Filas e Pilhas
Prof. Wellington S. Martins

Conteúdo

1 Fila do SUS **2**

2 acampamento **3**

3 JFILA14 - Fila **4**

4 Anões **5**

5 Estratégia **7**

6 Balanço de Parênteses **9**

7 Expressões **10**

8 Trilhos **12**

9 Caminho **13**

10 Polonesa **14**

1 Fila do SUS

 (++)

Os pacientes que chegam na fila do SUS passam por uma triagem imediatamente e vão para a fila de atendimento. Na triagem a enfermeira anota o horário de entrada do paciente e quantos minutos ele tem até que sua condição de saúde se torne crítica. Sabe-se que os pacientes são atendidos de 30 em 30 minutos (sempre nas horas cheias ou meias horas) quando na fila de atendimento. O inicio da triagem e do atendimento se dá às 7h da manhã, se não há nenhum paciente sendo atendido e a fila está vazia, o primeiro paciente é atendido no instante que chega na triagem. O médico atende até o último paciente na fila. A preocupação é se algum paciente atingiu uma condição crítica enquanto não tenha sido atendido. Para tanto você foi convidado para verificar na fila quantos pacientes atingem a condição crítica.

Entrada

A entrada começa com uma linha com o número inteiro N, $0 < N < 25$; o número de pacientes que chegam à triagem. A seguir são N linhas com os valores inteiros H, M e C, com $7 < H < 19$, e $0 \leq M < 60$, a hora e minuto que o paciente chega à triagem. O paciente da linha i sempre chega antes que, e no máximo junto com, o paciente da linha i + 1. E $0 \leq C \leq 720$ o número de minutos antes do paciente atingir a condição crítica de saúde.

Saída

Imprima o número de pacientes que atingiram a condição crítica ainda na fila de atendimento.

Exemplo

Entrada	Saída
4 7 0 20 7 0 30 7 30 20 8 15 30	1

Entrada	Saída
5 10 20 50 10 30 30 11 10 20 12 0 0 12 10 30	0

2 acampamento

 (++++)

No primeiro dia de acampamento, devido à forte chuva, as atividades recreativas ficaram limitadas e as crianças foram levadas para o ginásio de esportes. Foi realizada uma gincana e uma das atividades da mesma consistiu em agrupar as crianças em um círculo (organizado no sentido anti-horário) do qual seriam retiradas uma a uma até que sobrasse apenas uma criança, que seria a vencedora.

No momento em que entra no círculo, cada criança recebe uma pequena ficha que contém um valor de 1 a 500. Depois que o círculo é formado, conta-se, iniciando na criança que está ao lado da primeira que entrou no círculo , o número correspondente à ficha que a primeira detém. A criança onde o número contado cair, deve ser retirada do grupo, e a contagem inicia novamente da primeira criança que entrou, segundo o número da ficha da criança que acabou de ser eliminada. Para ficar mais interessante, quando o valor que consta na ficha é par, a contagem é feita no sentido horário e quando o valor que consta na ficha é ímpar, a contagem é feita no sentido anti-horário.

(Obs: como a contagem começa do que está ao lado, isso muda se for par ou ímpar, par a imediatamente ao lado no sentido horário, e se for ímpar no anti-horário).

Se a pessoa que entrou primeiro for retirada, assume seu posto a pessoa que entrou imediatamente depois (a segunda pessoa que entrou), se esta também for retirada, assume a que entrou imediatamente depois dela (a terceira pessoa que entrou), assim por diante.

A brincadeira fez muito sucesso e o administrador do acampamento pediu para que sua equipe desenvolva um programa para que no próximo evento ele saiba previamente qual criança irá ser a vencedora de cada grupo, com base nas informações fornecidas.

Entrada

A primeira linha de cada caso de teste contém um inteiro N $(1 \leq N \leq 100)$, indicando a quantidade de crianças que farão parte de cada círculo e participarão da brincadeira. Em seguida, as N linhas de cada caso de teste conterão duas informações, o Nome e o Valor $(1 \leq Valor \leq 500)$ que consta na ficha de cada criança, separados por um espaço, na ordem de entrada na formação do círculo inicial.

Obs: O Nome de cada criança não deverá ultrapassar 30 caracteres e é usado o caractere "_" para nomes espaço.

