

**1. RISTKÜLIKUD**

5 punkti

10 sekundit

Kirjutada programm, mis leiab tasandil antud punkti ja ristkülikute koordinaatide järgi nende ristkülikute arvu, mille sisse antud punkt jääb. On teada, et ristkülikute servad on koordinaattelgedega paralleelsed.

Sisend. Tekstifaili `RISTK.SIS` esimesel real on antud punkti koordinaadid  $x_p$   $y_p$ . Faili teisel real on ristkülikute arv  $N$  ( $1 \leq N \leq 100$ ) ja järgmisel  $N$  real igalühel ühe ristküliku vasaku alumise ja parema ülemise nurga koordinaadid  $x_{i1}$   $y_{i1}$   $x_{i2}$   $y_{i2}$ . Kõik koordinaadid on täisarvud, mille absoluutväärtus ei ületa 10000. Ühel real olevad arvud on üksteisest eraldatud tühikutega.

Väljund. Tekstifaili `RISTK.VAL` ainsale reale väljastada nende ristkülikute arv, mille sisse faili esimesel real näidatud punkt jääb. Ristküliku serval või nurgal asuv punkt lugeda ristküliku sees olevaks.

<u>Näide.</u>	<code>RISTK.SIS</code>	<code>RISTK.VAL</code>
	1 1	2
	3	
	-1 0 4 3	
	1 1 6 7	
	-2 2 7 4	

**2. BITIJADAD**

10 punkti

10 sekundit

Kirjutada programm, mis leiab kõik bitijadad, mida saab moodustada, kasutades etteantud arvu nulle ja ühtesid.

Sisend. Tekstifaili `JADAD.SIS` ainsal real on tühikuga eraldatud täisarvud  $M$  ja  $N$  ( $1 \leq M+N \leq 15$ ).  $M$  on nullide ja  $N$  ühtede arv otsitavates bitijadades.

Väljund. Tekstifaili `JADAD.VAL` väljastada kõik sellised bitijadad, milles on täpselt  $M$  nulli ja täpselt  $N$  ühte. Iga jada väljastada eraldi reale ja numbrid ühel real väljastada vahetult üksteise kõrvale. Jadade järjekord failis pole oluline, kuid iga jada peab esinema täpselt üks kord.

<u>Näide.</u>	<code>JADAD.SIS</code>	<code>JADAD.VAL</code>
	2 1	100
		010
		001

**3. SXML**

20 punkti

10 sekundit

SXML (ingl *Simplified XML*) on tekstimärgenduskeele XML lihtsustatud versioon. SXML-sõnum koosneb sõnadest, mille vahel on eraldajad ja märgendid. SXML-sõnad on suurtest ja väikestest ladina tähtedest koosnevad mittetühjad sõned. SXML-eraldajad on tühik, tabulaator (ASCII kood 9) ja reavahetus. SXML-fraas määratakse järgmiste reeglitega:

- tühi sõne on SXML-fraas;
- kui  $F$  on SXML-fraas ja  $G$  on SXML-sõna või SXML-eraldaja, siis  $FG$  on SXML-fraas;
- kui  $F$  on SXML-fraas ja  $G$  on SXML-sõna, siis  $\langle G \rangle F \langle /G \rangle$  on SXML-fraas;  $\langle G \rangle$  nimetatakse selle fraasi algus- ja  $\langle /G \rangle$  lõpumärgendiks ning sõna  $G$  nende märgendite nimeks;

SXML-sõnum määratakse järgmise reegluga:

- kui  $F$  on SXML-fraas ning  $G_1$  ja  $G_2$  on (tühjad või mittetühjad) SXML-eraldajate jasad, siis  $G_1 \langle SXML \rangle F \langle /SXML \rangle G_2$  on SXML-sõnum.

Kirjutada programm, mis kontrollib, kas sisendfailis olevad sümbolid moodustavad korrektse SXML-sõnumi.

**Sisend.** Tekstifailis `SXML.SIS` on kuni 100000 sümbolit kuni 100 sümboli pikkuste ridadena. On teada, et ühegi märgendi nimi ei ole pikem kui 50 sümbolit ja märgendid ei sisaldu üksteises rohkem kui 100 taseme sügavuselt.

**Väljund.** Kui sisendfailis on korrektne `SXML`-sõnum, väljastada tekstifaili `SXML.VAL` ainsale reale sõna `JAH`. Kui sisendfailis ei ole korrektne `SXML`-sõnum, väljastada tekstifaili `SXML.VAL` esimesele reale sõna `EI` ja teisele reale kaks tühikuga eraldatud täisarvu: esimese sellise positsiooni rea- ja veerunumber, et faili algusest kuni sellele positsioonile eelnevani ulatuv tekstiosa on mingi korrektse `SXML`-sõnumi algus, aga faili algusest kuni selle positsioonini ulatuv tekstiosa enam ei ole. Rea- ja veerunumbrid algavad ühest. Kui sisendfailis on lubamatu reavahetus, siis väljastada reanumbriks selle rea number, mille lõpus lubamatu reavahetus on, ja veerunumbriks 0. Kui sisend lõpeb liiga vara (st kui faili lõpp on lubamatus kohas), väljastada rea- ja veerunumbriteks 0 0.

