



Postfiksna notacija

Leta 1920 je poljski matematik in logik Jan Lukasiewicz razvil formalni sistem, ki je dopuščal zapis aritmetičnih izrazov brez oklepajev. Zapis je imenoval prefiksno oziroma postfiksno notacijo, kjer so operatorji pred oziroma za operandi, za razliko od infiksne, kjer so operatorji med operandi. Na primer izraz $(4 + 5) * 6$ zapisan v infiksni notaciji, se zapiše v postfiksni notaciji kot $4\ 5\ +\ 6\ *$.

Definicija problema

Za dan aritmetični izraz zapisan v postfiksni notaciji izračunajte njegovo vrednost. Uporabimo štiri osnovne računske operacije $+$, $-$, $*$ in $/$.

Predstavitev algoritma

```
((15/(7-(1+1)))*3)-(2+(1+1))
15 7 1 1 + - / 3 * 2 1 1 + + -
  7 1 1 + - / 3 * 2 1 1 + + -    15
    1 1 + - / 3 * 2 1 1 + + -    15 7
      1 + - / 3 * 2 1 1 + + -    15 7 1
        + - / 3 * 2 1 1 + + -    15 7 1 1
          - / 3 * 2 1 1 + + -    15 7 2
            / 3 * 2 1 1 + + -    15 5
              3 * 2 1 1 + + -    3
                * 2 1 1 + + -    3 3
                  2 1 1 + + -    9
                    1 1 + + -    9 2
                      1 + + -    9 2 1
                        + + -    9 2 1 1
                          + -    9 2 2
                            -    9 4
                              5
```

V prvi vrstici je zapisan izraz v infiksni notaciji, v drugi pa je zapis tega izraza v postfiksni notaciji. Sledijo vrstice, ki prikazujejo potek računanja.

Vhodni podatki

S pomočjo ukaza `input` preberete vrstico z izrazom zapisanim v postfiksni notaciji. Zapisi števil in operatorje ločujejo presledki.

Izhodni podatki

Na standardni izhod izpišite rezultat.

Primer

Vhod 15 7 1 1 + - / 3 * 2 1 1 + + -

Izhod 5

Rešitev

```
#!/usr/bin/env python3
stack=[]

def add(x,y):
    return(x+y)
def diff(x,y):
    return(x-y)
def mult(x,y):
    return(x*y)
def div(x,y):
    return(x/y)

op={'+':add,'-':diff,'*':mult,'/':div}

line=input('racun: ')
line=line.split()

for item in line:
    if item in op.keys():
        y=stack.pop()
        x=stack.pop()
        stack.append(op[item](x,y))
    else:
        stack.append(float(item))

print(stack.pop())
```