Revisar paréntesis

Límite de tiempo: 1s Límite de memoria: 1 GB

Enunciado

Mi jefe está cansado de leer mis demostraciones matemáticas. Dice que siempre escribo unos paréntesis (incluyendo corchetes y llaves) terribles. Según mi jefe, ninguna fórmula matemática debería tener k o más paréntesis anidados, y por supuesto, deben estar bien balanceados.

Dada una fórmula s y un entero positivo k, hay que responder:

- `MAL` si los paréntesis de la fórmula no están balanceados,
- `DEMASIADO PROFUNDO` si están balanceados pero la profundidad máxima de anidamiento es $\geq k$,
- `BIEN` en cualquier otro caso.

Solo importan los paréntesis de tipo `([{` y `}])`. Cualquier otro carácter de la fórmula se ignora. Los paréntesis están balanceados si cada paréntesis que se abre se cierra después con uno del mismo tipo, y los pares no se cruzan: `([])` está balanceada, pero `([)]` no.

La profundidad de anidamiento en un punto de la fórmula es el número de paréntesis abiertos que aún no se han cerrado. Por ejemplo, `([a]+{b})` tiene profundidad máxima 2.

Entrada

La primera línea contiene un entero t, el número de fórmulas a revisar.

Cada fórmula ocupa dos líneas: la primera con el entero k y la segunda con la cadena s. La cadena puede contener espacios, pero no empieza ni termina por espacio.

Salida

Para cada fórmula, imprime una línea con `MAL`, `DEMASIADO PROFUNDO` o `BIEN`.

Restricciones

$1 \leq t \leq 1000, \quad 1 \leq k \leq 10^6, \quad 1 \leq |s|$.

La suma de las longitudes de todas las cadenas no supera $2 \cdot 10^5$.

Ejemplo

Entrada de ejemplo

```
4
2
((a+b)/c
10
([a+b)+c]
2
([a]+{b})
3
x + [y*(z+3)]+ {1+a}
```

Salida de ejemplo

```
MAL
MAL
DEMASIADO PROFUNDO
BIEN
```

Nota

La primera fórmula deja un paréntesis sin cerrar. En la segunda, el `)` cierra un `[`, así que los pares se cruzan. La tercera está balanceada pero llega a profundidad $2 \geq k$. La cuarta llega a profundidad $2 < 3$.