

# 1 Binarno Stablo i Rotacije

Dano je binarno stablo od  $n$  čvorova s korijenom 1 i u svakom čvoru se nalazi neki broj  $x_i$ . Rotacija čvora je operacija kojom zamjenimo lijevo i desno dijete tog čvora. Ako ne postoji lijevo dijete, onda samo premjestimo desno dijete na lijevu stranu. Isto vrijedi i ako nema desnog djeteta. Inorder ispis stabla je kada prvo ispišemo lijevo podstablo čvora, zatim ispišemo broj koji se nalazi u trenutnom čvoru i na kraju ispišemo desno podstablo trenutnog čvora. Vaš zadatak je naći stablo sa leksikografski najmanjim inorder obilaskom koje možete dobiti koristeći samo operaciju rotacije navedenu gore.

## 1.1 Ulaz

U prvom retku se nalazi broj  $n$ , ( $1 \leq n \leq 2 \cdot 10^5$ ), broj čvorova u stablu. U sljedećem retku se nalazi  $n$  brojeva  $x_i$ , ( $1 \leq x_i \leq 10^9$ ), broj koji se nalazi u  $i$ -tom čvoru. Zatim u sljedećih  $n - 1$  redaka dana su dva broja,  $u_i$  i  $v_i$ , čvorovi  $u_i$  i  $v_i$  su povezani bridom.

U zadatku će uvijek biti dano valjano binarno stablo s korijenom 1.

## 1.2 Izlaz

U jednom retku ispišite  $n$  brojeva, leksikografski minimalan inorder ispis binarnog stabla koje se može dobiti koristeći samo operaciju rotacije.

U 50

## 1.3 Testni primjer

### 1.3.1 Ulaz primjera

```
5
8 1 4 2 6
1 2
5 1
5 4
3 5
```

### 1.3.2 Izlaz primjera

```
1 8 2 6 4
```

## 1.4 Testni primjer

### 1.4.1 Ulaz primjera

```
10
1 2 3 4 5 6 7 8 9 10
1 2
2 3
3 4
4 5
4 6
6 7
7 8
8 9
3 10
```

#### 1.4.2 Izlaz primjera

```
1 2 5 4 6 7 8 9 3 10
```

### 1.5 Testni primjer

#### 1.5.1 Ulaz primjera

```
7
2 2 2 1 3 1 1
1 2
1 3
2 4
2 5
3 6
3 7
```

#### 1.6 Izlaz primjera

```
1 2 1 2 1 2 3
```

## 2 Stablo i Putevi 2

Dano je binarno stablo od  $n$  vrhova ukorijenjeno u vrhu 1 te je dan broj  $k$ . Vaš zadatak je izračunati broj puteva duljine manjih ili jednakih  $k$  koji prolaze kroz vrh 1. Duljina puta između vrhova  $u$  i  $v$  je broj bridova na tom putu. Puteve od  $u$  do  $v$  i od  $v$  do  $u$  brojimo kao jedan put. Budući da broj puteva može biti jako velik, molimo vas da ispišete traženi broj modulo  $10^9 + 7$ .

## 2.1 Ulaz

U prvom retku su dana dva broja  $n$  i  $k$ , ( $1 \leq k \leq n \leq 10^5$ ).

U sljedećih  $n - 1$  redaka se nalaze parovi  $u_i$  i  $v_i$ , ( $1 \leq u_i, v_i \leq n$ ), što znači da postoji brid između  $u_i$  i  $v_i$  u binarnom stablu.

Uvijek će vam biti dano valjano binarno stablo.

## 2.2 Izlaz

U jedini redak ispišite jedan broj, broj puteva duljine manjih ili jednakih  $k$  koji u sebi imaju vrh 1. Taj broj puteva ispišite modulo  $10^9 + 7$ .

