

1.

A városi rendőrség egy veszélyes bűnöző elfogását tervezi, aki gépkocsival folyamatosan közlekedik a város utcáin. A rendőrségnek korlátozottak a lehetőségei, nem tud például minden kereszteződésbe rendőrt állítani az elfogás érdekében. Ravasz őrmesternek az alábbi kitűnő ötlete támadt. Egy kijelölt K kereszteződésből indulva bejárja a város utcáit és úgy egyirányúsítja azokat, hogy a bűnöző előbb-utóbb úgyis eljut a K kereszteződésbe, ahol egy másik rendőr várakozik, aki elfogja a bűnözőt. A kapitánynak nagyon tetszik az ötlet, de kiköti, hogy Ravasz őrmesternek is be kell tartania a közlekedési szabályokat:

A már egyirányúsított utcában az őrmester is csak a jelzett irányban közlekedhet. Továbbá, ha az A kereszteződésből a B -be vezető utcát akarja egyirányúsítani, azt csak úgy teheti, hogy elmegy az A -ba, ott elhelyez egy 'behajtani tilos' táblát B irányában, ezután elmegy az utcában a B kereszteződésig és ott elhelyezi az A -irányába mutató 'egyirányú utca' táblát. Ugyanazon utcában többször is járhat, de csak a már beállított irányban.

Így a megoldás egyértelműen leírható az őrmester útja során érintett kereszteződések sorozatával, mivel ha az útja során egy olyan A kereszteződésbe ér, amelyből a B irányában halad tovább, akkor ha ez az utca még nem volt egyirányúsítva, akkor egyirányúsítja, egyébként csak halad tovább az utcában. A város úthálózata összefüggő, azaz minden kereszteződésből el lehet jutni bármely másik kereszteződésbe.

Írjon programot, amely kiszámít egy olyan bejárési sorozatot, amelyet bejárva és elvégezve az egyirányúsítást, a bűnöző előbb-utóbb feltűnik abban a kereszteződésben, ahonnan az őrmester indult. Minden utcában mindkét irányban lehet közlekedni, de az utcán megfordulni tilos. Kereszteződésben azonban meg lehet fordulni. Az egyirányúsítás következtében csak az a kereszteződés lehet olyan, hogy nem indul belőle egyetlen egyirányú út sem, amelyikből az őrmester indult.

Az ELFOGO.BE állomány első sorában a kereszteződések ($1 \leq N \leq 200$) száma, a második sorában az utcák ($1 \leq M \leq 10000$) száma van. A következő M sor mindegyike egy A B számpárt ($1 \leq A, B \leq 200$) tartalmaz egy szóközzel elválasztva, ami azt jelenti, hogy az A kereszteződésből megy egy (kétirányú) utca a B kereszteződésig. Két kereszteződés között legfeljebb egy utca lehet.

Az ELFOGO.KI állomány első sorába az őrmester bejáró útját megadó kereszteződések L számát, a második sorba pedig a bejárt L kereszteződés sorszámát kell írni, egy-egy szóközzel elválasztva.

2.

Egy térképrészleten minden ország be van festve valamilyen színnel. Egy kommandónak el kell jutnia az egyik országból egy másikba. A kommandósoknak, hogy ne ismerjék fel őket, minden országban olyan színű trikót kell hordaniuk, amilyen az adott ország színe a térképen. **Határozza meg azt a legrövidebb útvonalat, amelynek a bejárásához a lehető legkevesebb számú trikóra van szüksége a kommandó tagjainak.**

A KOMx.BE állomány első sorában egy egész szám ($2 \leq N \leq 100$) van: az országok száma. A következő N sorban ismeretlen számú egész szám található: az egyes országok és szomszédjaik országcódja. A következő N sorban az egyes országok színkódja található (ugyancsak egész számok, az országcódok növekvő sorrendjében). Az utolsó sorban 2 egész szám azonosítja a kiindulási országot és a célországot.

A KOMx.KI állományba és a képernyőre a felhasználandó trikók színkódját és az útvonalon fekvő országok országcódját kell kiírni. Ha több azonos értékű megoldás van, elég egyet megadni.

Megjegyzés [SzP1]:

Algoritmus: **nem** súlyozott gráf legrövidebb út...
Menet közben feljegyzendő minden érintett ponthoz az odáig szükséges minimális színek számát és színeket.

