

Vaje pri predmetu Programiranje II

Brainf*ck interpreter

Brainf*uck je kultni univerzalni programski jezik, ki ni namenjen praktični uporabi. Znan je po svoji izjemni varčnosti z možnimi ukazi, ki jih je natanko osem. Leta 1993 ga je ustvaril Urban Müller, da bi imel programski jezik, za katerega bi lahko naredil najmanjši možni prevajalnik. Naloga zahteva, da v C-ju napišete prevajalnik za programski jezik Brainf*ck.

Osnova jezika je polje (array) 30000-ih byte-ov in kazalec (pointer) nanj. Ko program poženemo, kaže kazalec na prvi element v polju, vsi elementi polja pa so 0.

Ukaz	Pomen
+	prišteje ena bajtu, na katerega kaže trenutni kazalec
-	odšteje ena od bajta, na katerega kaže trenutni kazalec
>	trenutni kazalec premakne na naslednjega
<	trenutni kazalec premakne na prejšnjega
.	izpiše vrednost bajta, na katerega kaže trenutni kazalec, na standardni izhod
,	v bajt, na katerega kaže trenutni kazalec, shrani vrednost iz standardnega vhoda (*)
[če je vrednost bajta, na katerega kaže trenutni kazalec, neničelna, nadaljaj z naslednjim ukazom, drugače se premakni na ukaz za pripadajočim]
]	če je vrednost bajta, na katerega kaže trenutni kazalec, neničelna, se vrni na ukaz za pripadajočim [, drugače nadaljaj z naslednjim ukazom

Naloga zahteva, da program izvede celoten brainfu*k program, katerega izvorna koda predstavlja besedilo, ki je podano preko standardnega vhoda. Ukaze predstavljajo le znaki iz zgornje table, vse ostale znake mora interpreter ignorirati (ostali znaki so lahko prisotni in predstavljajo komentarje). Za lažjo predstavo kako se izvaja enostaven program napisan v brainf*cku, si lahko ogledate vizualizacijo na [spletni aplikaciji](#).

Vhodni podatki in izhod

Primer 1: V vhodnih podatkih bo podana izvorna koda programa. Primer vhoda:

```
+++++.>+++++
+++++.>+++++
+++++--.>+++++
+++++.>+++++
+++++
+,-----.>
+++++.
```

Na izhodu pričakujemo izhod brainfu*ck programa, za zgornji primer:

```
Hello!
```

Primer 2: Primer vhoda, ki predstavlja brainfu*ck program z zanko in komentarji:

```
+++++ +++++ initialize counter (cell #0) to 10
[ use loop to set 70/100/30/10
> +++++ ++ add 7 to cell #1
> +++++ +++++ add 10 to cell #2
> +++ add 3 to cell #3
> + add 1 to cell #4
<<<< - decrement counter (cell #0)
]
> ++ . print 'H'
> + . print 'e'
+++++ ++ . print 'l'
. print 'l'
+++ . print 'o'
> ++ . print ' '
<< +++++ +++++ +++++ . print 'W'
> . print 'o'
+++ . print 'r'
----- - . print 'l'
----- - - . print 'd'
> + . print '!'
> . print '\n'
```

Izhod za zgornji program:

```
Hello World!
```

Primer 3: Primer vhoda, ki predstavlja brainfu*ck program z gnezdenimi zankami in komentarji:

```
+>>+++[
  <<[>+<-] zanka
  >[<+>-]>-
]<< zdaj je v tem kazalcu shranjena vrednost 2*2*2 * 2*2*2 ali 64
. izpiše to vrednost; kar v ASCII predstavlja znak @
```

Izhod za zgornji program:

```
@
```

Izvorna koda programa ne bo presegala 60000 znakov.

Ocenjevalni primeri ne bodo vsebovali premikanja kazalca pred prvi ali za zadnji element v polju.

(*) Vhodni ocenjevalni primeri, ne bodo vsebovali ukaza ' , ', ki predstavlja branje iz standardnega vhoda.

Ne pozabite, da mora program vrniti izhodno kodo 0!