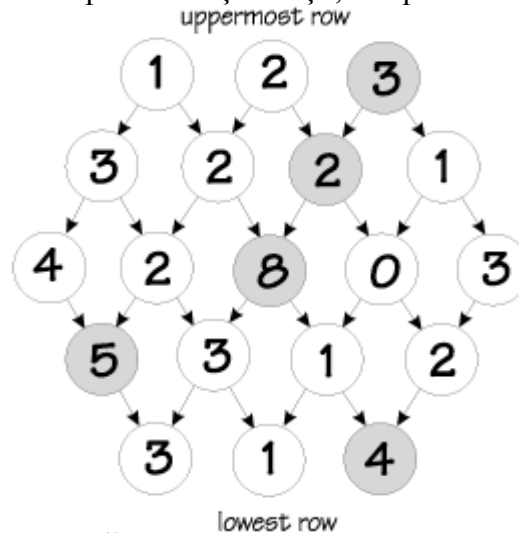


Skaitļu šūnas (Somija)

Pirmajā zīmējumā attēlots skaitļu šūnu fragments, kura šūnās (zīmējumā aplīšos) ir ierakstīti skaitļi (malas garums attēlotajam šūnu fragmentam ir 3). Maršruts sākas no jebkuras šūnas augšējā līnijā un beidzas ar kādu šūnu apakšējā līnijā. No katras šūnas maršruts var turpināties tikai vai nu diagonāli uz leju pa labi vai arī diagonāli uz leju pa kreisi. Veidojot maršrutu caur skaitļu šūnām, jums ir atļauts *ne vairāk kā vienu reizi* izpildīt divu skaitļu maiņu *ne vairāk kā vienā* skaitļu šūnu horizontālajā rindā. (Maiņa būtībā nozīmē, ka vienā izvēlētajā rindā jums ir atļauts pārvietot lielāko šīs rindas skaitli uz jebkuru vietu tajā pašā rindā). Jūsu uzdevums ir uzrakstīt programmu, kas skaitļu šūnu fragmentam atrod lielāko starp visu iespējamo maršrutu skaitļu summām, izmantojot iespēju vienu reizi izpildīt skaitļu maiņu, kā aprakstīts iepriekš.



1.zīmējums. Šūnu fragments ar malas garumu 3.

Ierobežojumi:

- Skaitļi šūnās ir veseli skaitļi no 0 līdz 99;
- Šūnu fragmenta malas garums ir naturāls skaitlis robežās no 1 līdz 99.

Ievaddati: Skaitļu šūnu fragmenta malas garums ir ievaddatu faila HON.IN pirmajā rindā. Ja fragmenta malas garums ir n , fragments sastāv no $2n-1$ skaitļu šūnu rindas. Nākamajās $2n-1$ ievaddatu faila rindās ir doti katrai šūnu rindai atbilstošie skaitļi, kā tas attēlots zemāk:

```
3
1 2 3
3 2 2 1
4 2 8 0 3
5 3 1 2
3 1 4
```

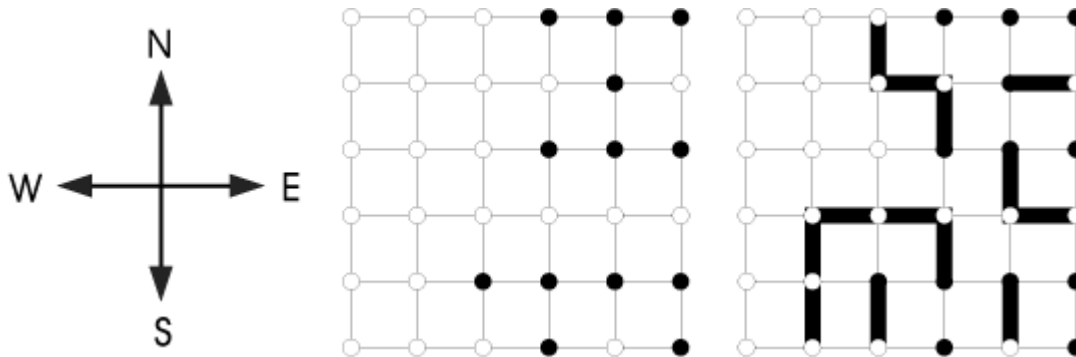
Izvaddati: Vislielākā summa jāieraksta kā vesels skaitlis failā HON.OUT 1.zīmējumā attēlotajam piemēram tas ir:

```
22
```

