

## A Empilhando Caixas

O chefe do almoxarifado de uma pequena empresa deu a seguinte instrução aos seus funcionários: uma caixa só pode ser colocada em cima de uma caixa que contém massa maior ou igual a da caixa a ser empilhada.

As caixas a serem empilhadas tem mesmas dimensões e são fabricadas com mesmo material, de modo que são diferenciadas apenas pelas etiquetas que indicam a massa do material armazenado na caixa.

Com a chegada de uma nova remessa de material, os funcionários reservaram o espaço necessário para uma nova pilha de caixas. Um funcionário trazia as caixas do caminhão para a porta do almoxarifado, enquanto um segundo funcionário consultava a etiqueta da caixa recém-chegada: se a massa contida nela permitisse o empilhamento, o funcionário a colocava na pilha; caso contrário, separava a caixa numa sala ao lado.

Sabendo que a pilha pode ter, no máximo, seis caixas, e conhecida a ordem em que as caixas foram retiradas do caminhão e analisadas, determine a massa total que o funcionário conseguiu empilhar.

### Entrada

A entrada consiste em uma série de casos de teste. A primeira linha de um caso de teste contém a quantidade  $N$  ( $1 \leq N \leq 20$ ) de caixas que serão retiradas do caminhão. As  $N$  linhas seguintes contém as massas  $m_i$  ( $1 \leq m_i \leq 100$ ), em ordem de retirada, de cada caixa  $i$  ( $1 \leq i \leq N$ ). A massa de cada caixa é dada em kilogramas.

### Saída

Para cada caso de testes deve ser impressa a mensagem “Massa empilhada:  $M$  kg”, onde  $M$  é a massa total do material contido nas caixas que foram empilhadas, seguida de uma quebra de linha.

Exemplos de entradas	Exemplos de saídas
1	Massa empilhada: 12 kg
12	
3	Massa empilhada: 20 kg
10	Massa empilhada: 70 kg
10	Massa empilhada: 120 kg
15	
6	
30	
20	
25	
12	
20	
8	
10	
40	
25	
30	
25	
20	
8	
10	
2	
2	
1	

*Este problema foi elaborado para ensino e docência. Quaisquer coincidências com problemas já existentes favor entrar em contato ([edsonalves@unb.br](mailto:edsonalves@unb.br)) para que as devidas providências sejam tomadas.*

## B Matrículas e Lista de Espera

O sistema acadêmico de uma universidade deve matricular os alunos em disciplinas no início de cada semestre. O registro de cada aluno contém quatro campos: nome, matrícula, posição no fluxo e situação (indica se o aluno está ou não em condição).

As disciplinas tem vagas limitadas, de modo que os candidatos a uma dada disciplina são ordenados segundo os seguintes critérios:

1. os alunos em condição tem precedência em relação aos alunos regulares;
2. se dois alunos estão na mesma situação, tem precedência aquele que tem maior posição no fluxo;
3. se dois alunos tem mesma situação e mesma posição no fluxo, será aplicada a ordem de inscrição no sistema (quem solicitou a disciplina antes tem precedência).

Uma vez ordenados os candidatos, eles são matriculados em sequência, do primeiro até o limite de vagas. Os candidatos não matriculados formam a lista de espera, que respeita a ordenação já estabelecida.

Conhecidos o número de vagas em uma dada disciplina e os candidatos, determine o nome do primeiro candidato da lista de espera, se houver.

### Entrada

A primeira linha da entrada informa o número  $V$  ( $1 \leq V \leq 1.000$ ) de vagas na disciplina, enquanto a segunda linha consiste no número  $M$  ( $1 \leq M \leq 10.000$ ) de candidatos à disciplina.

Em seguida, há  $M$  linhas com os dados dos candidatos, separados por espaços em branco: nome  $N$  (uma string composta por até 30 caracteres alfabéticos), posição no fluxo  $P$  ( $1 \leq P \leq 20$ ) e situação  $S$  (um único caractere, 'R' para regular e 'C' para condição). A ordem destas linhas corresponde a ordem de inscrição dos alunos.

### Saída

A saída consiste no nome do primeiro aluno da lista de espera, seguido de uma quebra de linha. Caso não seja formada uma lista de espera, a saída deve ser a mensagem “Todos foram matriculados”, sem aspas, também seguida de uma quebra de linha.

Exemplos de entradas	Exemplos de saídas
2	Maria
3	
Maria 2 R	
Joao 4 C	
Beto 1 C	
3	Tatiane
5	
Carlos 3 R	
Ana 5 R	
Lucas 1 R	
Saulo 4 R	
Tatiane 2 R	
5	Todos foram matriculados
3	
Yolanda 3 C	
Mauro 8 R	
Bruno 2 C	

*Este problema foi elaborado para ensino e docência. Quaisquer coincidências com problemas já existentes favor entrar em contato (edsonalves@unb.br) para que as devidas providências sejam tomadas.*

## C Pratos e Talheres

A cantina de uma escola de ensino fundamental tem quatro conjuntos de pratos e talheres, que se diferenciam por suas cores: azul, verde, branco e preto.

Durante o intervalo do lanche, uma funcionária lava os pratos sujos que estão empilhados à sua esquerda, retirando um prato do topo da pilha a cada  $t_1$  segundos. Caso a pilha esteja vazia no momento em que ela tentar retirar um prato, ela descansará  $2t_1$  segundos e tentará retirar um prato novamente.

