

# Programiranje 1 — peta domača naloga

Rok za oddajo: nedelja, 1. decembra 2019, ob 23:55

## Brainfuck

### Naloga

Brainfuck je preprost programski jezik, ki ga je leta 1993 definiral Urban Müller. Napišite program, ki prebere in izvede program v tem jeziku.

Brainfuckov *program* je sestavljen iz znakov `+`, `-`, `>`, `<`, `.`, `,`, `[` in `]`, ki predstavljajo posamezne *ukaze*. Program lahko uporablja tudi *vhod* — poljubno dolgo zaporedje števil od 0 do vključno 255. Program se izvaja na hipotetičnem *stroju*, ki bere in izvršuje njegove ukaze. Ukazi rukujejo s *trakom*, neskončnim zaporedjem *celic*, ki lahko vsebujejo števila od 0 do vključno 255. Vse celice traku na začetku vsebujejo število 0. Stroj na začetku postavi *bralno-pisalno glavo* na poljubno celico traku (ni pomembno, katero, saj so vse enakovredne), nato pa prične izvajati program ukaz za ukazom. Ukazi se izvršujejo na sledeči način:

- + Če *trenutna celica* (tj. celica traku, na kateri se trenutno nahaja bralno-pisalna glava) vsebuje število 255, se njena vsebina nastavi na 0, sicer pa se poveča za 1.
- Če je vsebina trenutne celice enaka 0, se nastavi na 255, sicer pa se zmanjša za 1.
- > Prestavi bralno-pisalno glavo za eno celico v desno.
- < Prestavi bralno-pisalno glavo za eno celico v levo.
- . Izpiše vsebino trenutne celice na zaslon.
- , Prebere naslednje število z vhoda in ga vpiše v trenutno celico. Če na vhodu ni ničesar (več), v celico vpiše število 0.
- [ Če je vsebina trenutne celice različna od 0, se bo v naslednjem koraku izvršil naslednji ukaz, sicer pa ukaz, ki sledi ujemajočemu znaku `]`. S trakom in bralno-pisalno glavo se v obeh primerih ne zgodi nič.
- ] Če je vsebina trenutne celice enaka 0, se bo v naslednjem koraku izvršil naslednji ukaz, sicer pa ukaz, ki sledi ujemajočemu znaku `[`. S trakom in bralno-pisalno glavo se v obeh primerih ne zgodi nič. Pari znakov `[` in `]` so lahko poljubno gnezdeni.

Tabela 1 prikazuje izvajanje programa `++[>,-.<-].` z vhodom  $\langle 15, 0, 3 \rangle$ .

### Vhod

V prvi vrstici vhoda je podan veljaven program v brainfucku, v ostalih pa vhod programa. Program je sestavljen iz največ  $10^4$  znakov, dolžina vhoda pa je lahko poljubna. Za posamezne skrite testne primere velja sledeče:

- Primeri 1–10: Program je sestavljen samo iz znakov `+`, `-` in `.` (pika). Vhod je prazen.
- Primeri 11–20: Program vsebuje tudi znake `>`. Vhod je prazen.

Tabela 1: Primer izvajanja programa v brainfucku. Z okvirjem so označeni trenutno obravnavani ukaz, naslednji element na vhodu in trenutna celica.

Korak	Program in trenutni ukaz	Vhod po izvedbi trenutnega ukaza	Trak po izvedbi trenutnega ukaza	Izpis
	$+ + [ > , - . < - ] .$	$\boxed{15} \ 0 \ 3$	$\dots 0 \boxed{0} \ 0 \ 0 \dots$	
1	$\boxed{+} + [ > , - . < - ] .$		$\dots 0 \boxed{1} \ 0 \ 0 \dots$	
2	$+ \boxed{+} [ > , - . < - ] .$		$\dots 0 \boxed{2} \ 0 \ 0 \dots$	
3	$+ + \boxed{[} > , - . < - ] .$			
4	$+ + [\boxed{>} , - . < - ] .$		$\dots 0 2 \boxed{0} \ 0 \dots$	
5	$+ + [ > \boxed{,} - . < - ] .$	$15 \boxed{0} \ 3$	$\dots 0 2 \boxed{15} \ 0 \dots$	
6	$+ + [ > , \boxed{-} . < - ] .$		$\dots 0 2 \boxed{14} \ 0 \dots$	
7	$+ + [ > , - \boxed{.} < - ] .$			14
8	$+ + [ > , - . \boxed{<} - ] .$		$\dots 0 \boxed{2} \ 14 \ 0 \dots$	
9	$+ + [ > , - . < \boxed{-} ] .$		$\dots 0 \boxed{1} \ 14 \ 0 \dots$	
10	$+ + [ > , - . < - \boxed{]} .$			
11	$+ + [\boxed{>} , - . < - ] .$		$\dots 0 1 \boxed{14} \ 0 \dots$	
12	$+ + [ > \boxed{,} - . < - ] .$	$15 \ 0 \boxed{3}$	$\dots 0 1 \boxed{0} \ 0 \dots$	
13	$+ + [ > , \boxed{-} . < - ] .$		$\dots 0 1 \boxed{255} \ 0 \dots$	
14	$+ + [ > , - \boxed{.} < - ] .$			255
15	$+ + [ > , - . \boxed{<} - ] .$		$\dots 0 \boxed{1} \ 255 \ 0 \dots$	
16	$+ + [ > , - . < \boxed{-} ] .$		$\dots 0 \boxed{0} \ 255 \ 0 \dots$	
17	$+ + [ > , - . < - \boxed{]} .$			
18	$+ + [ > , - . < - ] \boxed{.}$			0

- Primeri 21–25: Program vsebuje tudi znake <. Vhod je prazen.
- Primeri 26–30: Program vsebuje tudi znake , (vejica). Vhod je poljubno dolg.
- Primeri 31–40: Program vsebuje tudi znake [ in ], vendar brez gnezdenja. Vhod je poljubno dolg.
- Primeri 41–50: Program vsebuje vse znake in poljubno stopnjo gnezdenja znakov [ in ]. Vhod je poljubno dolg.

## Izhod

Na izhod izpišite izhod programa **po največ  $10^4$  korakih (izvršenih ukazih)**, saj ni nujno, da se program ustavi. Vsako izhodno število izpišite v svojo vrstico.

## Javni testni primer

Vhod:

```
++[>,-.<-].
15
0
3
```

Izhod:

```
14
255
0
```

## Napotek

Program v brainfucku lahko z enim stavkom preberete v tabelo znakov:

```
Scanner sc = new Scanner(System.in);
...
char[] program = sc.nextLine().toCharArray();
```

## Namig

Kako daleč se lahko na traku premaknemo v (največ)  $10^4$  ukazih programa?

## Oddaja naloge

Svoj program oddajte v obliki ene same datoteke z nazivom `DN05_vvvvvvvv.java`, pri čemer `vvvvvvvv` nadomestite s svojo vpisno številko.