3.

Egy pszichológiai vizsgálatban azt jegyezték fel, hogy egy N tanulóból álló csoportban ki kit tart a legszimpatikusabbnak.

A BARAT.BE állomány első sorában a tanulók száma ($2 \leq N \leq 1000$), a további N sorban a tanulók neve és a nekik legszimpatikusabb tanuló neve, egyetlen szóközzel elválasztva. A nevek szóközt nem tartalmazhatnak, hosszuk legfeljebb 20.

Készítsen programot, amely a képernyőre és a BARAT.KI állományba írja (soronként) az alábbiakat:

- A. Az első sorba a **senkinek sem szimpatikus tanuló** nevét, egy-egy szóközzel elválasztva.
- B. A második sorba azokat a párokat, amelyek **tagjai egymásnak a legszimpatikusab-
bak, de senki másnak nem szimpatikus egyik sem**. A párok tagjai közé kötőjelet (-),
a párok közé pedig szóközt kell írni.
- C. A harmadik sorba a **legkedveltebb tanuló** nevét (akit a legtöbben adtak meg leg-
szimpatikusabbként), ha több van, akkor mindegyiket, egy-egy szóközzel elvá-
lasztva.

Megjegyzés [SzP2]:

Bemenő fokszám=0

Megjegyzés [SzP3]:

Bemenő fokszámok=1 és van el: $A \rightarrow b$,
 $B \rightarrow A$

Megjegyzés [SzP4]:

Bemenő fokszám maximális

4.

Egy vállalat az ország különböző városaiban levő üzemeiben alkatrészeket termel. A heti termelést a hét végén kamionokkal szállítja a központi raktárába. A kamionforgalom korlátozása miatt minden városból pontosan egy másik városba (egy irányban) mehetnek a kamionok közvetlenül. Ezért a vállalat úgy tervezi a szállításokat, hogy minden olyan városból, amelybe más városból nem lehet eljutni, egy-egy kamiont indít, a többi városból viszont egyet sem. A korlátozások miatt így minden kamion útja a központi raktárig egyértelműen meghatározott.

Minden kamion, amely útja során áthalad egy városon, az ott termelt alkatrészekből bármennyit felvehet, feltéve, hogy nincs tele. Ismerve a városokban termelt alkatrészek számát, ki kell számítani azt a legkisebb kamion kapacitást, amellyel a szállítás megoldható, ha minden kamion azonos kapacitású.

A KAMION.BE állomány első sorában a városok N ($1 < N \leq 200$) száma van A központi raktár az 1. városban van, és onnan nem kell szállítani. Az állomány következő $N-1$ sorának mindegyike két egész számot tartalmaz, egy szóközzel elválasztva. Az állomány I -edik sorában az első szám azt a várost adja meg, ahova az I -edik városból mehet kamion. A második szám pedig az I -edik városban termelt alkatrészek száma. (Az 1. városból kivezető út nincs megadva.)

A KAMION.PAS (vagy KAMION.C) program a KAMION.KI állományba és a képernyőre azt a legkisebb kamion kapacitást (egész szám) írjon ki, amekkora kapacitású kamionokkal az összes alkatrész elszállítható.

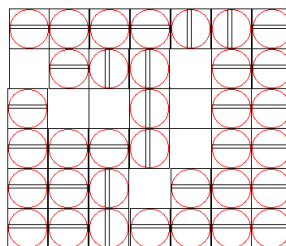
5.

Egy jármű négyzetácsos elrendezésű pályaelemekből álló, M sorból, N oszlopból álló pályán mozoghat. Minden pályaelem vagy üres, vagy a közepén áthaladó sínt tartalmaz, amelyen a jármű haladhat. Egy pályaelem négy szomszédja a négyzetácsos elrendezésben a tőle balra, jobbra, lefelé vagy fölfelé lévő pályaelem. A jármű egy lépésben a következő három lehetséges mozgást végezheti:

1. 90 fokkal elfordítja azt a pályaelemet, amelyen éppen áll.
2. Átmegy egy szomszédos pályaelemre, feltéve, hogy azon a sín olyan irányban áll, hogy az csatlakozik az aktuális pályaelemen lévő sínhez.
3. 90 fokkal elfordít egy szomszédos pályaelemet.