**Näide 1.**

<code>SXML.SIS</code>	<code>SXML.VAL</code>
<code>&lt;SXML&gt;</code>	<code>JAH</code>
<code>  See on SXML test</code>	
<code>&lt;aaa&gt;testtesttest&lt;/aaa&gt;</code>	
<code>&lt;/SXML&gt;</code>	

**Näide 2.**

<code>SXML.SIS</code>	<code>SXML.VAL</code>
<code>&lt;SXML&gt;AAA&lt;AAA&gt;xxx</code>	<code>EI</code>
<code>yyy zzz&lt;bb&gt;misiganes</code>	<code>3 12</code>
<code>&lt;/bb&gt;  &lt;/AaA&gt;&lt;/SXML&gt;</code>	

**Näide 3.**

<code>SXML.SIS</code>	<code>SXML.VAL</code>
<code>&lt;SXML&gt;Poolik fail&lt;/SX</code>	<code>EI</code>
	<code>0 0</code>

#### 4. FAILI VALIMINE

30 punkti

10 sekundit

Faili avamise dialoogis näidatakse kasutajale kõigi aktiivses kataloogis olevate failide nimesid tähestikulises järjekorras. Vajaliku faili väljavalimiseks võib kasutada tähe- ja nooleklahve.

Dialoogi avanemisel on valitud nimekirja esimene fail. Täheklahvide vajutamisel valitakse nimekirjast esimene fail, mille nime algus on vastav tähejada. Kui vajutada sellisele täheklahvile, et moodustuv tähejada pole ühegi failinime algus, siis seda vajutust ignoreeritakse. Klahvi `<UP>` vajutamisel valitakse hetkel aktiivsele failile nimekirjas eelnev fail (kui valitud on esimene fail, liigub `<UP>` viimasele failile). Klahvi `<DOWN>` vajutamisel valitakse hetkel aktiivsele failile nimekirjas järgnev fail (kui valitud on viimane fail, liigub `<DOWN>` esimesele failile). Nooleklahvi vajutamisel “unustab” dialoog ära juba vajutatud täheklahvide jada ja kui nooleklahvide järel vajutatakse uuesti täheklahve, valitakse uueks aktiivseks failiks esimene, mille nime algus on uus tähejada.

Näiteks, kui kataloogis on failid nimedega `aaa`, `aba`, `abb`, `abc`, `cab` ja `cbb`, siis klahvide `<a>` ja `<b>` vajutamise järel on aktiivne fail `aba`. Kui seejärel vajutada klahvi `<DOWN>`, saab aktiivseks fail `abb` ja kui seejärel vajutada klahvi `<c>`, saab aktiivseks fail `cab` (mitte `abc`).

Kirjutada programm, mis leiab minimaalse võimaliku klahvivajutuste arvuga viisi kasutajale vajaliku faili valimiseks sellises dialoogis.

**Sisend.** Tekstifaili `FAILID.SIS` esimesel real on failide arv  $N$  ( $1 \leq N \leq 1000$ ). Järgmisel  $N$  real on failide nimed. Failide nimed koosnevad kuni 20 väikesest ladina tähest ja on esitatud tähestiku järjekorras. Viimasel real on otsitava faili nimi. See nimi on nimekirjas olemas.

**Väljund.** Tekstifaili `FAILID.VAL` väljastada vajalike klahvivajutuste loetelu, iga klahvivajutus eraldi reale. Kui minimaalse pikkusega jadasid on mitu, väljastada ükskõik milline neist.

<u>Näide.</u>	FAILID.SIS	FAILID.VAL
	8	<a>
	aaaa	<b>
	aaab	<UP>
	aaac	
	aaad	
	aaae	
	abcd	
	abdd	
	bcde	
	aaae	

**5. LIINIBUSSID**

35 punkti

10 sekundit

Linnas on  $N$  bussipeatust, mis on ühendatud ringliinideks nii, et iga peatus on täpselt ühel liinil. Buss sõidab ühest peatusest järgmisse täpselt 1 minuti. Ühel hetkel (kui kõik bussid on peatustes) fikseerib dispetšer kõigi busside asukohad. Kirjutada programm, mis arvutab välja, mitme minuti pärast on kõik bussid jälle samades peatustes.

Sisend. Tekstifaili BUSSID.SIS esimesel real on peatuste arv  $N$  ( $1 \leq N \leq 1000$ ) ja järgmisel  $N$  real igaühel kaks tühikuga eraldatud peatusenime. Rida kujul `aaa bbb` tähendab, et peatus `bbb` on mingil liinil `aaa` järel järgmine peatus. Peatuste nimed koosnevad kuni 20 väikesest ladina tähest või numbrist ja iga peatuse nimi esineb täpselt ühel real esimesena ja täpselt ühel real teisena.

Väljund. Tekstifaili BUSSID.VAL ainsale reale väljastada vähim positiivne minutite arv, mille järel kõigi busside asukohad selle linna bussiliinidel korduvad. On teada, et see arv ei ületa 2147483647.

<u>Näide.</u>	BUSSID.SIS	BUSSID.VAL
	5	6
	aaa eee	
	ddd bbb	
	ccc aaa	
	bbb ddd	
	eee ccc	