1.zīmējumā maršruts ar lielāko summu ($3+2+8+5+4=22$) ir ieēnots. Atzīmēsim, ka skaitlis '5' fragmenta ceturtajā rindā no augšas ir jāsamaina uz šīs rindas trešo šūnu (skaitot no kreisās puses).

Elektroniskā plate (Lietuva)

Uz kvadrātiskas plates virsmas ir iegravēts kvadrātisks režģis. Punktus, kuros krustojas divas režģa līnijas, saucim par mezglpunktiem. Režģī ir $n \times n$ mezglpunktu.



2.attēls. Uzdevuma (centrā) un atrisinājuma (pa labi) piemērs.

Daži mezglpunkti satur kontaktus. Jūsu uzdevums ir lietojot elektriskās ķēdes savienot šos kontaktus ar mezglpunktiem, kas atrodas uz plates robežas. Ķēde var tikt izvietota tikai pa režģa līnijām (piemēram, tā nevar tikt izvietota slīpi). Divām dažādām ķēdēm nevar būt kopīgi punkti, tādējādi tās nevar iet caur kopīgu režģa līniju vai kopīgu mezglpunktu. Ķēde nevar iet arī pa režģa līniju, kas atrodas uz plates robežas (t.i. līdzko ķēde sasniedz plates robežu, tā ir jāpārtrauc), kā arī tā nevar iet caur mezglpunktu, kas satur citu kontaktu.

2. attēla centrā ir dots elektroniskās plates, kas satur kontaktus, piemērs. Kontakti ir attēloti ar melniem punktiem.

Uzdevums. Uzrakstiet programmu, kas savieno maksimālo iespējamo kontaktu skaitu ar mezglpunktiem uz plates robežas. Kontakti, kas jau atrodas uz plates robežas, apmierina prasības, un elektriskā ķēde šiem kontaktiem vispār nav nepieciešama.

Ja eksistē vairāki risinājumi, atrodiet jebkuru no tiem.

Ievaddati: Ievaddati ir doti teksta failā "ELE.IN". Pirmā faila rinda satur veselu skaitli n ($3 \leq n \leq 15$).

Tālāk seko n rindas, kur katrā rindā ir doti n cipari, kas atdalīti ar vienu tukšumsimbolu. Šie cipari var būt 1 vai 0. Viens (1) nozīmē, ka atbilstošais mezglpunkts satur kontaktu, bet nulle (0) nozīmē, ka atbilstošais mezglpunkts kontaktu nesatur.

Mezglpunkti ir sanumurēti no 1 līdz $n \times n$ pa rindām no kreisās uz labo pusi sākot ar augšējo rindu un uz leju. Par kontakta identifikātoru kalpo atbilstošā mezglpunkta numurs.

Izvaddati: Rezultāts jāizvada teksta failā "ELE.OUT". Pirmajai faila rindai jāsaturs vesels skaitlis k – lielākais iespējamais kontaktu skaits, kurus var savienot ar mezglpunktiem uz plates robežas. Katrā no nākošajām k rindām jābūt vienas elektriskās ķēdes, kas savieno atbilstošo kontaktu ar mezglpunktu uz plates robežas, aprakstam. Vispirms jānodod atbilstošā kontakta identifikators, pēc tam burtu virkne, kas apraksta ķēdes virzienus: E – uz austrumiem, W – uz rietumiem, N – uz ziemeļiem, S – uz dienvidiem. Starp identifikatoru un burtu virkni jābūt vienam tukšumsimbolam, bet starp virknes burtiem tukšumsimboli nedrīkst būt.