Írjon programot, amely kiszámítja azt a legkevesebb lépésszámot, amely megtételével a jármű a pálya bal felső (1,1) pontjából eljuthat a jobb alsó (M,N) pontjába.

A PALYA.BE állomány első sorában a pálya méretét megadó M N számpár van egy szóközzel elválasztva ($1 \leq M, N \leq 100$). A következő M sor mindegyike N számot tartalmaz egy-egy szóközzel elválasztva:



0: az adott pályaelem nem tartalmaz sít (üres),

1: a pályaelemen a sín vízszintes irányban áll,

2: a pályaelemen a sín függőleges irányban áll.

A képernyőre és a PALYA.KI állomány első és egyetlen sorába azt a legkisebb lépésszámot kell írni, amely megtételével a jármű a pálya bal felső pontjából eljuthat a jobb alsó pontjába. Ha a jármű nem tud eljutni, akkor a -1 értéket kell kiírni.

6.

Egy nagyszabású építkezéshez részletes tervet készítettek, amelyben többek között azt is megadják, hogy egy-egy munka hány napig tart. A munkák sorrendje nem tetszőleges. A tervben a sorrendet **a b** feltételpárok írják elő. Az **a b** feltételpár azt jelenti, hogy a **b** munkát csak az **a** munka befejezése után lehet elkezdni. Az építkezés az 1. napon kezdődik. **N** munka esetén a munkákat az **1, ..., N** számokkal azonosítjuk. (Természetesen irányított kör nincs.)

Írjon programot az alábbi részfeladatok megoldására!

A. Számítsa ki, hogy a terv végrehajtásához minimálisan hány nap szükséges!

B. Adja meg az egyes munkák legkorábbi kezdési idejét a számuk sorrendjében! (Ez az a legkorábbi időpont, amikor az adott munkát el lehet kezdeni, mert az összes korábbi, ugyancsak a lehető legkorábban elkezdett munka már befejeződött.)

A TERVx.BE állomány első sora a munkák **N** ($1 \leq N \leq 100$) számát tartalmazza. A második sorban **N** db pozitív egész szám van, ahol az **i**-edik szám a sorban az **i**-edik munka végrehajtásához szükséges napok száma (≤ 100). A harmadik sorban a feltételpárok **M** ($0 \leq M \leq 1000$) száma van. A következő **M** sor mindegyike egy-egy **a b** számpárt tartalmaz ($1 \leq a, b \leq N$); ami azt jelenti, hogy a **b** munkát csak az **a** munka befejezése után lehet elkezdni.

A TERVx.KI szöveges állományba és a képernyőre két sorba kell írni az **A.**, **B.** részfeladatok megoldását. Ha valamelyik részfeladat megoldása hiányzik, akkor a megfelelő sor legyen üres! A **B.** részfeladat esetén (2. sor) a számsorozat elemeit egy-egy szóköz válassza el.

7.

Egy nagyszabású építkezéshez részletes tervet készítettek, amelyben többek között azt is megadják, hogy egy-egy munka hány napig tart. A munkák sorrendje nem tetszőleges. A tervben a sorrendet **a b** feltételpárok írják elő. Az **a b** feltételpár azt jelenti, hogy a **b** munkát csak az **a** munka befejezése után lehet elkezdni. Az építkezés az 1. napon kezdődik. **N** munka esetén a munkákat az **1, ..., N** számokkal azonosítjuk.

Adjon meg egy ún. kritikus utat! (Kritikus útnak nevezzük olyan munkák egy sorozatát, amelyeket az adott sorrendben lehet elvégezni, és

- a munkák összideje megegyezik a terv végrehajtásához minimálisan szükséges idővel, továbbá
- minden egyes munka legkorábbi és legkésőbbi kezdési ideje ugyanaz.)

A TERVx.BE állomány első sora a munkák **N** ($1 \leq N \leq 100$) számát tartalmazza. A második sorban **N** db pozitív egész szám van, ahol az **i**-edik szám a sorban az **i**-edik

Megjegyzés [SzP5]:

A gráf felépítése után a lehetséges kezdő- és végpontok megkeresésével kezdeni a munkát. (Befok=0 / kifok=0)

Megjegyzés [SzP6]:

A kezdőmunkákból (kezdőpontokból), ahonnan 1-kor indul a munka, az egyes végmunkákhoz (végpontokhoz) keressük a maximális hosszúságú utat. (A legrövidebbhez hasonlóan; de ha ugyanahhoz a ponthoz érünk, növelhetjük az úthosszt, s ekkor a sorba újra bekerül.)