U 20U 50

## 2.3 Testni primjer

### 2.3.1 Ulaz primjera

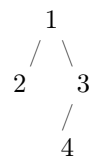
```
4 2
1 2
2 3
1 4
```

### 2.3.2 Izlaz primjera

```
5
```

### 2.3.3 Pojašnjenje

Stablo je oblika



i putevi duljine najviše dva koji prolaze kroz vrh 1 su

```
1
1-3
1-3-4
2-1
2-1-3
```

## 2.4 Testni primjer

### 2.4.1 Ulaz primjera

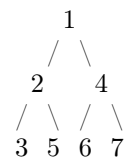
7 4  
1 2  
2 3  
2 5  
1 4  
4 6  
4 7

### 2.4.2 Izlaz primjera

16

### 2.4.3 Pojašnjenje

Stablo je oblika



## 2.5 Testni primjer

### 2.5.1 Ulaz primjera

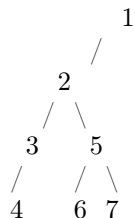
7 3  
1 2  
2 3  
2 5  
5 6  
5 7  
3 4

### 2.5.2 Izlaz primjera

7

### 2.5.3 Pojašnjenje

Stablo je oblika



## 3 Šah na Beskonačnoj Ploči

Nakon što je trgovac Berti našao izgubljene figure i stavio ih na svoju beskonačnu šahovsku ploču, vidio je da figure nisu zadovoljne svojom trenutnom pozicijom. Bijele figure žele biti što dalje od crnih, a crne žele biti što dalje od bijelih. Svaka figura zna matematiku i zna da ako se nalazi na poziciji  $(x_1, y_1)$ , a figura suprotne boje na  $(x_2, y_2)$  onda su one međusobno udaljene  $|x_1 - x_2| + |y_1 - y_2|$  koraka. Kako figura suprotne boje može biti jako puno, one ne znaju koja figura im je najbliža.

Pomozite trgovcu Bertiju da za svaku figuru odredi koliko je udaljena najbliža figura suprotne boje, to jest za bijelu figuru pronađite udaljenost do najbliže crne i za crnu udaljenost do najbliže bijele.

### 3.1 Ulaz

U prvom retku se nalazi broj figura  $n$ , ( $2 \leq n \leq 2 \cdot 10^5$ ). U sljedećem retku se nalazi string  $s$  od  $n$  znakova,  $s[i] \in \{0, 1\}$ , ako je  $s[i] = 0$  onda je  $i$ -ta figura bijela figura, a inače za  $s[i] = 1$   $i$ -ta figura je crna. Zatim u sljedećih  $n$  redaka dano je po dva broja  $x_i$  i  $y_i$ , ( $-10^9 \leq x_i, y_i \leq 10^9$ ) koordinate  $i$ -te figure.

Uvijek će biti barem jedna crna i barem jedna bijela figure tako da je moguće naći udaljenost.

### 3.2 Izlaz

U jednom retku ispišite  $n$  brojeva,  $i$ -ti broj je udaljenost  $i$ -te figure do njoj najbliže figure suprotne boje.

U 20U 50

### 3.3 Testni primjer

#### 3.3.1 Ulaz primjera

```
6
000111
```

-1 0  
-2 -2  
2 2  
1 0  
2 -2  
-2 2

### 3.3.2 Izlaz primjera

2 4 3 2 4 3

## 3.4 Testni primjer

### 3.4.1 Ulaz primjera

3  
011  
0 1  
1 0  
-1 0

### 3.4.2 Izlaz primjera

2 2 2

## 3.5 Testni primjer

### 3.5.1 Ulaz primjera

8  
01010101  
10000000 10000000  
-10000000 -10000000  
1010 88  
0 0  
-33223322 19  
10 10  
12 12  
10101010 10101010

### 3.5.2 Izlaz primjera

179797980 76776697 1078 24 33223341 4 4 20200922

## 4 Rastući Nizovi

Kiki se pretplatio na uslugu *Postani narančast* gdje svaki od sljedećih  $k$  dana dobije jednu kutiju punu naranči. Kada dobije kutiju, Kiki će izabrati točno jednu naranču i pojest će ju, te će baciti ostale u smeće jer je to previše naranči za jedan dan. To znači da će Kiki pojesti  $k$  naranči sveukupno.

Vođen učenjem sv. Heona o *Grind*-u, Kiki je odlučio da svaka naranča koju pojede mora biti strogo veća od prošle. To jest ako je prvi dan pojeo naranču veličine  $a_1$ , drugi dan naranču veličine  $a_2$ , treći dan  $a_3$  itd. do zadnjeg dana  $a_k$ , onda mora vrijediti  $a_1 < a_2 < a_3 < a_4 < \dots < a_k$ . On zna da ne treba postojati jedinstven niz jedenja naranči i zato ga zanima koliko točno takvih nizova postoji kako bi mogao izabrati najbolji. Budući da broj rastućih nizova naranči može biti prevelik, on vas moli da ga ispišete modulo  $10^9 + 7$ .

