, quando um time ganha um jogo, ele recebe 3 pontos; se o jogo termina empatado, ambos os times recebem 1 ponto; e o perdedor não recebe nenhum ponto.

Dada a classificação atual dos times e o número de times participantes na Copa do Mundo, sua tarefa é de determinar quantos jogos terminaram empatados até o momento.

Entrada

A entrada é composta por diversos casos de teste. A primeira linha de um caso de teste contém dois inteiros T e N , indicando respectivamente o número de times participantes e o número de partidas jogadas. Cada uma das T linhas seguintes contém o nome de um time (uma cadeia de no máximo 10 letras e dígitos), seguido de um espaço em branco, seguido do número de pontos que o time obteve até o momento.

Restrições:

- $2 \leq T \leq 200$
- $0 \leq N \leq 10^4$
- Nome de time contém no máximo 10 caracteres, que podem ser dígitos ou letras maiúsculas e minúsculas sem acento.

Saída

Para cada um dos casos de teste seu programa deve imprimir uma única linha contendo um número inteiro, representando a quantidade de jogos que terminaram empatados até o momento.

Exemplos

| Exemplo de Entrada | Saída esperada |
|--|----------------|
| 3 3 Brasil 3 Australia 3 Croacia 3 | 0 2 |
| 3 3 Brasil 5 Japao 1 Australia 1 0 0 | |

Problema C

Registrador de Deslocamento

Nome do arquivo fonte: *registrador.c*, *registrador.cpp* ou *registrador.java*

Um Registrador de Deslocamento é um circuito que desloca de uma posição os elementos de um vetor de bits. O registrador de deslocamento tem uma entrada (um bit) e uma saída (também um bit), e é comandado por um pulso de relógio. Quando o pulso ocorre, o bit de entrada se transforma no bit menos significativo do vetor, o bit mais significativo é jogado na saída do registrador, e todos os outros bits são deslocados de uma posição em direção ao bit mais significativo do vetor (em direção à saída).

Um Registrador de Deslocamento com Retroalimentação Linear (em Inglês, LFSR) é um registrador de deslocamento no qual o bit de entrada é determinado pelo valor do ou-exclusivo de alguns dos bits do registrador antes do pulso de relógio. Os bits que são utilizados na retroalimentação do registrador são chamados de torneiras. A Figura 1 mostra um LFSR de 8 bits, com três torneiras (bits 0, 3 e 5).

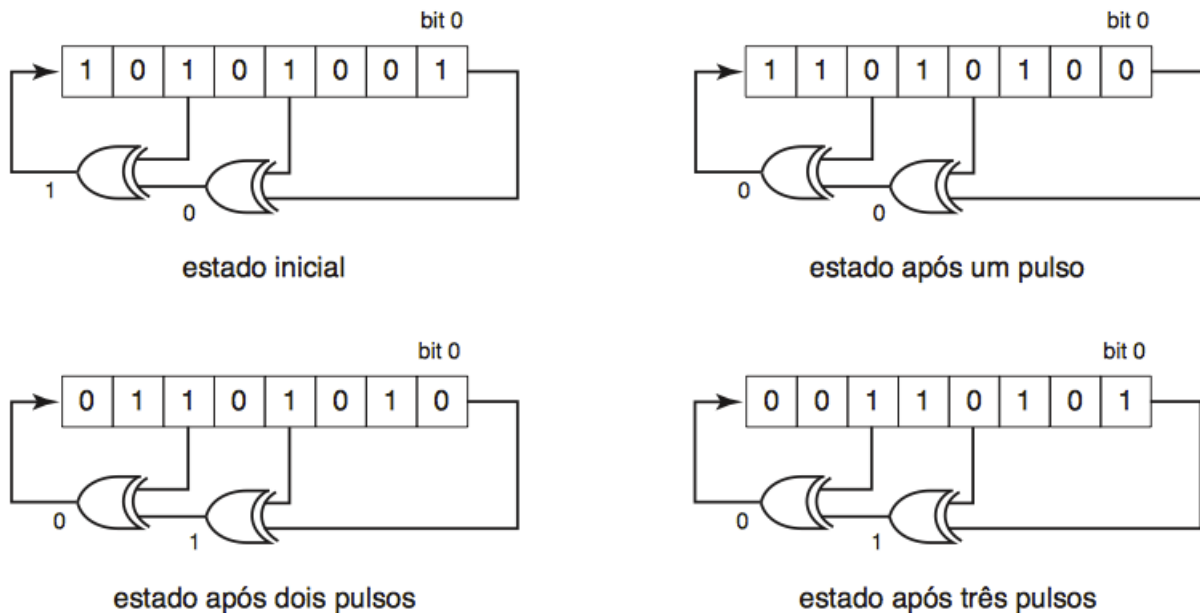


Figura 1: Exemplo de LFSR com 8 bits

Neste problema, você deve escrever um programa que, dados o número de bits de um LFSR, quais bits são utilizados na retroalimentação, um estado inicial e um estado final do LFSR, determine quantos pulsos de relógio serão necessários para que, partindo do estado inicial, o LFSR chegue ao estado final (ou determinar que isso é impossível).

Entrada

A entrada contém vários casos de teste. Cada caso de teste é composto por três linhas. A primeira linha contém dois números inteiros N , T , indicando respectivamente o número de bits ($2 \leq N \leq 32$) e o número de torneiras ($2 \leq T \leq N$). Os bits são identificados por inteiros de 0 (bit menos significativo) a $N - 1$ (bit mais significativo). A segunda linha contém T inteiros, separados por espaços, apresentando os identificadores dos bits que são torneiras, em ordem crescente. O bit 0 sempre é uma torneira. A terceira linha contém dois números em notação hexadecimal I e F , separados por um espaço em branco, representando respectivamente o estado inicial e o estado final do LFSR.

Saída

Para cada caso de teste da entrada seu programa deve imprimir uma única linha. Se for possível chegar ao estado final a partir do estado inicial dado, a linha da saída deve conter apenas um inteiro, o menor número de pulsos de relógio necessários para o LFSR atingir o estado final. Caso não seja possível, a linha deve conter apenas o caractere '*'.

Exemplos

| Exemplo de Entrada | Saída esperada |
|--------------------|----------------|
| 8 3 | 3 |
| 0 3 5 | * |
| a9 35 | 61 |
| 5 2 | |
| 0 4 | |
| 1b 2 | |
| 7 3 | |
| 0 2 3 | |
| 4d 1a | |
| 0 0 | |