Elektriskās ķēdes jāizvada atbilstošo kontaktu identifikatoru augšanas secībā.

Piemērs:

Ievaddati (fails ELE.IN)

```
6
0 0 0 1 1 1
0 0 0 0 1 0
0 0 0 1 1 1
0 0 0 0 0 0
0 0 1 1 1 1
0 0 0 1 0 1
```

Izvaddati (fails ELE.OUT)

```
6
11 E
16 NWN
17 SE
27 S
28 NWWSS
29 S
```

"Laika zonas" (Zviedrija)

Jūs esat darījumu cilvēks, kurš sadarbojas ar cilvēkiem no visas pasaules. Vienas dienas laika jūs no katras laika zonas saņemat tieši vienu ziņojumu, tai skaitā arī no tās laika zonas, kurā jūs atrodaties. Par katru ziņojumu ir zināms arī tā sūtīšanas lokālais laiks (ziņojums tiek piegādāts tai pašā brīdī). Diemžēl tūkstošgades kļūdas dēļ šie lokālie laiki ir doti bez jebkādam norādēm, no kuras laika zonas atbilstošais ziņojums ir sūtīts.

Uzdevums: par katru ziņojumu noskaidrot, no kuras laika zonas tas ir sūtīts.

Šajā uzdevumā stundu skaits dienā n var būt no 5 līdz 60. Laika zonu skaits vienmēr sakrīt ar stundu skaitu dienā un katrā laika zonā laika nobīde ir vesels skaits stundu.

Laika zonas ir sanumurētas no 0 līdz $n-1$. Jūs dzīvojat un saņemat ziņojumus GMT, kas ir 0-tā laika zona bez laika nobīdes. Laika zonas tiek skaitītas no jusējās uz rietumiem, tas nozīmē, jums jāpieskaita pie z -zonas lokāla laika z stundas, lai iegūtu jūsu zonas laiku. Atzīmēsim, ka tas nav parastais zonu skaitīšanas veids, parasti zonu 2 apzīmētu kā "-2:00".

Piemēram, ja otrās laika zonas lokālais laiks ir 03:15, tad jūsu laika zonā (GMT) laiks ir 05:15.

Divi dažādi ziņojumi nevar pienākt vienā un tajā pašā minūtē, un visi ziņojumi pienāk dienas laikā (tas ir no 0:00 līdz $(n-1):59$), bet nekad divi ziņojumi nepienāk vienlaicīgi. Datuma līnija atrodas starp 0-to un pēdējo zonu un tādejādi var tikt ignorēta.

Ievaddati: Teksta faila "ZON.IN" pirmā rinda satur laika zonu skaitu n ($5 \leq n \leq 60$), kas ir arī stundu skaits dienā un ziņojumu skaits, ko jūs vienā dienā saņemat.

Sekojošajās n rindās doti lokālie ziņojumu izsūtīšanas laiki formā $hhmm$ (2 cipari stundām un 2 cipari minūtēm, kur $0 \leq hh \leq n-1$ un $0 \leq mm \leq 59$). Rindas ir sakārtotas hronoloģiskā secībā, t.i. ziņojums, kas pienāca pirmais, ir pirmajā rindā, otrais ziņojums otrajā, utt.

Visos testa piemēros būs viens vienīgs atrisinājums.

Izvaddati: Teksta failam "ZON.OUT" jāsatur tieši viena rinda, kurā jābūt n skaitļiem no 0 līdz $n-1$, kuri norāda laika zonas, no kurām n ziņojumi tika sūtīti. Pirmais skaitlis atbilst pirmajam laikam, kas dots ievaddatos, otrais skaitlis otrajam, utt.

Piemērs:

ZON.IN

5

0017

0250

0400

0201

0002

ZON.OUT

3 1 0 2 4

Ievērojiet, ka trešais ziņojums noteikti ir no 0-tās zonas, jo pretējā gadījumā tas būtu pienācis vēlāk par 4:59, kas ir pēdējā minūte piecu stundu dienā.