Saída

Apresente a mensagem Vencedor(a) : xxxxxx, com um espaço após o sinal ":"indicando qual é a criança do grupo que venceu a brincadeira.

Exemplo

Entrada	Saída
5 Maria 7 Pedro 9 Joao_Vitor 5 Isabel 12 Laura 8	Vencedor(a) : Isabel

3 JFILA14 - Fila

(+) Com a proximidade da Copa do Mundo, o fluxo de pessoas nas filas para compra de ingressos aumentou consideravelmente. Como as filas estão cada vez maiores, pessoas menos pacientes tendem a desistir da compra de ingressos e acabam deixando as filas, liberando assim vaga para outras pessoas. Quando uma pessoa deixa a fila, todas as pessoas que estavam atrás dela dão um passo a frente, sendo assim nunca existe um espaço vago entre duas pessoas. A fila inicialmente contém N pessoas, cada uma com um identificador diferente. Joãozinho sabe o estado inicial dela e os identificadores em ordem das pessoas que deixaram a fila. Sabendo que após o estado inicial nenhuma pessoa entrou mais na fila, Joãozinho deseja saber o estado final da fila.

Entrada

A primeira linha contém um inteiro N representando a quantidade de pessoas inicialmente na fila. A segunda linha contém N inteiros representando os identificadores das pessoas na fila. O primeiro identificador corresponde ao identificador da primeira pessoa na fila. É garantido que duas pessoas diferentes não possuem o mesmo identificador. A terceira linha contém um inteiro M representando a quantidade de pessoas que deixaram a fila. A quarta linha contém M inteiros representando os identificadores das pessoas que deixaram a fila, na ordem em que elas saíram. É garantido que um mesmo identificador não aparece duas vezes nessa lista.

Saída

Seu programa deve imprimir uma linha contendo N - M inteiros com os identificadores das pessoas que permaneceram na fila, em ordem de chegada.

Restrições

- $1 \leq N \leq 50000$
- $1 \leq M \leq 50000$ e $M < N$
- Cada identificador está entre 1 e 100000

Exemplo

Entrada	Saída
8 5 100 9 81 70 33 2 1000 3 9 33 5	100 81 70 2 1000

Entrada	Saída
4 10 9 6 3 1 3	10 9 6

4 Anões

 (++)

Branca de Neve e os N anões vivem na floresta. Enquanto os anões trabalham em sua mina, Branca de Neve desperdiça o seu tempo em redes sociais. Todas as manhãs, os anões formam uma longa fila e vão assobiando para a mina. Branca de Neve corre ao redor deles e faz um monte de fotografias, para fazer o *upload* para a sua rede social favorita. Quando os anões entram na mina, Branca de Neve volta para casa e passa a selecionar as fotos mais bonitas. Cada anão tem uma touca colorida, e há C cores diferentes. Para Branca de Neve a imagem é bonita se mais da metade da toucas são da mesma cor. Em outras palavras, se houver K anões na imagem, ela é bonita se estritamente mais do que $K/2$ anões têm as toucas de mesma cor. Escreva um programa que irá verificar, para um conjunto de M fotos, se elas são bonitos e, neste caso, qual a cor predominante.

Entrada

A primeira linha da entrada conterá um inteiro T, o número de casos de teste, que virão na sequência. A primeira linha de cada caso de teste conterá dois inteiros N e C ($3 \leq N \leq 300000$, $1 \leq C \leq 10000$), o número de anões e o número de cores. A segunda linha conterá N inteiros entre 1 e C (inclusive), as cores das toucas dos anões, ordenados segundo a fila que eles formaram naquela manhã. A terceira linha conterá um inteiro M ($1 \leq M \leq 10000$), o número de fotos. As M linhas seguintes conterão dois inteiros A e B ($1 \leq A \leq B \leq N$). Cada linha descreve uma foto, a qual contém todos os anões a partir do Aésimo até o B-ésimo.

Saída

A saída deve conter, para cada caso de teste, M linhas de saída. Para cada imagem escreva "Nao" se a Branca de Neve não achou que a foto é bonita, e "Sim X", onde X é a cor predominante nas toucas dos anões na imagem, se ela achou a foto bonita.