Megjegyzés [SzP7]:

Valamely kezdőpontból odáig szükséges leghosszabb út hossza.

munka végrehajtásához szükséges napok száma (≤ 100). A harmadik sorban a feltételpárok **M** ($0 \leq M \leq 1000$) száma van. A következő **M** sor mindegyike egy-egy **a b** számpárt tartalmaz ($1 \leq a, b \leq N$); ami azt jelenti, hogy a **b** munkát csak az **a** munka befejezése után lehet elkezdni.

A TERV_x.KI szöveges állományba és a képernyőre 2 sorba írja a választ. Az első sor a kritikus út pontjai számát tartalmazza, a 2. sor magukat a pontok sorszámain, egy-egy szóközzel elválasztva.

8.

Egy nagyszabású építkezéshez részletes tervet készítettek, amelyben többek között azt is megadják, hogy egy-egy munka hány napig tart. A munkák sorrendje nem tetszőleges. A tervben a sorrendet **a b** feltételpárok írják elő. Az **a b** feltételpár azt jelenti, hogy a **b** munkát csak az **a** munka befejezése után lehet elkezdni. Az építkezés az 1. napon kezdődik. **N** munka esetén a munkákat az **1, ..., N** számokkal azonosítjuk.

Adja meg, hogy minimum hány vállalkozóval kell szerződést kötni a munkák elvégzésére, ha egy vállalkozó egy időben csak egy munkán tud dolgozni, és ha minden munka a lehető legkorábban kezdődik!

A TERV_x.BE állomány első sora a munkák **N** ($1 \leq N \leq 100$) számát tartalmazza. A második sorban **N** db pozitív egész szám van, ahol az *i*-edik szám a sorban az *i*-edik munka végrehajtásához szükséges napok száma (≤ 100). A harmadik sorban a feltételpárok **M** ($0 \leq M \leq 1000$) száma van. A következő **M** sor mindegyike egy-egy **a b** számpárt tartalmaz ($1 \leq a, b \leq N$); ami azt jelenti, hogy a **b** munkát csak az **a** munka befejezése után lehet elkezdni.

A TERV_x.KI szöveges állományba és a képernyőre kell írni megoldást.

9.

Egy nagyszabású építkezéshez részletes tervet készítettek, amelyben többek között azt is megadják, hogy egy-egy munka hány napig tart. A munkák sorrendje nem tetszőleges. A tervben a sorrendet **a b** feltételpárok írják elő. Az **a b** feltételpár azt jelenti, hogy a **b** munkát csak az **a** munka befejezése után lehet elkezdni. Az építkezés az 1. napon kezdődik. **N** munka esetén a munkákat az **1, ..., N** számokkal azonosítjuk.

Adja meg, hogy minimálisan mennyi az összes állási idő! (Az állási idő abból adódik, hogy egy vállalkozó két, időben egymást követő munka elvégzése között várakozásra kényszerülhet, ha a soron következő munkát még nem kezdheti el.) Feltesszük, hogy minden munka a lehető legkorábban kezdődik, és a kivitelezők a lehető legkevesebb vállalkozót alkalmazzák.

A TERV_x.BE állomány első sora a munkák **N** ($1 \leq N \leq 100$) számát tartalmazza. A második sorban **N** db pozitív egész szám van, ahol az *i*-edik szám a sorban az *i*-edik munka végrehajtásához szükséges napok száma (≤ 100). A harmadik sorban a feltételpárok **M** ($0 \leq M \leq 1000$) száma van. A következő **M** sor mindegyike egy-egy **a b** számpárt tartalmaz ($1 \leq a, b \leq N$); ami azt jelenti, hogy a **b** munkát csak az **a** munka befejezése után lehet elkezdni.

A TERV_x.KI szöveges állományba és a képernyőre kell írni megoldást.

10.

Egy üzemben a gyártást automatizálták. A szerszámgépek egy nagy gépcsarnokban négyzetrács mentén vannak elhelyezve. A műszak végén robotok gyűjtik össze a szerszámgépek gyártotta alkatrészeket. A robotok négyzetrács alakú pályán mozognak a szerszámgépek fölötti térben. A négyzetrács bal felső sarkából, az (1,1) pontból indulnak, és a jobb alsó sarokba viszik el az alkatrészeket. A robotokat úgy tervezték, hogy csak jobbra és „lefelé” haladhatnak.

