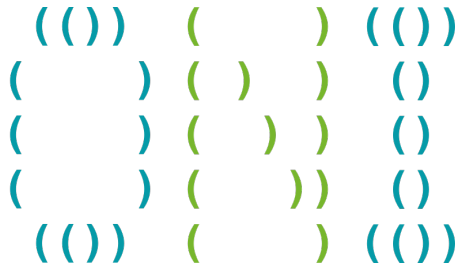
Problema B - Remoção de Parênteses

O João está a desenvolver um novo processador de texto que usa pares de parênteses '(' e ')' para definir zonas de itálico e negrito no texto. Para se diferenciar dos produtos no mercado, o João está a tentar implementar novas operações para ajudar os utilizadores a apagar texto, que serão especificadas nas partes seguintes.



Serão dadas $\boldsymbol{T}$ strings, a i-ésima das quais terá tamanho $\boldsymbol{N_i}$ e contém apenas parênteses '(' e ')'. Em cada parte abaixo, deverás indicar como o João pode aplicar a sua operação de remoção para limpar cada string por completo.

Parte I

Para esta parte podes realizar a seguinte operação múltiplas vezes sobre uma string:

- Escolhe uma posição *entre* dois caracteres da string.
- Seleciona o primeiro '(' à esquerda dessa posição.
- Seleciona o primeiro ')' à direita dessa posição.
- Remove ambos os parênteses e todos os caracteres entre eles.

A operação só pode ser aplicada se houver um '(' à esquerda e um ')' à direita da posição escolhida.

Para cada string, determina se a é possível remover completamente recorrendo apenas a essa operação.

Exemplo

Suponhamos que $\boldsymbol{N} = 4$ e a string é `(())`. Podemos escolher a posição entre o 2.º e o 3.º parênteses: essa operação remove o 2.º e o 3.º parênteses, restando `()`. De seguida, escolhemos a posição entre os restantes parênteses que são assim removidos e deixando a string vazia. Logo, a resposta é sim.

Restrições

São garantidos os seguintes limites em todos os casos de teste desta parte que irão ser colocados ao programa:

$1 \leq \boldsymbol{T} \leq 20$	Número de strings
$1 \leq \boldsymbol{N_i} \leq 1\,000$	Tamanho de cada string

Os casos de teste desta parte do pertencem todos a um grupo:

Grupo	Número de Pontos	Restrições adicionais
1	20	Sem restrições adicionais

Parte II

Para esta parte deves *imprimir* uma sequência de posições que se existir, ao serem escolhidas nessa ordem, resulta na eliminação completa da string usando a operação descrita na Parte I.

As posições são indexadas entre os caracteres, como no seguinte exemplo para `()()`:

índice das posições entre caracteres

1 2 3 4 5

( ) ( )

1 2 3 4

índice dos caracteres

Após cada operação, a nova string resultante redefine as futuras numerações de posições.

Exemplo

Para $\boldsymbol{n} = 6$ e string `(())()`. Uma sequência válida de posições é 3, 4, 2, pois (os `|` indicam a posição a remover):

- posição 3: `((|))()` → remove interior → `()()`;
- posição 4: `()(|)` → remove par direito → `()`;
- posição 2: `(|)` → remove ambos → vazio.

Restrições

São garantidos os seguintes limites em todos os casos de teste desta parte que irão ser colocados ao programa:

$1 \leq \boldsymbol{T} \leq 20$	Número de strings
$1 \leq \boldsymbol{N_i} \leq 1\,000$	Tamanho de cada string

Os casos de teste desta parte do problema estão organizados em dois grupos com restrições adicionais diferentes:

Grupo	Número de Pontos	Restrições adicionais
2	15	$\boldsymbol{N_i} \leq 100$
3	15	Sem restrições adicionais

Parte III

Para esta parte considera agora que tens um *cursor* que:

- Começa na posição 1 (à esquerda de toda a string).
- Pode mover-se uma posição de cada vez para a direita (entre 1 e $N + 1$).
- Pode apagar o carácter imediatamente à esquerda do cursor.
- Pode executar a operação da Parte I para a posição atual do cursor.