Exemplo

Entrada:
1 10 3 1 2 1 2 1 2 3 2 3 3 8 1 2 1 3 1 4 1 5 2 5 2 6 6 9 7 10

Saída:
Nao Sim 1 Nao Sim 1 Nao Sim 2 Nao Sim 3

5 Estratégia

 (++)

João, José e Joaquim gostam de participar de competições de programação. Eles têm estratégias diferentes e nós gostaríamos de saber qual estratégia é a melhor.

João simplesmente resolve os problemas na ordem em que os recebe dos organizadores do concurso. José primeiro lê todos os os problemas e os resolve em ordem crescente de dificuldade. Joaquim também lê todos os problemas em primeiro lugar, mas ele é muito ambicioso e, portanto, os resolve em ordem decrescente de dificuldade.

A dificuldade de um problema é medida em quantidade de minutos que os rapazes levam para resolver o problema. Reunimos estatísticas e consultamos o oráculo Thierson, assim nós sabemos, para todos os tipos de problemas, quanto tempo os rapazes vão necessitar. Nós também descobrimos que, para cada problema, os três sempre precisam do mesmo tempo (que depende da dificuldade do problema). Assim, eles diferem apenas por suas estratégias.

Para diversas competições, nós gostaríamos que você nos dissesse o vencedor, o número de problemas resolvidos e qual sua pontuação. A pontuação para um único problema é o tempo em minutos desde o início do concurso até a sua resolução. O quadro geral de pontuação é a soma das pontuações dos problemas resolvidos. Os rapazes nunca erram, assim você não tem que lidar com penalidades. O vencedor é aquele que resolver mais problemas e, em caso de empate, aquele com a menor pontuação. Se ainda houver um empate, os três concordam que José seja o vencedor, pois ele sempre traz deliciosos bolos de arroz para os demais.

Entrada

A primeira linha da entrada conterá um inteiro T, o número de casos de teste. Cada caso de teste descreve um concurso e sua primeira linha diz quanto tempo dura a competição (de 30 a 1440 minutos) e o número de problemas (de 3 a 24). Em uma segunda linha você vai ter as dificuldades dos problemas, como explicado acima eles dizem quantos minutos (de 1 a 600) os rapazes precisam para resolver os problemas.

Saída

A saída deve conter, para cada caso de teste, uma linha contendo "Cenario #i:", onde i é o número do caso de teste, a partir de 1. Em seguida, deve ser impressa uma única linha dizendo quem ganha, o número de problemas que ele resolve e sua pontuação. Use o formato exato, conforme mostrado abaixo no exemplo de saída, mesmo que o vencedor só resolva 0 ou 1 problema. Termine a saída de cada caso de teste com uma linha em branco.

Exemplo

Entrada:
2 180 6 23 42 170 33 7 19 60 2 43 17

Saída:
Cenario #1: Jose ganha, com 5 problemas resolvidos e pontuacao de 288. Cenario #2: Jose ganha, com 2 problemas resolvidos e pontuacao de 77.

6 Balanço de Parênteses

 (++)

Dada uma expressão qualquer com parênteses, indique se os parênteses estão corretos ou não, sem levar em conta o restante da expressão. Por exemplo:

a+(b*c)-2-a está correto (a+b*(2-c)-2+a)*2 está correto

enquanto

(a*b-(2+c) está incorreto 2*(3-a)) está incorreto)3+b*(2-c)(está incorreto

Ou seja, todo parênteses que fecha deve ter um outro parênteses que abre correspondente e não pode haver parênteses que fecha sem um previo parenteses que abre e a quantidade total de parenteses que abre e fecha deve ser igual.

Obs: a estrutura de dados deve ser implementada de forma dinâmica.

Entrada

A entrada contém uma expressão de no máximo 1000 caracteres.

Saída

A saída deverá conter Correta se estiver correta e Incorreta e estiver incorreta.

Exemplo

Entrada	Saída
`)3+b*(2-c)(`	`Incorreta`

7 Expressões

(++++) Pedrinho e Zezinho estão precisando estudar resolução de expressões matemáticas para uma prova que irão fazer. Para isso, eles querem resolver muitos exercícios antes da prova. Como sabem programar, então decidiram fazer um gerador de expressões matemáticas.