Írjon olyan programot, amely kiszámítja, hogy **minimálisan hány robotot kell elindítani** az összes alkatrész begyűjtéséhez, feltéve hogy minden robot tetszőleges számú alkatrészt szállíthat.

A feladat megoldásához rendelkezésre áll az üzemszámok térképe, amely egy 0 és 1 elemeket tartalmazó $M \times N$ -es mátrix (M a sorok, N az oszlopok száma). A 0 jelöli az üres helyet gépcsarnokban, az 1 pedig azt, hogy azon a helyen szerszám van.

A ROBOT.BE állomány első sorában az M és N értékek vannak, egyetlen szóközzel elválasztva ($1 \leq M, N \leq 100$). A következő M sor mindegyike a térkép egy-egy sora, azaz minden sorban N szám (0 vagy 1) van, a számokat ismét egyetlen szóköz választja el.

A feladat megoldásaként egyetlen számot kell írni a ROBOT.KI állományba.

11.

A Mars felszínén különleges járművekkel kőzetmintákat gyűjtenek és juttatnak el az adóállomásra. A járművek csak négyzetácsos útvonalon tudnak közlekedni, a kiindulási helytől csak dél vagy kelet felé és kizárólag sziklamentes terepen tudnak haladni. Térkép mutatja, hogy hol (melyik rácsponthoz) van (egy) kőzetminta, és hogy hol sziklás a terep. A térkép egy $M \times N$ -es mátrix, amely a különböző területfajtákat a következőképpen jelöli:

üres terület: 0

kőzetmintás terület: 1

sziklás terület: 2

A járművek az (1,1) pontból indulnak, ez a térképen ábrázolt terep bal felső sarka. Az adóállomás az (M,N) pontban van (M a sorindex, N az oszlopindex).

Írjon programot, amely kiszámítja **az összegyűjthető kőzetminták számának maximumát**, ha feltesszük, hogy elegendő számú jármű áll rendelkezésre.

A MARS.BE állomány első sorában az M és N értékek vannak egyetlen szóközzel elválasztva ($1 \leq M, N \leq 100$). A következő M sor mindegyike a térkép egy sorát adja meg, azaz minden sorban N szám (0, 1 vagy 2) van, ahol a számokat ismét egyetlen szóköz választja el.

A feladat megoldásaként egyetlen számot kell írni a képernyőre és a MARS.KI állományba.

12.

Johannesburg rendőrsége az összes bűnözőt ismeri a városban. Tudják, hogy két bűnöző akkor tartozik ugyanabba a bandába, ha legalább egyszer közösen követtek el bűncselekményt (a bandák tagjai közül csak a közös bűncselekményt elkövetők ismerik egymást).

A bűnözőket a nevük helyett a sorszámukkal azonosítják (1-től kezdve, folyamatos számozással). A rendőrség a BANDAx.BE szöveges állományban tárolja az egy bandába tartozó bűnözők sorszámát. Az állomány első sorában a bűnözők száma (maximum 50), minden további sorában (maximum 100) pedig két bűnöző sorszámát van (egyetlen szóközzel elválasztva), akik közösen követtek el bűncselekményt. Lehetnek olyan magányos bűnözők is, akik egyetlen bandának sem tagjai (ahol bandákkal kapcsolatos eredményt kérünk, ott őket nem szabad figyelembe venni).

Készítsen programot, amely a BANDAx.KI állományba és a képernyőre kiírja, egy-egy sorba, az alábbiakat. (Ha valamelyik megoldás több számból áll, akkor közéjük egy-egy szóközt kell írni.

A. A **magányos bűnözők sorszámát**.

B. A **bandák számát**.

C. Az egyes **bandák legtöbb kapcsolattal rendelkező tagját**.

Megjegyzés [SzP8]:

Mo.-ötlet:

Gráf-felepítés úgy, h. a pont első irányított éle a jobb szomszédhoz, a második az alsóhoz vezet.

Annyi mélységi bejárás (=robotszám), ahány kell ahhoz, h. az összes ponthoz eljussunk; ha egy ponthoz eljutottunk, a pont törölhető.

