

# Postfix Evaluation

Batas Waktu	1 detik
Batas Memori	16 MB

Pak Agri baru saja menemukan sebuah kalkulator tua. Kalkulator tersebut memiliki cara yang aneh untuk melakukan perhitungan, yaitu bahwa kalkulator tersebut hanya menerima operasi yang dituliskan menggunakan notasi *postfix*. Notasi *postfix* berbeda dengan notasi *infix* yang biasa digunakan sehari-hari, dimana operator diletakkan di tengah-tengah kedua bilangan yang dioperasikan. Pada notasi *postfix* dituliskan terlebih dahulu kedua bilangan yang dioperasikan, kemudian baru operatornya. Sebagai contoh, jika kita biasa menuliskan "5+7" dengan notasi *infix*, maka pada notasi *postfix* kita tuliskan "57+". Berikut adalah definisi formal notasi *postfix*.

- Sebuah bilangan sembarang  $n$  adalah sebuah ekspresi *postfix*, dengan nilai =  $n$  itu sendiri.
- Jika  $x$  dan  $y$  adalah dua string yang merupakan ekspresi *postfix* dengan nilai masing-masing adalah  $val\_x$  dan  $val\_y$ , maka ekspresi berikut adalah juga ekspresi *postfix*:
  - $xy+$  dengan nilai  $val\_x + val\_y$
  - $xy*$  dengan nilai  $val\_x * val\_y$
  - $xy/$  dengan nilai  $val\_x / val\_y$
  - $xy-$  dengan nilai  $val\_x - val\_y$

Sebagai contoh ekspresi *postfix* "35+" bernilai 8 sedangkan "42+3/" bernilai 2. Keunggulan dari notasi *postfix* adalah bahwa notasi ini tidak memerlukan penggunaan kurung seperti halnya pada *infix*, sebagai contoh, pada notasi *infix*  $3+5*4=23$  sedangkan  $(3+5)*4=32$ . Pada notasi *postfix*, keduanya dapat ditulis sebagai ekspresi yang berbeda, yaitu  $354*+$  dan  $35+4*$ .

Pak Agri sangat tertarik untuk mempelajari notasi ini lebih lanjut, akan tetapi kalkulator tersebut terbatas jumlah operasinya. Oleh karena itu, Pak Agri meminta bantuan Anda untuk mensimulasikan kalkulator tersebut dengan komputer Anda.

## Format Masukan

Input terdiri dari satu baris string berisi sebuah ekspresi dalam notasi *postfix*. Anda dapat asumsikan bahwa bilangan pada input selalu bilangan satu digit (0...9) untuk memudahkan Anda. Artinya, input tidak pernah menuliskan bilangan negatif ataupun bilangan lebih besar dari 9. Namun, hasil dari operasi tentunya dapat bernilai negatif ataupun lebih besar dari 9. Panjang string input tidak pernah lebih dari 100 karakter.

## Format Keluaran

Keluaran adalah berupa satu baris, yaitu nilai dari ekspresi *postfix* yang diberikan. Operasi pembagian selalu dilakukan dengan pembulatan ke bawah, misalnya  $5/3=1$  dan  $9/4=2$ . Apabila ada pembagian dengan 0, maka nilai ekspresi tidak terdefinisikan, dan Anda hanya perlu mengeluarkan output: "Error: Divide By 0!". Output selalu diakhiri dengan *newline*.

## Contoh Masukan

35+4\*

## Contoh Keluaran

32

## Contoh Masukan

422-/-

## Contoh Keluaran

Error: Divide By 0!