O gerador de expressões que eles criaram funciona em duas fases. Na primeira fase é gerada uma cadeia de caracteres que contém apenas os caracteres '{', '[', '(', '}', ']' e ')'. Na segunda fase, o gerador adiciona os números e operadores na estrutura criada na primeira fase. Uma cadeia de caracteres é dita bem definida (ou válida) se atende as seguintes propriedades:

1. Ela é uma cadeia de caracteres vazia (não contém nenhum caractere).
2. Ela é formada por uma cadeia bem definida envolvida por parênteses, colchetes ou chaves. Portanto, se a cadeia S é bem definida, então as cadeias (S), $[S]$ e $\{S\}$ também são bem definidas.
3. Ela é formada pela concatenação de duas cadeias bem definidas. Logo, se as cadeias X e Y são bem definidas, a cadeia XY é bem definida.

Depois que Pedrinho e Zezinho geraram algumas expressões matemáticas, eles perceberam que havia algum erro na primeira fase do gerador. Algumas cadeias não eram bem definidas. Eles querem começar a resolver as expressões o mais rápido possível, e sabendo que você é um ótimo programador, resolveram pedir que escreva um programa que dadas várias cadeias geradas na primeira fase, determine quais delas são bem definidas e quais não são.

Entrada

A entrada é composta por diversas instâncias. A primeira linha da entrada contém um inteiro T indicando o número de instâncias. Em seguida temos T linhas, cada uma com uma cadeia A.

Saída

Para cada instância imprima uma linha contendo a letra S se a cadeia é bem definida, ou a letra N caso contrário.

Restrições

- $1 \leq T \leq 20$
- a cadeia de caracteres A tem entre 1 e 100000 caracteres.
- a cadeia de caracteres A contém apenas caracteres '{', '[', '(', '}', ']' e ')'.

Exemplo

Entrada	Saída
12 () [] {} (] }{ ([{}]) {}()[] ()) {[] ((({}{}()[[]])(){}){} ((((((((((){([])}]))))))))))	S S S N N S S N N N S N

8 Trilhos

 (++)

Há uma famosa estação de trem na cidade PopPush. Esta cidade fica em um país incrivelmente acidentado e a estação foi criada no último século. Infelizmente os fundos eram extremamente limitados naquela época. Foi possível construir somente uma pista. Além disso, devido a problemas de espaço, foi feita uma pista apenas até a estação.

A tradição local é que todos os comboios que chegam vindo da direção A continuam na direção B com os vagões reorganizados, de alguma forma. Suponha que o trem que está chegando da direção A tem N $\leq$ 1000 vagões numerados sempre em ordem crescente 1,2, ..., N. O primeiro que chega é o 1 e o último que chega é o N. Existe um chefe de reorganizações de trens que quer saber se é possível reorganizar os vagões para que os mesmos saiam na direção B na ordem a1, a2, an..

O chefe pode utilizar qualquer estratégia para obter a saída desejada. Por exemplo se o chefe quer saber se a saída 5,4,3,2,1 é possível em B, nesse caso, basta o chefe deixar todos os vagões entrarem na estação (do 1 ao 5) e depois retirar um a um: retira o 5, retira o 4, retira o 3, retira o 2 e por último retira o 1. Desta forma a resposta seria Yes. Vagão que entra na estação só pode sair para a direção B e é possível incluir quantos forem necessários para retirar o primeiro vagão desejado.

Entrada

Na primeira linha de cada bloco há um inteiro N que é a quantidade de vagões. Em cada uma das próximas linhas de entrada haverá uma permutação dos valores 1,2,. . . , N.

Saída

A saida deve conter Yes se for possível e No se não for.

Exemplo

Entrada	Saída
5 5 4 3 2 1	Yes

Entrada	Saída
5 5 4 1 2 3	No

9 Caminho

 (++)

A família de João acaba de se mudar para uma nova cidade e hoje é o seu primeiro dia de escola. Ele tem uma lista de instruções para andar de sua casa até a escola. Cada instrução descreve uma curva que ele deve fazer. Por exemplo, a lista:

D
QUINZE
D
F
D
ESCOLA

significa que ele deve virar à direita na rua Quinze, em seguida, virar à direita na rua F e, finalmente, virar à direita na escola. Sua tarefa é escrever um programa que irá criar instruções para caminhar na direção oposta: a partir de sua escola até sua casa. Você pode assumir que a lista de João contém pelo menos duas, mas no máximo cinco instruções. Você também pode assumir que cada linha contém no máximo 10 caracteres (letras maísculas). A última instrução será sempre virar para a escola.