Megjegyzés [SzP9]:

Mo.-ötlet:

Az előzőhöz hasonló, a válasz az elért pontok száma.

A beolvasásnál lehet optimalizálni: ahova nem lehet eljutni, azt a gráfba fel sem kell venni.

Megjegyzés [SzP10]:

Fokszám=0

Megjegyzés [SzP11]:

komponensek

Megjegyzés [SzP12]:

Komponensenként: fokszám maximumú elem.

13.

Johannesburg rendőrsége az összes bűnözőt ismeri a városban. Tudják, hogy két bűnöző akkor tartozik ugyanabba a bandába, ha legalább egyszer közösen követtek el bűncselekményt (a bandák tagjai közül csak a közös bűncselekményt elkövetők ismerik egymást).

A bűnözőket a nevük helyett a sorszámmal azonosítják (1-től kezdve, folyamatos számozással). A rendőrség a BANDAx.BE szöveges állományban tárolja az egy bandába tartozó bűnözők sorszámmát. Az állomány első sorában a bűnözők száma (maximum 50), minden további sorában (maximum 100) pedig két bűnöző sorszáma van (egyetlen szóközzel elválasztva), akik közösen követtek el bűncselekményt. Lehetnek olyan magányos bűnözők is, akik egyetlen bandának sem tagjai (ahol bandákkal kapcsolatos eredményt kérünk, ott őket nem szabad figyelembe venni).

Készítsen programot, amely a BANDAx.KI állományba és a képernyőre kiírja, egy-egy sorba, az alábbiakat. (Ha valamelyik megoldás több számból áll, akkor közéjük egy-egy szóközt kell írni.

- A. A legnagyobb banda tagjainak számát.
- B. Az egyes bandák kulcsembereit, azaz azokat, akiket börtönbe zárva a bandák a lehető legtöbb független részre esnek szét. Az egyes bandák azon minimális számú kulcsembereit, akiket börtönbe zárva a bandák legalább két részre esnek.

Megjegyzés [SzP13]:
Legtöbb pontú komponens

Megjegyzés [SzP14]:
Összetett, ezért inkább a folytatást...

Megjegyzés [SzP15]:
A legtöbb fokszámú pontokból annyit törölni, h. ketté essen a komponens.

14.

Johannesburg rendőrsége az összes bűnözőt ismeri a városban. Tudják, hogy két bűnöző akkor tartozik ugyanabba a bandába, ha legalább egyszer közösen követtek el bűncselekményt (a bandák tagjai közül csak a közös bűncselekményt elkövetők ismerik egymást).

A bűnözőket a nevük helyett a sorszámmal azonosítják (1-től kezdve, folyamatos számozással). A rendőrség a BANDAx.BE szöveges állományban tárolja az egy bandába tartozó bűnözők sorszámmát. Az állomány első sorában a bűnözők száma (maximum 50), minden további sorában (maximum 100) pedig két bűnöző sorszáma van (egyetlen szóközzel elválasztva), akik közösen követtek el bűncselekményt. Lehetnek olyan magányos bűnözők is, akik egyetlen bandának sem tagjai (ahol bandákkal kapcsolatos eredményt kérünk, ott őket nem szabad figyelembe venni).

Készítsen programot, amely a BANDAx.KI állományba és a képernyőre kiírja a biztonságosan szervezett bandák számát (és tagságát külön sorban), azaz az olyanokét, amelyekből a rendőrség nem tud úgy letartóztatni egyetlen bűnözőt, hogy a banda több részre essen szét.

Megjegyzés [SzP16]:
A komponens bármely pontja törlése után is összefüggő marad.

15.

Johannesburg rendőrsége az összes bűnözőt ismeri a városban. Tudják, hogy két bűnöző akkor tartozik ugyanabba a bandába, ha legalább egyszer közösen követtek el bűncselekményt (a bandák tagjai közül csak a közös bűncselekményt elkövetők ismerik egymást).

A bűnözőket a nevük helyett a sorszámmal azonosítják (1-től kezdve, folyamatos számozással). A rendőrség a BANDAx.BE szöveges állományban tárolja az egy bandába tartozó bűnözők sorszámmát. Az állomány első sorában a bűnözők száma (maximum 50), minden további sorában (maximum 100) pedig két bűnöző sorszáma van (egyetlen szóközzel elválasztva), akik közösen követtek el bűncselekményt. Lehetnek olyan magányos bűnözők is, akik egyetlen bandának sem tagjai (ahol bandákkal kapcsolatos eredményt kérünk, ott őket nem szabad figyelembe venni).

