

## 2 Regex

Svi smo se susretali sa problemom traženja neke riječi u nekom tekstu. U najboljem slučaju smo znali riječ koju tražimo. Ali obično ne znamo tačnu riječ koju tražimo, nego samo neke informacije o riječi. Na primjer, znamo da je prvo slovo veliko, da se riječ sastoji samo od slova abecede, i da se riječ završava na *-ić*. Da bi iskazali oblik riječi kompjuteru, kojeg ćemo koristiti za pretragu velike količine podataka, napravljen je namjenski jezik za opis neke familije riječi. Taj jezik se naziva *regular expressions*, skraćeno *regex*. Sam regex ima mnogo varijanti. Možda ste se već susreli sa najjednostavnijom: znakom *\** koji mijenja proizvoljan broj karaktera. Tako `ls *.txt` prikazuje listu svih fajlova koji imaju nastavak `.txt`.

Za potrebe našeg zadatka, za dati regex trebate naći prvo poklapanje unutar teksta koji može biti proizvoljne dužine. Koristićete pojednostavljenu varijantu regexa koja ima samo dva operatora:

1. Operator **+** koji mijenja **jedan ili više znakova**. Na primjer, `a+b` se poklapa i sa `axb` i sa `aab` ali ne i sa `ab`.
2. Operator **\*** koji mijenja **nula ili više znakova**. Tako se `a*b` poklapa i sa `axb` i sa `ab`.

### Ulaz i izlaz

Ulazne podatke čitate iz datoteke `regex.in`. U prvom redu se nalazi regex izraz ne duži od 10000 karaktera. Taj izraz ne sadrži nikakve znakove osim slova engleske abecede, brojeva, znaka `_` i operatora `+` i `*`. Iako se na kraju prvog reda mogu nalaziti i drugi znakovi (razmaci i sl.), njih trebate **ignorirati**. Operatori se ne mogu nalaziti na početku i na kraju. Neće biti primjera gdje se dva operatora nalaze na susjednim pozicijama. U sljedećoj liniji se nalazi broj  $0 < n < 2^{31} - 1$ . Nakon toga se u sljedećih  $n$  linija nalazi tekst koji trebate da pretražujete. Tekst može sadržavati bilo koji ASCII znak. Veličina teksta neće prelaziti 1GB memorije.

Kraj linije označava znak `\n`, čija je ASCII vrijednost 10.

Rezultat trebate ispisati u datoteku `regex.out`. Za svaku liniju trebate ispisati da li postoji dio te linije koji odgovara datom regexu (tj. poklapanje). Ako postoji, ispisati poziciju poklapanja unutar linije i dužinu tog poklapanja. Ako postoji više validnih poklapanja, ispisati prvo najkraće poklapanje. Ukoliko poklapanje ne postoji trebate ispisati samo -1. Napomenimo da pozicije brojite od nule.

### Testni primjeris

regex.in	regex.out
<code>a+b*c</code>	1 6
<code>3</code>	-1
<code>ta05xbcz</code>	0 4
<code>qabce \$</code>	
<code>a\$bce</code>	

### Ograničenja

Vaš program može raditi najduže 1 sekundu i ne smije koristiti više od 16 MiB memorije.