1. Salakood

1 sekund 30 punkti

Vaatleme salakirjasüsteemi, kus teksti kodeerimiseks kasutatakse salasõna ja 26×26 tähest koosnevat tabelit, mille esimeses reas on ladina tähestik ja iga järgmine rida on saadud eelmist ühe tähe võrra vasakule nihutades:

	a	**b**	**c**	...	**z**
a	a	b	c	...	z
b	b	c	d	...	a
⋮	⋮	⋮	⋮		⋮
z	z	a	b	...	y

Teksti kodeerimisel asendatakse kodeeritava teksti i. täht tähega, mis asub salasõna i. tähega märgitud reas avateksti i. tähega märgitud veerus. Kõik kodeeritava teksti mittetähed jäävad endiseks. Kui kodeeritavas tekstis on rohkem tähti kui salasõnas, kasutatakse salasõna viimase tähe järel uuesti esimest, siis teist jne.

Leida algoritm eelkirjeldatud viisil kodeeritud teksti dekodeerimiseks ja kirjutada programm, mis oskab tekste nii kodeerida kui dekodeerida.

Sisend. Tekstifaili `kood.sis` esimesel real on nõutava operatsiooni nimetus `KODEERIDA` või `DEKODEERIDA` ja teisel real 1 kuni 250 väikesest ladina tähest koosnev salasõna. Faili kolmandal real on töödeldava teksti ridade arv N ($1 \leqslant N \leqslant 250$) ja järgmisel N real tekst ise: igal real kuni 250 väikest ladina tähte, kirjavahemärki ja tühikut.

Väljund. Tekstifaili `kood.val` väljastada täpselt N rida: sisendis antud teksti töötlemise tulemus.

Näide.

`kood.sis`
```
KODEERIDA
salasona
2
esimene rida
teine rida
```

`kood.val`
```
wstmwbr radl
twwae jioa
```

Näide.

`kood.sis`
```
DEKODEERIDA
salasona
2
wstmwbr radl
twwae jioa
```

`kood.val`
```
esimene rida
teine rida
```

2. Nelinurga analüüs

1 sekund 30 punkti

Maamõõtja teatas maaomanikule tema krundi nelja piiripunkti koordinaadid, kuid jättis teatamata maatüki kuju. Kuna maatüki müügihind sõltub ka selle kujust, tahab maaomanik seda siiski teada. Kuna korrapärasema kujuga maatükk on kallim, peaks see (eelistuse järjekorras) olema kas ruut, ristkülik, romb, rööpkülik, trapets või muu nelinurk.

Kirjutada programm, mis kontrollib maamõõtja antud punktide koordinaatide põhjal, millise kujundi me saame, kui joonistame pideva joone esimesest punktist teise, teisest kolmandasse, kolmandast neljandasse ja neljandast tagasi esimesse.

Sisend. Tekstifaili `neli.sis` neljal real on igaühel ühe piiripunkti täisarvulised koordinaadid

kujul x_i y_i ($|x_i| \leqslant 10\,000$, $|y_i| \leqslant 10\,000$).

Väljund. Tekstifaili `neli.val` ainsale reale väljastada krundi kuju hinnang: `RUUT`, `RISTKULIK`, `ROMB`, `ROOPKULIK`, `TRAPETS` või `MUU`, või sõna `VIGA`, kui antud punktid antud järjekorras ei moodusta üldse mingit nelinurka.

Näide.

`neli.sis`
```
1 1
1 3
3 3
3 1
```

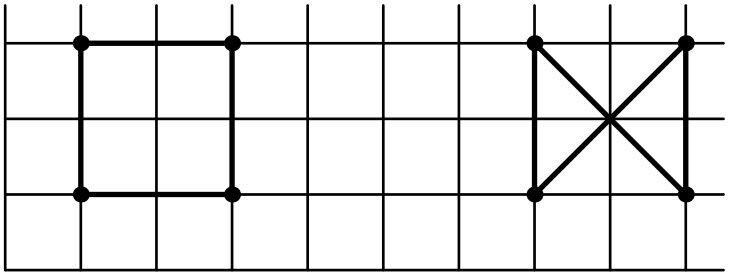
`neli.val`
```
RUUT
```

Näide.

`neli.sis`
```
7 1
7 3
9 1
9 3
```

`neli.val`
```
VIGA
```

Allolev joonis kujutab näidetes toodud "krunte": esimene vasakul ja teine paremal.



3. Teedekaart

2 sekundit — 40 punkti

Ühes riigis on N linna ja nende vahel M maanteed. Iga maantee ühendab omavahel kaht linna ja väljaspool linnu teed omavahel ei lõiku. Kirjutada programm, mis leiab iga linna jaoks talle kõige lähemal ja temast kõige kaugemal asuva linna kauguse, kui eeldada, et kõik liikumised igast linnast igasse teise toimuvad alati lühimat võimalikku teed mööda.

Sisend. Tekstifaili `teed.sis` esimesel real linnade arv N ($2 \leqslant N \leqslant 1000$) ja maanteede arv M ($1 \leqslant M \leqslant 10\,000$). Linnad on nummerdatud $1 \ldots N$. Faili järgmisel M real on igaühel kolm täisarvu: ühe maantee kirjeldus kujul l_1 l_2 d ($1 \leqslant l_1, l_2 \leqslant N$, $l_1 \neq l_2$, $1 \leqslant d \leqslant 1000$), mis tähendab, et see maantee ühendab omavahel linnu l_1 ja l_2 ning selle pikkus on d kilomeetrit. On teada, et igast linnast on võimalik jõuda igasse teise linna.

Väljund. Tekstifaili `teed.val` väljastada täpselt N rida, igale reale kaks tühikuga eraldatud täisarvu. Reale number i väljastada linnale i lähima teise linna kaugus d_i ja temast kaugeima teise linna kaugus D_i.

Näide.

`teed.sis`
```
4 4
1 3 10
1 4 15
2 3 10
3 4 12
```

`teed.val`
```
10 20
10 22
10 12
12 22
```