Készítsen programot, amely a BANDAx.KI állományba és a képernyőre kiírja a kockázatosan szervezett bandák számát, azaz az olyanokét, amelyekben csak a főnök

ismer mindenkit, és ezért az ő letartóztatása esetén a banda magányos bűnözőkre esik szét.

16.

Johannesburg rendőrsége az összes bűnözőt ismeri a városban. Tudják, hogy két bűnöző akkor tartozik ugyanabba a bandába, ha legalább egyszer közösen követtek el bűncselekményt (a bandák tagjai közül csak a közös bűncselekményt elkövetők ismerik egymást).

A bűnözőket a nevük helyett a sorszámukkal azonosítják (1-től kezdve, folyamatos számozással). A rendőrség a BANDAx.BE szöveges állományban tárolja az egy bandába tartozó bűnözők sorszámát. Az állomány első sorában a bűnözők száma (maximum 50), minden további sorában (maximum 100) pedig két bűnöző sorszámát van (egyetlen szóközzel elválasztva), akik közösen követtek el bűncselekményt. Lehetnek olyan magányos bűnözők is, akik egyetlen bandának sem tagjai (ahol bandákkal kapcsolatos eredményt kérünk, ott őket nem szabad figyelembe venni).

Készítsen programot, amely a BANDAx.KI állományba és a képernyőre kiírja a teljesen szervezett bandák számát, azaz az olyanokét, amelyekben mindenki ismer mindenkit.

17.

Egy turista a turistaházból a lehető legrövidebb idő alatt szeretne eljutni a forráshoz. Útja hegyeken, völgyeken, szakadékokon vezet át. Minden út észak-déli vagy kelet-nyugati irányú. Előfordulhat, hogy a nagy szintkülönbség miatt kerület kell tennie. Segítsen neki az útvonal megtervezésében!

Az úthálózat alkotta rácsot, ahol a rácsponthoz az útelágazások, egy NxM-es tömbbel ábrázoljuk; az egyes cellák értéke a domborzat adott rácsponthoz mért magassága. A turistaház a bal felső sarokban (az (1,1) koordinátájú pontban), a forrás a jobb alsó sarokban (az (N,M) koordinátájú pontban) található.

Egy útszakasz megtétele sík terepen egy percig tart. Minden méter szintkülönbség leküzdése további egy perccel növeli meg ezt az időt. Ha két pont szintkülönbsége nagyobb az előre megadott K értéknél, a turistának kerület kell tennie.

Írjon programot, amely meghatározza a túra megtételéhez szükséges legrövidebb időt és az útvonal szakaszainak számát!

A HEGYx.BE állomány első sorában N, M és K értéke ($1 \leq N \leq 100$, $1 \leq M \leq 100$) található, egy-egy szóközzel elválasztva. A következő N sor mindegyike M rácsponthoz magasságát tartalmazza, szóközzel elválasztva.

A HEGYx.KI állományba és a képernyőre a túra megtételéhez szükséges legrövidebb időt és az útszakaszok számát kell írni. Ha nem lehet eljutni a célba, a program írja ki a NEM ER CÉLBA szöveget!

18.

Egy nagyszabású építkezéshez részletes tervet készítettek, amelyben többek között azt is megadják, hogy egy-egy munka hány napig tart. A munkák sorrendje nem tetszőleges. A tervben a sorrendet a b feltételpárok írják elő. Az a b feltételpár azt jelenti, hogy a b munkát csak az a munka befejezése után lehet elkezdni. Az építkezés az 1. napon kezdődik. N munka esetén a munkákat az 1,...,N számokkal azonosítjuk. Feltevésként, hogy egyetlen egy munkával kezdődik és végződik az építkezés. (természetesen irányított kör nincs)

Írjon programot az alábbi részfeladatok megoldására!

A. Számítsa ki, hogy a terv végrehajtásához minimálisan hány nap szükséges!

Megjegyzés [SzP17]:

A komponensnek van olyan pontja, amelyből minden pontba vezet el.

Megjegyzés [SzP18]:

A többi pont fokszáma 1.

