

**LATVIJAS REPUBLIKAS 19. INFORMĀTIKAS
OLIMPIĀDES II POSMA UZDEVUMI
VECĀKAJAI (10.-12. klašu) GRUPAI**



1. "LAIMĪGIE SKAITĻI"

(50 punkti)*

Par *laimīgu skaitli* sauksim tādu skaitli, kuru var sadalīt divos nepārtrauktos fragmentos tā, lai ciparu summa abos būtu vienāda. Piemēram, skaitli 17282 var sadalīt fragmentos 172 un 82, skaitli 1213141516171819 fragmentos 12131415161 un 71819, bet skaitli 2172823 šādi sadalīt nav iespējams.

Ja skaitlis nav laimīgs, tad ļausim no tā sākuma un/vai beigām dažus ciparus pēc kārtas nodzēst cerībā, ka atlikušais skaitlis izrādīsies laimīgs. Tā no skaitļa 2172823 var iegūt laimīgus skaitļus 17282 un 21728.

Uzrakstiet programmu, kas nosaka, vai ievadītais skaitlis ir laimīgs, un, ja nav, kādu lielāko laimīgo skaitli no tā var iegūt, nodzēšot ciparus no tā sākuma un/vai beigām!

Ievaddati

Teksta faila `laimigie.dat` vienīgajā rindā dots naturāls skaitlis, kas sastāv no nenulles cipariem, un ciparu skaits tajā nepārsniedz 10000.

Izvaddati

Teksta faila `laimigie.rez` vienīgajā rindā jāizvada naturāls skaitlis - lielākais laimīgais skaitlis, kādu iespējams iegūt no dotā, ja nepieciešams, nodzēšot dažus ciparus no tā sākuma un/vai beigām.

Ja ar iepriekšminētajām darbībām laimīgu skaitli iegūt nav iespējams, failā jāizvada viens vārds "NEVAR".

Piemēri

| | |
|--|--|
| Ievaddati (fails <code>laimigie.dat</code>) | Izvaddati (fails <code>laimigie.rez</code>) |
| 17282 | 17282 |
| Ievaddati (fails <code>laimigie.dat</code>) | Izvaddati (fails <code>laimigie.rez</code>) |
| 63 | NEVAR |
| Ievaddati (fails <code>laimigie.dat</code>) | Izvaddati (fails <code>laimigie.rez</code>) |
| 2172823 | 21728 |
| Ievaddati (fails <code>laimigie.dat</code>) | Izvaddati (fails <code>laimigie.rez</code>) |
| 118 | 11 |

*) Katra testa izpildei tiks dota viena vai divas (tikai uzdevuma "Laimīgie skaitļi" diviem pēdējiem testiem) sekundes. Katram uzdevumam būs 10 testi. Par pilnīgi pareizu laikā izpildītu testu tiks piešķirti 5 punkti.

2. "RINDA PĒC DESĀM"

(50 punkti)*

Pie lielisko desu stenda vēl pirms tā atvēršanas ir izveidojusies gara rinda. Šajā rindā stāv $2n+1$ pircējs (n - naturāls skaitlis), un katram pircējam ir noteikts (katram atšķirīgs) ietekmes līmenis desu ražošanas biznesā (naturāls skaitlis - zemāko ietekmi apzīmē ar 1, pašu lielāko - ar $2n+1$).

Kad desu stends uzsāk darbu, pirmais rindā stāvošais pircējs iegādājas desas un atstāj rindu. Tikko kā šis pircējs ir apkalpots, desu pārdevējs aiz cieņas pret ietekmīgajām personām desu ražošanas biznesā, apkalpo bez rindas pircēju ar vislielāko ietekmi.

Pēc šo divu pircēju apkalpošanas rindā ir palicis $2n-1$ pircējs, no kuriem atkal pirmais rindā esošais pircējs iegādājas desas, un pēc tā tiek apkalpots pircējs ar vislielāko ietekmi.

Tā šis process turpinās, pārmaiņus apkalpojot pirmo un tad ietekmīgāko (no palikušajiem) pircēju, līdz rindā palicis tikai viens pircējs.

Jūsu uzdevums – uzrakstīt programmu, kas dotai pircēju secībai nosaka, kurš no tiem rindā paliks pēdējais.

Ievaddati

Ievaddatu faila `desas.dat` pirmajā rindiņā ir dots naturāls skaitlis n ($1 \leq n \leq 10^5$), un pēc tam nākamajās $2n+1$ rindās attiecīgais rindas sastāvs, sākot ar tuvāko desu stendam. Faila $i+1$ -ajā rindā ($1 \leq i \leq 2n+1$) dots i -tā rindā esošā pircēja ietekmes līmenis.