Kutije mogu imat različiti broj naranči i sve naranče su **različitih** veličina, pa čak i one iz različitih kutija. Kiki mora pojesti točno jednu naranču svaki dan i to tim redom kojim kutije dolaze.

### 4.1 Ulaz

U prvom retku nalazi se broj  $k$ , broj dana/kutija. U sljedećih  $k$  redaka nalazi se prvo broj  $n_i$ , i zatim u istom retku  $n_i$  brojeva  $a_{ij}$ , ( $1 \leq a_{ij} \leq 10^9$ ).

U svim testnim primjerima će biti  $\sum_{i=1}^k n_i \leq 2 \cdot 10^5$ .

### 4.2 Izlaz

Ispišite jedan broj, koliko rastućih nizova možete načiniti koristeći dane kutije.

### 4.3 Testni primjer

#### 4.3.1 Ulaz primjera

```
2
3 1 3 5
4 2 4 6 8
```

#### 4.3.2 Izlaz primjera

```
9
```

### 4.4 Testni primjer

#### 4.4.1 Ulaz primjera

```
4
5 3 2 1 5 100001
2 4 6
1 7
3 1000 100 101
```

#### 4.4.2 Izlaz primjera

```
21
```

### 4.5 Testni Primjer

#### 4.5.1 Ulaz primjera

```
2
1 2
1 1
```

#### 4.5.2 Izlaz primjera

```
0
```

#### 4.5.3 Pojašnjenje

U prvoj kutiji Kiki ima naranču 2, a u drugoj 1 i nikako ne može dobiti rastući niz naranči.

## 5 BST Ubacivanja

Dano je  $n$  brojeva  $x_i$ , i tim redom ih ubacujete u binarno stablo pretraživanja (BST). Vaš zadatak je ispisati koliko različitih nizova je moglo dati ovo binarno stablo pretraživanja koje ste dobili.

Budući da broj nizova može biti jako velik, ispišite ga modulo  $10^9 + 7$ .

### 5.1 Ulaz

U prvom retku nalazi se broj  $n$ , ( $1 \leq n \leq 2 \cdot 10^5$ ). U sljedećem retku nalazi se  $n$  različitih brojeva  $x_i$ , ( $1 \leq x_i \leq 10^9$ ), brojeve koje tim redom morate ubaciti u BST.

### 5.2 Izlaz

Ispišite jedan broj, s koliko različitih nizova možete dobiti to binarno stablo pretraživanja modulo  $10^9 + 7$ . U taj broj uključite i dani niz.

U 50U 80



## 5.3 Testni primjer

### 5.3.1 Ulaz primjera

```
4  
1 2 3 4
```

### 5.3.2 Izlaz primjera

```
1
```

## 5.4 Testni primjer

### 5.4.1 Ulaz primjera

```
8  
5 2 3 1 100 6 8 7
```

### 5.4.2 Izlaz primjera

```
70
```

## 5.5 Testni primjer

### 5.5.1 Ulaz primjera

```
15  
8 4 12 2 6 10 14 1 3 5 7 9 11 13 15
```

### 5.5.2 Izlaz primjera

```
21964800
```

## 6 Build-a-Calculator

Jeste li se kada zapitali kako je napravljen programski jezik? Ako jeste, odlično, jer se u gradu otvorio novi kurs zvan *Build your own Calculator* u kojem u samo dva dana možete naučiti kako isprogramirati svoj vlastiti jednostavni programski jezik. Kao primjer "programskog jezika" za prvi zadatak dobili ste da morate isprogramirati kalkulator koji podržava 4 "operacije": 1. prepoznavanje nenegativnog broja `int`, 2. zagrade (`x`) gdje je `(` znak za lijevu zagradu, `)` znak za desnu zagradu i `x` izraz, 3. zbrajanje dva broja `x+y` gdje je `+` znak a `x` i `y` su izrazi, 4. i množenje dva broja `x*y` gdje je `*` znak a `x` i `y` su izrazi.

Na žalost tutori nisu pazili i zamjenili su redosljed množenja i zbrajanja. Tako ovaj kalkulator mora prvo provesti zbrajanje a tek onda množenje.

Neki primjeri valjanih izraza koje ovaj kalkulator podržava su  $* 1+2*3+4$  koji se redom evaluira  $1+2*3+4 = 3*3+4 = 3*7 = 21$ ,  $* 1+(2*3)+4$  koji se redom evaluira  $1+(2*3)+4 = 1+6+4 = 7+4 = 11$ ,  $*$  i  $((1))*0$  koji se redom evaluira  $((1))*0 = (1)*0 = 1*0 = 0$ .

