

# Visita de John McCarthy

Tus esfuerzos por meter a tus padres en el mundo de la computación va dando sus frutos. Y no sólo eso; el hecho de que el ENIAC haya sido donado al pueblo de Hill Valley hace que personas que dentro de unos años van a ser eminencias pasen por aquí para verlo.



Un ejemplo es el mítico John McCarthy que, entre otras cosas, acuñará el término de “inteligencia artificial” en una conferencia el año que viene (en verano de 1956). Hace sólo 4 años que terminó su tesis y hasta que en 1962 pase a ser profesor de la Universidad de Standford, anda moviéndose de un sitio a otro y se ha pasado por aquí.

Ha sido una suerte que haya venido a dar una charla al instituto. Se ha ganado al público contando cómo inventó un exprimidor de naranjas hidráulico, mientras tú te morías de ganas por preguntarle sobre sus desarrollos futuros. De todas formas, te ha parecido curioso que haya estado hablando sobre las distintas formas de escribir expresiones matemáticas y haya mencionado la notación *prefija* que será la forma utilizada en el LISP que inventará dentro de 3 años.

Tras la charla, tus padres quieren aprender a evaluar ese tipo de expresiones, así que te toca estudiarlas a tí antes de que te pregunten.

En las expresiones escritas en notación prefija los operadores se colocan antes que los operandos. Así, por ejemplo, lo que en la notación tradicional (infija) escribiríamos como  $3 + 4$ , se escribe como  $+ 3 4$ .

Otros ejemplos son:

Notación infija	Notación prefija
$3 + 4$	$+ 3 4$
$3 - (4 \times 5)$	$- 3 \times 4 5$
$(3 + 4) \times 5$	$\times + 3 4 5$
$(3 - 4) \div (5 + 2)$	$\div - 3 4 + 5 2$

## Entrada

El programa leerá, de la entrada estándar, un número indicando la cantidad de casos de prueba que tendrá que procesar. Cada uno, en una línea, contendrá una expresión en notación polaca. Por simplicidad, los números tendrán únicamente un dígito. Los operadores válidos serán suma (+), resta (-), multiplicación (\*) y división (/). El primer operando se proporciona antes que el segundo, y la división será entera.

## Salida

Para cada caso de prueba el programa deberá escribir, en una línea independiente, el valor de la expresión. Se garantiza que tanto los resultados parciales como los finales serán menores que  $2^{31}$  en valor absoluto, y nunca habrá divisiones por 0.

## Entrada de ejemplo

```
4
+ 3 4
- 3 * 4 5
* + 3 4 5
/ - 3 4 + 5 2
```

## Salida de ejemplo

```
7
-17
35
0
```

## Nota

Este ejercicio debe verse en el contexto de la asignatura de Estructura de Datos y Algoritmos (EDA), FDI-UCM 2018/2019 (prof. Marco Antonio Gómez Martín). Por tanto *no* vale cualquier solución, sino sólo aquellas que utilicen los conceptos de EDA. Es muy posible que se den aclaraciones adicionales en clase a este respecto.