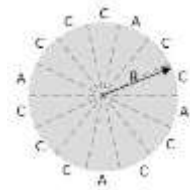


U1. Vadovų susitikimas. Planuojamas šalių vadovų susitikimas. Jie turi susėsti prie apvalaus R spindulio stalo taip, kad kiekvienam sėdinčiam tektų bent X kv. vnt. stalo ploto. A grupės šalių vadovai negali sėdėti greta vienas kito, tarpuose sėdinčių kitų šalių atstovų skaičius gali skirtis 1. Parašykite programą, kuri nustatytų, ar vadovai galės susėsti pagal reikalavimus bei pateiktą susėdimo tvarką (A šalių atstovai žymimi raide A, kitų šalių – raide C). Tvarka pradeda nuo A šalių atstovų, kuriuos skiria daugiausia kitų šalių dalyvių.



Duomenys. Tekstinio failo 'U1.txt' eilutėje nurodytas stalo spindulys R, minimalus viename sėdinčiam skirtas plotas X kv. vnt., dalyvių iš A šalių skaičius m ($m > 1$), bendras dalyvių skaičius n ($n > m$).

Rezultatai. Į ekraną atspausdinkite „GALIMA“, viename dalyviui tenkantį plotą ir susėdimo tvarką, jeigu dalyviai gali susėsti; „NEPAKANKA VIETOS“, viename dalyviui tenkantį plotą ir susėdimo tvarką, jeigu galima sudaryti tinkamą sėdėjimo tvarką, tačiau nepakanka vietos; „NEGALIMA“ kitais atvejais.

U1. txt	Ekranas
3 2 4 14	GALIMA 2.02 ACCCACCACCACC
2 2 4 14	NEPAKANKA VIETOS 0.90 ACCCACCACCACC
3 2 8 14	NEGALIMA

U2. Krepšinio rungtynės

Krepšinio turnyro organizatoriams informacija apie dviejų krepšinio komandų A ir B tarpusavio rungtynes buvo pateikta kaip atskirų žaidėjų metimų ir pelnytų taškų stenograma. Parašykite programą, kuri iš pateiktų duomenų nustatytų, kuri komanda laimėjo ir išrinktų rezultatyviausią laimėjusios komandos žaidėją (surinkusį daugiausiai taškų). Esant keliems žaidėjams su vienodu surinktų taškų skaičiumi – parinkite žaidėją, kurio numeris mažesnis.

Duomenys. Tekstinio failo 'U2.txt' pirmoje eilutėje pateikta, kiek A ir B komandose yra žaidėjų Z ($2 \leq Z \leq 12$). Žemiau kiekvienoje eilutėje yra informacija apie žaidėjus (pirma išvardinti komandos A žaidėjai, paskui komandos B), kur pateiktas žaidėjo numeris NR ($1 \leq NR \leq 12$), metimų skaičius N ($2 \leq N \leq 100$) ir toliau išvardinti atskirais metimais žaidėjo pelnyti taškai T ($0 \leq T \leq 3$). 0 pelnytų taškų reiškia, kad žaidėjas metimo nepataikė.

Rezultatai. Į ekraną pirmoje eilutėje atspausdinkite, kuri komanda laimėjo, o antroje eilutėje – laimėjusios komandos rezultatyviausio žaidėjo numerį, pelnytų taškų sumą, taiklių ir netaiklių metimų skaičius.

U2. txt	Ekranas
2 2 1 5 2 2 3 0 2 2 6 1 2 3 0 0 0 1 6 0 1 1 0 2 3 2 7 1 3 0 2 1 3 2	Laimėjo B komanda. Rezultatyviausio B komandos žaidėjo nr. 2. (Pelnyta taškų: 12; taiklių metimų sk.: 6; netaiklių metimų sk.: 1)

U3. Sniego valymas. Sniegas užklojo miesto gatves. Jas reikia valyti. Mieste yra N mazgų ir M vienos krypties kelių. Kiekvienas kelias jungia du skirtingus mazgus x_i ir y_i , kryptis yra iš x_i į y_i . Ant i-tojo kelio yra w_i tonų sniego. Kiekvieną dieną valymo mašina startuoja iš mazgo A, važiuoja iki mazgo B pagal kelių kryptis ir sustoja. Bet kurią mazgą (A ir B taip pat) galima pravažiuoti per dieną daug kartų. Vienu pravažiuoimu nuvaloma viena tono sniego. Ten, kur sniego nėra, mašina važiuoti negali. Kai kurie keliai turi istorinę vertę, todėl juos nuvalyti būtina. Be to, iš mazgo A galima pasiekti bet kurią istorinės vertės kelią, važiuojant tik istorinės vertės keliais. Parašykite programą, kuri nurodytų, kokia tvarka reiktų valyti kelius, kad valymas vyktų kuo daugiau dienų ir būtų nuvalyti visi istorinės vertės keliai.

Duomenys. Tekstinio failo 'U3.txt' pirmoje eilutėje nurodytas N ($2 \leq N \leq 100$) – mazgų kiekis, M ($0 \leq M \leq 1000$) – kelių kiekis, mazgai A ir B ($1 \leq A, B \leq N$; $A \neq B$). Tolesnėse M eilučių įvardinti vienos krypties keliai 4 sveikaisiais skaičiais x_i (pradžios mazgas), y_i (pabaigos mazgas), w_i (sniego tonos), t_i (1 – istorinis kelias, 0 – ne) ($1 \leq x_i, y_i \leq N$, $x_i \neq y_i$, $0 \leq w_i \leq 100$, $0 \leq t_i \leq 1$).

Rezultatai. Į ekraną pirmoje eilutėje išveskite maksimalų dienų skaičių, o tolesnėse eilutėse – kiekvienos dienos kelius, išvardijant pravažiuojamus mazgus pagal kelio kryptį.

U3. txt	Ekranas
4 7 1 4 1 2 3 1 2 1 100 0 2 4 1 0 1 3 1 0 3 4 4 0 2 3 2 1 1 4 2 0	6 1 3 4 1 4 1 4 1 2 4 1 2 3 4 1 2 3 4