Agora é sempre possível eliminar a string toda, mas o João quer poupar tempo. Determina o número mínimo de operações (movimentos e remoções) necessárias.

Exemplo

Para $N = 5$ e string `)()()`. O cursor começa na posição 1, ou seja, `|)()()`

1. Move-se para a direita (cursor em 2), resultando em `)|()()`
2. Apaga o carácter à esquerda (`)`), resultando em `|()()`
3. Move-se para a direita (cursor em 2), resultando em `(|)()`
4. Move-se para a direita (cursor em 3), resultando em `()|()`
5. Executa a operação da Parte I que remove a string completa.

Total de **5** operações.

Restrições

São garantidos os seguintes limites em todos os casos de teste desta parte que irão ser colocados ao programa:

$1 \leq T \leq 20$	Número de strings
$1 \leq N_i \leq 1\,000$	Tamanho de cada string

Os casos de teste desta parte do problema estão organizados em dois grupos com restrições adicionais diferentes:

Grupo	Número de Pontos	Restrições adicionais
4	10	$N_i \leq 10$
5	20	$N_i \leq 100$
6	20	Sem restrições adicionais

Sumário de subtarefas

Os casos de teste do problema estão organizados em seis grupos com restrições adicionais diferentes:

Grupo	Número de Pontos	Parte	Restrições adicionais
1	20	Parte I	Sem restrições adicionais
2	15	Parte II	$\boldsymbol{N_i} \leq 100$
3	15	Parte II	Sem restrições adicionais
4	10	Parte III	$\boldsymbol{N_i} \leq 10$
5	20	Parte III	$\boldsymbol{N_i} \leq 100$
6	20	Parte III	Sem restrições adicionais

Formato de Input

A primeira linha contém um inteiro $\boldsymbol{P}$, a parte do problema (pode ser 1, 2 ou 3).

A segunda linha contém um inteiro $\boldsymbol{T}$, o número de strings a eliminar.

Seguem-se $\boldsymbol{T}$ pares de linhas, sendo que o i-ésimo par é composto por:

- Uma linha com um inteiro $\boldsymbol{N_i}$, o comprimento da i-ésima string.
- Uma linha com uma string de comprimento $\boldsymbol{N_i}$, contendo apenas '`(`' e '`)`'.

Formato de Output

O output deve conter $\boldsymbol{T}$ linhas, com o seguinte conteúdo:

Parte I

Imprime `SIM` se for possível remover completamente a string, ou `NAO` caso contrário.

Parte II

Imprime uma linha com um inteiro $\boldsymbol{K}$ (número de operações) e, na linha seguinte, $\boldsymbol{K}$ inteiros - as posições escolhidas, na ordem em que as operações são executadas.

Se não for possível eliminar completamente a string, deves imprimir apenas o inteiro -1.

Parte III

Imprime um único inteiro: o número mínimo de operações (movimentos do cursor e remoções) necessários para eliminar a string.

Input do Exemplo 1

```
1
2
4
(())
4
))()
```

Output do Exemplo 1

```
SIM
NAO
```

Explicação do Exemplo 1

Este exemplo corresponde ao exemplo da Parte I mencionado no enunciado.

Input do Exemplo 2

```
2
2
6
(())()
4
((((
```

Output do Exemplo 2

```
3
3 4 2
-1
```

Explicação do Exemplo 2

Este exemplo corresponde ao exemplo da Parte II mencionado no enunciado.

Input do Exemplo 3

```
3
1
5
)()()
```

Output do Exemplo 3

```
5
```

Explicação do Exemplo 3

Este exemplo corresponde ao exemplo da Parte III mencionado no enunciado.

Organização

APDSI U.PORTO FACULDADE DE CIÊNCIAS UNIVERSIDADE DO PORTO PRINCETON UNIVERSITY

Alto Patrocínio

Com o Alto Patrocínio de Sua Excelência O Presidente da República

REPÚBLICA PORTUGUESA EDUCAÇÃO, CIÊNCIA E INOVAÇÃO

Patrocinadores

CIÊNCIA VIVA Agência Nacional para a Cultura Científica e Tecnológica

FUNDAÇÃO CALOUSTE GULBENKIAN