Entrada

A entrada conterá um inteiro T, o número de casos de testes, e, para cada caso de teste, uma sequência de pares de linhas com a lista de instruções para o caminho de casa até a escola. A primeira linha de cada par é a instrução de virar à direita "D" ou à esquerda "E" e a segunda linha é o nome da rua. A última linha de cada caso de teste contém a palavra "ESCOLA".

Saída

A saída deve conter, para cada caso de teste, uma sequência de linhas que especifiquem o caminho de volta da escola até a casa de João.

Exemplo

Entrada:
2 D QUINZE D QUATRO D ESCOLA E PRINCIPAL D ESCOLA

Saída:
Vire a ESQUERDA na rua QUATRO. Vire a ESQUERDA na rua QUINZE. Vire a ESQUERDA na sua CASA. Vire a ESQUERDA na rua PRINCIPAL. Vire a DIREITA na sua CASA.

10 Polonesa

 (++)

Usualmente uma express ao aritmética é escrita com os operadores entre os respectivos operandos (no caso de operadores binários), por isso essa notação é chamada de *Infixa*. Uma forma alternativa de escrever uma expressão aritmética é utilizando-se a *Notação Polonesa Inversa* ou *Notação Posfixa*. Essa notação foi inventada pelo filósofo e cientista da computação australiano Charles Hamblin em meados dos anos 1950 com base na *Notação Polonesa*, introduzida em 1920 pelo matemático polonês Jan Lukasiewicz.

A tabela a seguir mostra alguns exemplos de operações e notações nessas duas notações:

Operação	Notação convencional	Notação polonesa inversa
$a+b$	$a+b$	$a\ b\ +$
$\frac{a+b}{c}$	$(a+b)/c$	$a\ b+c\ /$
$\frac{a\cdot b-c\cdot d}{e\cdot f}$	$((a*b)-(c*d))/(e*f)$	$a\ b*c\ d\ *-\ e\ f*/$

Note-se que a ordem dos operandos é a mesma nas notações infixa e posfixa. Além disso, a notação posfixa dispensa parênteses.

A sua tarefa é construir um programa para traduzir expressões escritas com a infixa para a notação posfixa. As expressões são formadas por constantes numéricas, variáveis e operadores aritméticos. Para simplificar, considere que na expressão infixa são usadas apenas as operações aritméticas de soma, subtração, divisão, multiplicação e exponenciação ($+$, $-$, $/$, $*$ e ^). Considere também que nomes de variáveis são formados por apenas uma letra maíscula (A . . Z), as constantes numéricas são formadas por apenas um dígito (1 . . 9) e não há espaços antes ou após um operador aritmético, no início ou no final da express ao. Se para cada símbolo de "abre parênteses" em uma expressão na notação infixa há um correspondente símbolo de "fecha parênteses", diremos que a expressão está bem formada.

Entrada

Os dados de entrada são compostos por vários casos de teste. Cada caso de teste contém uma expressão aritmética escrita na notação infixa, com $0 \leq n \leq 10^2$ operadores.

Saída

Para cada caso de teste, imprima a expressão na notação infixa e a correspondente expressão aritmética escrita na notação posfixa (ou o termo "Expressao invalida" caso a expressão na notação infixa esteja mal formada). Imprima uma linha em branco após cada caso de teste.

Exemplo

Entrada:
O/4 (E/5+(8)^U+4)+7^(9)*2 5*4/(M+8+T−B*8^T^T Q*H*S/2*(Q/(J)^Y*C

Saída:
O/4 O 4 / (E/5+(8)^U+4)+7^(9)*2 E 5 / 8 U ^ + 4 + 7 9 ^ 2 * + 5*4/(M+8+T−B*8^T^T Expressao invalida Q*H*S/2*(Q/(J)^Y*C Expressao invalida