Postoje i izrazi koje ovaj kalkulator ne podržava a neki od njih su  $* (1+2)*$  jer za  $*$  fali desni broj,  $* ()*2$  jer za  $*$  će faliti lijevi broj,  $* (1+2$  jer zagrade nisu valjane,  $* i (1+2)(3+4)$  jer kalkulator ne može vratiti dva broja (nedostaje operacija između zagrada).

Matematički, ovaj kalkulator možemo zapisati sljedećom gramatikom.

```

<start> ::= <mult_expr>
<mult_expr> ::= <add_expr> '*' <mult_expr> | <add_expr>
<add_expr> ::= <par_expr> '+' <add_expr> | <par_expr>
<par_expr> ::= int | '(' <mult_expr> ')'
```

gdje  $|$  znači ili se dogodila lijeva stvar ili se dogodila desna stvar a int je broj veći ili jednak 0 od najviše 8 znamenaka.

Tutori su zabranili korištenje programa Yacc, pa je vaš zadatak implementirati ovaj kalkulator koji ukoliko je izraz valjan vrati rješenje modulo  $10^9 + 7$  a inače vrati  $-1$ .

## 6.1 Ulaz

U prvom retku dan je broj  $t$ , ( $1 \leq t \leq 100$ ), broj izraza koje trebate evaluirati. Zatim slijedi  $t$  redova, u svakom po jedan string. String se sastoji isključivo od znamenka, zagrade,  $+$  i/ili  $*$ .

Suma duljina izraza neće biti veća od  $2 \cdot 10^5$ . U testnim primjerima neće biti razmaka. Izraze oblika  $00010+2$  evaluirajte u 12.

## 6.2 Izlaz

Ispišite jedan broj, rješenje modulo  $10^9 + 7$  ili  $-1$  ukoliko izraz nije valjan.

U 50

## 6.3 Testni primjer

### 6.3.1 Ulaz primjera

```

3
1*2+3
3+22*13+33
1+2+3+
```

### 6.3.2 Izlaz primjera

5  
1150  
-1

## 6.4 Testni primjer

### 6.4.1 Ulaz primjera

5  
123{\it 321}222+1312\*445+123\*444445+0\*0  
11111111\*11111111  
0+0+00+12+01  
0+3\*2  
42

### 6.4.2 Izlaz primjera

0  
786790129  
13  
6  
42

## 6.5 Testni primjer

### 6.5.1 Ulaz primjera

3  
((232+32)\*(14+30\*3))  
(((3)))\*(0+(32+(43)))  
(32+64+128\*32+64+128

### 6.5.2 Izlaz primjera

34848  
225  
-1

## 7 Latin dot kom

Nedavno na natjecanju iz inženjerskog programiranja Leon i Damjan su predstavili svoj software [Latinary](<http://gramatika.latinary.com/>) (jako je dobar, orz). Latinary je online hrvatsko-latinski rječnik koji sadržava do 20,000 riječi i izraza, te stranica također nudi materijale poput gramatike i izreka.

Trgovcu Bertiju se stranica jako svidjela i zato je odlučio Leonu i Damjanu napisati svoj putopis na latinskom koji je pun izreka koje je sakupio na svojim putovanjima kroz svijet. Taj putopis se sastoji od riječi koje su odvojene razmakom i u njemu nema točke ni zareza. Svaka riječ je sastavljena od najviše 10 malih slova engleske abecede.

Sada Leona i Damjana jako zanima koliko njihovih izreka se nalazi u tom tekstu. Izreka je niz od  $k$  riječi i ona se nalazi u tekstu ako  $\text{text}[i] = \text{izreka}[0]$ ,  $\text{text}[i + 1] = \text{izreka}[1]$ ,  $\dots$ ,  $\text{text}[i + k - 1] = \text{izreka}[k - 1]$  za neki  $i$ .

Ako se neka izreka pojavljuje dva puta u tekstu, nju brojte samo jednom.

Kako ne želimo da zadatak bude pretežak, reći ćemo da se ista riječ neće pojavljivati u dvije različite izreke i točno jednom će biti u istoj izreci. Međutim, može se pojavljivati više puta u tekstu, što ne bi trebalo predstavljati problem.

