

Validando Expresiones

Descripción del problema

Un profesor de matemática elemental está enseñando a sus alumnos como construir expresiones aritméticas. Básicamente, los conceptos a remarcar son el uso de operadores binarios y el de llaves, paréntesis y corchetes.

Como tiene una cantidad enorme de alumnos por sala y muchos horarios en el día, para simplificar la tarea, te pide que le construyas un programa **VALIDAR** en C, C++ o Pascal que verifique si la expresión tiene un error de sintaxis. En caso que esto ocurra, debemos informar el primer carácter que lo evidencie.

Nota:

Los operadores e identificadores de las expresiones están formados por un solo carácter.

Los únicos operadores validos son: suma("+"), resta binaria ("-"), Multiplicación ("*") y división ("/").

Los operandos, sean variables o números, están formados por un solo carácter alfanumérico (de "a" la "z" o de la "A" a la "Z" mayúscula o del "0" al "9")

Datos de entrada

Se recibe por la entrada estándar:

- Primera línea: el número n ($1 \leq n \leq 1000$) que indica la longitud máxima de la expresión.
- Segunda línea: n caracteres correspondientes a la expresión que se debe examinar.

Datos de salida

El programa debe imprimir en la salida estándar el número que representa la posición del primer carácter erróneo en la expresión (se define que las posiciones de la expresión se enumeran de 1 a n)

En caso de que la expresión sea correcta, se debe imprimir "ENTRADA CORRECTA"

Ejemplo 1

En el caso de que la entrada fuera:

```
22
3+{6*s/(3-(4+5)/A))*9}
```

La salida debería ser:

```
19
```

Ejemplo 2

En el caso de que la entrada fuera:

```
21
3+{6*/(3-(4+5)/A))*9}
```

La salida debería ser:

```
6
```

Ejemplo 3

En el caso de que la entrada fuera:

```
30
3+{6*[3-(4+5S)/A*9]}ccccccccccc
```

La salida debería ser:

```
13
```