O funcionário que é responsável pelo recolhimento dos pratos coloca um novo prato sujo no topo da pilha de pratos a serem lavados a cada  $t_2$  segundos, enquanto houverem pratos sujos a serem recolhidos.

Se as duas ações devem acontecer em um mesmo instante, a funcionária tentará retirar (ou retirará, se for possível) o prato antes da inserção do próximo prato a ser lavado, se houver.

Considere que, no instante inicial, a pilha de pratos sujos contenha apenas um prato azul e que não serão adicionados nem removidos pratos no instante inicial. Nestas condições, determine a cor do  $N$ -ésimo prato a ser lavado, dada a sequência de pratos sujos a serem inseridos na pilha pelo funcionário responsável.

### Entrada

A primeira linha da entrada contém o valor de  $N$  ( $1 \leq N \leq 10.000$ ), que representa a posição do prato a ser determinado, na ordem em que foram lavados, sendo que a contagem começa no número um.

A próxima linha contém os intervalos de tempo  $t_1$  e  $t_2$  ( $1 \leq t_1, t_2 \leq 1.000$ ), em segundos, que os funcionários esperam para pegar ou inserir um prato na pilha, respectivamente, separados por um espaço em branco.

Na linha subsequente está a quantidade  $M$  ( $N - 1 \leq M \leq 100.000$ ) de pratos sujos a serem inseridos na pilha. As próximas  $M$  linhas da entrada descrevem os pratos sujos a serem inseridos na pilha, em ordem, de acordo com a sua cor.

### Saída

A saída do programa deverá ser a cor do  $N$ -ésimo prato lavado, seguida de uma quebra de linha. O nome da cor deve iniciar com letra maiúscula, e os demais caracteres devem ser todos minúsculos.

Exemplos de entradas	Exemplos de saídas
1 20 30 0	Azul
2 20 30 1 preto	Preto
3 20 30 3 preto branco verde	Verde
5 125 99 6 preto branco verde azul preto branco	Branco

*Este problema foi elaborado para ensino e docência. Quaisquer coincidências com problemas já existentes favor entrar em contato ([edsonalves@unb.br](mailto:edsonalves@unb.br)) para que as devidas providências sejam tomadas.*

## D Sistema Distribuído

No processo de inicialização de um sistema são carregados  $M$  processos em um vetor, os quais são delegados aos terminais de processamento. O sistema atribui a cada processo carregado um identificador numérico sequencial, a partir do número um.

Ao finalizar o processo a ele atribuído, o terminal envia ao sistema a mensagem “delete  $p$ ”, onde  $p$  é o identificador do processo concluído. O sistema deve remover o processo finalizado do vetor e liberar o identificador para ser reutilizado na criação de novos processos.

Ao longo do processamento, um terminal pode exigir a criação de novos processos auxiliares através da mensagem “new subprocess.”. O sistema deve definir como identificador deste novo processo o menor número inteiro positivo não atribuído a nenhum processo presente no vetor, e inserir este novo processo no vetor do sistema, segundo as regras de inserção.

As operações de remoção e inserção de processos no vetor do sistema seguem as seguintes regras:

1. A inserção é feita sempre ao final do vetor, após o último processo pendente;
2. A remoção é feita através da troca dos identificadores do processo a ser removido e do processo que ocupa a última posição do vetor. Após a permuta, o último elemento é removido;
3. O vetor do sistema tem uma capacidade máxima de  $N$  processos. Caso uma inserção seja solicitada quando o vetor estiver cheio, o processo a ser inserido deve ser descartado;
4. Também deverá ser ignorada uma tentativa de remoção quando o vetor de processos estiver vazio ou o identificador do processo a ser removido não estiver presente no vetor.

Neste contexto, dada a capacidade máxima do vetor, a quantidade de processos carregados na inicialização e as requisições de inserção e remoção, determine o estado final do vetor.

### Entrada

A duas primeiras linhas da entrada contém a capacidade máxima  $N$  ( $1 \leq N \leq 1.000$ ) do vetor e o número  $M$  ( $1 \leq M \leq N$ ) de processos a serem carregados na inicialização do sistema, respectivamente.

A terceira linha contém o número  $R$  ( $0 \leq R \leq 1.000$ ) de requisições de inserção e remoção recebidas pelo sistema. As próximas  $R$  linhas contém as mensagens destas requisições, na ordem recebida pelo servidor, conforme descritas anteriormente.

## Saída

A saída do programa deverá ser a mensagem “ $V = [id_1, id_2, \dots, id_T]$ ”, onde  $id_j$  é o identificador do  $j$ -ésimo processo no vetor. Cada identificador deve ser seguido de uma vírgula e um espaço em branco, exceto o último, e o conjunto de identificadores deve estar entre colchetes. Ao final da mensagem deve ser impressa uma quebra de linha.

Exemplos de entradas	Exemplos de saídas
10 3 0	$V = [1, 2, 3]$
10 4 2 delete 2. new subprocess.	$V = [1, 4, 3, 2]$
4 3 5 new subprocess. new subprocess. delete 5. delete 1. delete 4.	$V = [3, 2]$

*Este problema foi elaborado para ensino e docência. Quaisquer coincidências com problemas já existentes favor entrar em contato (edsonalves@unb.br) para que as devidas providências sejam tomadas.*