### 7.1 Ulaz

U prvom retku se nalaze dva broja  $n$  i  $m$ , ( $1 \leq n \leq 10^5$ ) ( $0 \leq m \leq 10^4$ ), broj riječi u putopisu i broj izreka u Latinary. U sljedećem retku se nalazi  $n$  stringova odvojenih razmakom koji su načinjeni isključivo od malih slova engleske abecede. Svaki string će imati najviše 10 slova. U sljedećih  $m$  redaka dan je prvo broj  $k_i$ , ( $1 \leq k_i \leq 10^4$ ), duljina  $i$ -te izreke i zatim u istom retku dano je  $k_i$  stringova odvojenih razmakom. Ti stringovi tvore  $i$ -tu izreku tim redom i svaki string ima najviše 10 slova.

Suma svih  $k_i$  će biti manja ili jednaka  $10^4$ .

### 7.2 Izlaz

U jednom retku ispišite koliko različitih izreka se nalazi u putopisu.

### 7.3 Testni primjer

#### 7.3.1 Ulaz primjera

```
12 3
carpe veni vidi vici alea est iacta diem diem est duobus diebus
3 veni vidi vici
3 alea iacta est
2 carpe diem
```

### 7.3.2 Izlaz primjera

1

## 7.4 Testni primjer

### 7.4.1 Ulaz primjera

8 1

abusus non tollit usum abusus non tollit usum

4 abusus non tollit usum

### 7.4.2 Izlaz primjera

1

## 7.5 Testni primjer

### 7.5.1 Ulaz primjera

25 4

qzsdijsaoei pfnduwgexp juxjdtfmpjr jfrgmqorxr kzxxbezkjf upwyxcrivm peujweqbqz xxa

5 rwegwfbio ffcbozajdz srvjrkwyob ekqctvfjhs qxybmwuqed

5 kpmogeejjf xxapkjsbmf ocwidjwyhw mdhothvnei afvkschznj

5 pfnduwgexp qzsdijsaoei kzxxbezkjf jfrgmqorxr juxjdtfmpjr

5 jrtwgazisx fkvvdvdjdh utuqumavfr hcrsxvqvme neajgiglrx

### 7.5.2 Izlaz primjera

1

## 8 Kreditna

Marko je kao član kluba redovitih letača sakupio mnogo nagradnih milja, te se konačno odlučio počastiti putovanjem u Australiju. Međutim, kako to obično biva, nagradne milje ne pokrivaju sve troškove putovanja, pa ga je gospodin iz avio-kompanije telefonski zamolio da mu Marko e-mailom pošalje **broj kreditne kartice** (niz od 16 znamenki).

Marko je svjestan da je to suludo, ali gospodin s druge strane telefonske linije inzistira na tome. Marko je kao kompromis ponudio da pošalje broj kartice **kriptiran sljedećim algoritmom**: \* Marko će pronaći najmanju znamenku u broju, a ako ih ima više, među njima će odabrat prvu s lijeva. Nazovimo ovu znamenku A. \* Zatim će pronaći najveću znamenku u broju, a ako ih ima više,

među njima će odabrat zadnju. Nazovimo ovu znamenku B. \* Znamenku A će uvećati za 1, osim ako je već jednaka broju 9. \* Znamenku B će umanjiti za 1, osim ako je već jednaka broju 0. \* Konačno, Marko će znamenkama A i B zamijeniti pozicije.

Na primjer, ako je broj njegove kartice 7691 0027 7960 3269, tada će Marko u e-mailu poslati broj: 7691 8027 7960 3261. Napišite program koji će gospodinu iz avio-kompanije pomoći da odredi broj Markove kreditne kartice.

## 8.1 Ulaz

U prvom i jedinom redu nalazi se niz od 16 znamenaka '0' – '9', broj koji je Marko poslao u e-mailu. Napomena: Broj kreditne kartice može početi sa znamenkom '0' i pazite pri ispisu da ne zaboravite '0' na početku ako bude bilo.

## 8.2 Izlaz

Ispišite sve moguće brojeve Markove kreditne kartice, po jedan u svaki red. Ako ne postoji niti jedan valjani broj kreditne kartice, ispišite "banana".