Izvaddati

Izvaddatu faila `desas.rez` vienīgajā rindiņā ir jāizvada tā pircēja ietekmes līmenis, kurš rindā paliks pēdējais.

Piemēri

| Ievaddati (fails <code>desas.dat</code>) | Izvaddati (fails <code>desas.rez</code>) |
|---|---|
| 2 | 3 |
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| 5 | |

| Ievaddati (fails <code>desas.dat</code>) | Izvaddati (fails <code>desas.rez</code>) |
|---|---|
| 2 | 2 |
| 4 | |
| 5 | |
| 1 | |
| 2 | |
| 3 | |

*) Katra testa izpildei tiks dota viena vai divas (tikai uzdevuma "Laimīgie skaitļi" diviem pēdējiem testiem) sekundes. Katram uzdevumam būs 10 testi. Par pilnīgi pareizu laikā izpildītu testu tiks piešķirti 5 punkti.

3. "LIELĀKĀ FIGŪRA"

(50 punkti)*

Taisnstūrveida tabulā ir N rindas un M kolonnas. Par *blakus rūtiņām* tabulā sauksim tikai tādas rūtiņas, kurām ir kopīga vertikāla vai horizontāla mala.

Katra rūtiņa ir vai nu tukša, vai arī tajā ir ierakstīta zvaigznīte.

Piemēram,

| | | | | | | | | |
|---|---|---|---|---|---|---|---|--|
| * | * | | | | | | | |
| * | | | * | * | * | * | | |
| | | * | * | * | * | * | * | |
| | | * | * | * | | * | * | |
| | | | * | * | * | * | | |
| | * | * | | | | | | |

Katra rūtiņa, kurā ir ierakstīta zvaigznīte, pieder kādai *figūrai*. Ja divas zvaigznītes atrodas blakus rūtiņās, tad tās pieder vienai figūrai. Tukšās rūtiņas figūrām nepieder. Par figūras *laukumu* sauksim figūrai piederošo rūtiņu skaitu. Augstāk dotajā piemērā ir trīs figūras.

Par figūras *perimetru* sauksim to figūrai piederošo rūtiņu skaitu, kuras vai nu atrodas tabulas malas (pirmā vai pēdējā rindā vai kolonnā) rūtiņās, vai arī kāda tās blakus rūtiņa ir tukša.

Uzrakstiet programmu, kas dotai tabulai atrod lielākās figūras laukumu un perimetru!

Ievaddati

Teksta faila *figura.dat* pirmajā rindā doti divi naturāli skaitļi $N(N \leq 250)$ un $M(M \leq 250)$, kas atdalīti ar tukšumsimbolu. Katrā no nākošajām N faila rindām dots pa M simboliem katrā. Simbols faila $i+1$ -ās rindas j -tajā kolonnā atbilst simbolam tabulas i -tās rindas j -tajā kolonnā. Ja failā ir simbols '*', tad tas nozīmē, ka atbilstošajā tabulas rūtiņā ir ierakstīta zvaigznīte, bet, ja failā ir simbols '.' (punkts), tad tas nozīmē, ka atbilstošā tabulas rūtiņa ir tukša.

Izvaddati

Teksta failā *figura rez* vienīgajā rindā jāizvada divi veseli nenegatīvi skaitļi, kas atdalīti ar tukšumsimbolu – lielākās figūras laukums un perimetrs. Ja ir vairākas figūras ar lielāko laukumu, tad jāizvada informācija par to, kurai perimetrs ir vislielākais.

Piemēri

| Ievaddati (fails <i>figura.dat</i>) | Izvaddati (fails <i>figura rez</i>) |
|---|--------------------------------------|
| 6 9 **..... * . **** . .. ***** .. *** . ** ... ***** **..... | 19 15 |
| 5 16 ***** . ***** . ***** ** . ** * . . *** * . . * . *** . * . . *** ***** . * . . * . . *** ***** . ***** | 17 17 |
| 1 1 . | 0 0 |

*) Katra testa izpildei tiks dota viena vai divas (tikai uzdevuma "Laimīgie skaitļi" diviem pēdējiem testiem) sekundes. Katram uzdevumam būs 10 testi. Par pilnīgi pareizu laikā izpildītu testu tiks piešķirti 5 punkti.