## 8.3 Testni primjer

### 8.3.1 Ulaz primjera

7691802779603261

### 8.3.2 Izlaz primjera

7691002779603269

## 8.4 Testni primjer

### 8.4.1 Ulaz primjera

8381950005575690

### 8.4.2 Izlaz primjera

banana

## 8.5 Testni primjer

### 8.5.1 Ulaz primjera

8446462287549826

### 8.5.2 Izlaz primjera

1446462287549896

8446462217549896

8446462287549196

## 9 Lubenice

Mirko je zadužen da donese lubenice na plažu. Kupio je A komada lubenica od kojih svaka ima masu od 5 kg, B lubenica od 4 kg, C od 3 kg, D od 2 kg i E lubenica od 1 kg. Za nošenje tih lubenica koristit će vrećice čija je maksimalna nosivost 5 kg.

Potrebno je odrediti koliki je **minimalan broj vrećica** koje Mirko treba kupiti da u njima donese sve lubenice.

Mirko će u svakoj vrećici nositi toliko lubenica da ne prekorači nosivost, neće rezati lubenice na manje komade, niti će stavljati vrećice jednu u drugu.

### 9.1 Ulaz

U prvom redu nalazi se pet cijelih brojeva iz teksta zadatka A, B, C, D i E, ( $0 \leq A, B, C, D, E \leq 1000$ ).

### 9.2 Izlaz

U prvi i jedini red potrebno je zapisati koliko vrećica Mirko treba kupiti.

### 9.3 Testni primjer

#### 9.3.1 Ulaz primjera

5 0 0 0 0

#### 9.3.2 Izlaz primjera

5

### 9.4 Testni primjer

#### 9.4.1 Ulaz primjera

0 0 0 0 5

#### 9.4.2 Izlaz primjera

1

### 9.5 Testni primjer

#### 9.5.1 Ulaz primjera

0 10 0 10 3

#### 9.5.2 Izlaz primjera

15

### 9.6 Testni primjer

#### 9.6.1 Ulaz primjera

1 1 5 4 3

#### 9.6.2 Izlaz primjera

7

## 10 Trokut

Nakon duljeg razmišljanja i traženja smisla života, mali Adrijan je odlučio postati umjetnik. Nažalost, nakon dugogodišnjeg uživanja u matematičkim porocima, kao svoje prvo umjetničko djelo naslikao je primjerak **trokuta**. No jako se izenadio kada mu ravnatelj galerije, stanoviti Mirko, nije želio izložiti sliku. Mислеći kako je problem u nedostatku detalja, mali Adrijan sada želi ukrasiti sliku te ponovno isprobati svoju sreću u umjetničkim galerijama.

Adrijan će ukrasiti sliku tako da na početku odabere neki prirodni broj  $N$ , te trokut kojeg je originalno nacrtao proglasi zanimljivim.

Nakon toga Adrijan  $N - 1$  puta ponovi sljedeći postupak: svakom trokutu kojeg je proglasio zanimljivim on spoji polovišta stranica. Na taj će način svaki zanimljivi trokut biti podijeljen na četiri manja trokuta: tri rubna i jedan središnji. Adrijan će zatim zanimljivim proglasiti sve rubne trokute.

Vaš zadatak je pomoći malom Adrijanu tako da napravite simulaciju izgleda njegovog konačnog remek dijela. Za detalje oko formata slike pogledajte izlazne podatke i priložene test primjere.



## 10.1 Ulaz

U prvom i jedinom redu se nalazi prirodni broj  $N$ , ( $1 \leq N \leq 10$ ) iz teksta zadatka.

## 10.2 Izlaz

Potrebno je nacrtati Adrijanovu sliku nakon ukrašavanja. Napomene: \* Jedini znakovi koje smijete koristiti u izlazu su: '/' (slash), '\' (backslash), '\_' (podvlaka), te '.' (točka). \* Izlaz vašeg programa mora točno odgovarati donjim primjerima. Najmanji trokut na izlazu mora sadržavati točno dva znaka na svojim "kosim" stranicama i točno 4 znaka na horizontalnoj stranici (tj. treba biti točno onih dimenzija kao rješenje za  $N = 1$ ). \* Nemojte ostavljati višak razmaka (ili točaka) na kraju bilo kojeg retka. Dakle, zadnji znak u svakom retku mora biti '\'. \* Nemojte ostavljati prazne redove.

## 10.3 Testni primjer

### 10.3.1 Ulaz primjera

1

### 10.3.2 Izlaz primjera

```
.\
/_--\
```

## 10.4 Testni primjer

### 10.4.1 Ulaz primjera

2

### 10.4.2 Izlaz primjera

```
....\
..\./--.\
.\./.../\
/_--\/_--\
```

## 10.5 Testni primjer

### 10.5.1 Ulaz primjera

3

### 10.5.2 Izlaz primjera