Megjegyzés [SzP19]:

A komponens olyan, h. ha k pontból áll, akkor k_alatt_a_2 fokszámú minden pontja.

Megjegyzés [SzP20]:

Mo-ötlet:
Gráf-felépítés – az élék hossza a megteendő szintkülönbség ($h_a \leq K$).
Súlyozott gráfbeli legrövidebb út megkeresése.

Megjegyzés [SzP21]:

A gráf felépítése után a kezdő- és végpont megkeresésével kell kezdeni a munkát.
(Befok=0 / kifok=0)

Megjegyzés [SzP22]:

A kezdőmunkából (kezdőpontból), ahonnan 1-kor indul a munka, a végmunkához (végponthoz) keressük a maximális hosszúságú utat. (A legrövidebbhez hasonlóan; de ha ugyanahhoz a ponthoz érve, növelhetjük az úthosszt, s ekkor a sorba újra bekerül.)

B. Adja meg a munkák legkésőbbi kezdési idejét a számuk sorrendjében! (Ez az a legkésőbbi időpont, amikor az adott munkát el kell kezdeni ahhoz, hogy ne késleltesse az építkezés befejezését.)

A TERV_x.BE állomány első sora a munkák N ($1 \leq N \leq 100$) számát tartalmazza. A második sorban N db pozitív egész szám van, ahol az i -edik szám a sorban az i -edik munka végrehajtásához szükséges napok száma (≤ 100). A harmadik sorban a feltételpárok M ($0 \leq M \leq 1000$) száma van. A következő M sor mindegyike egy-egy a b számpárt tartalmaz ($1 \leq a, b \leq N$); ami azt jelenti, hogy a b munkát csak az a munka befejezése után lehet elkezdni.

A TERV_x.KI szöveges állományba és a képernyőre két sorba kell írni az A., B. részfeladatok megoldását. Ha valamelyik részfeladat megoldása hiányzik, akkor a megfelelő sor legyen üres! A B. részfeladat esetén (2. sor) a számsorozat elemeit egy-egy szóköz válassza el.

19.

Egy nagyszabású építkezéshez részletes tervet készítettek, amelyben többek között azt is megadják, hogy egy-egy munka hány napig tart. A munkák sorrendje nem tetszőleges. A tervben a sorrendet a b feltételpárok írják elő. Az a b feltételpár azt jelenti, hogy a b munkát csak az a munka befejezése után lehet elkezdni. Az építkezés az 1. napon kezdődik. N munka esetén a munkákat az $1, \dots, N$ számokkal azonosítjuk. Feltesszük, hogy egyetlen egy munkával kezdődik és végződik az építkezés (természetesen irányított kör nincs).

Írjon programot az alábbi részfeladatok megoldására!

A. Adja meg az egyes munkák legkorábbi kezdési idejét a számuk sorrendjében! (Ez az a legkorábbi időpont, amikor az adott munkát el lehet kezdeni, mert az összes korábbi, ugyancsak a lehető legkorábban elkezdett munka már befejeződött.)

B. Adja meg, hogy minimum hány vállalkozóval kell szerződést kötni a munkák elvégzésére, ha egy vállalkozó egy időben csak egy munkán tud dolgozni, és ha minden munka a lehető legkorábban kezdődik!

A TERV_x.BE állomány első sora a munkák N ($1 \leq N \leq 100$) számát tartalmazza. A második sorban N db pozitív egész szám van, ahol az i -edik szám a sorban az i -edik munka végrehajtásához szükséges napok száma (≤ 100). A harmadik sorban a feltételpárok M ($0 \leq M \leq 1000$) száma van. A következő M sor mindegyike egy-egy a b számpárt tartalmaz ($1 \leq a, b \leq N$); ami azt jelenti, hogy a b munkát csak az a munka befejezése után lehet elkezdni.

A TERV_x.KI szöveges állományba és a képernyőre két sorba kell írni az A., B. részfeladatok megoldását. Az A. részfeladat esetén (1. sor) a számsorozat elemeit egy-egy szóköz válassza el.

20.

Egy nagyszabású építkezéshez részletes tervet készítettek, amelyben többek között azt is megadják, hogy egy-egy munka hány napig tart. A munkák sorrendje nem tetszőleges. A tervben a sorrendet a b feltételpárok írják elő. Az a b feltételpár azt jelenti, hogy a b munkát csak az a munka befejezése után lehet elkezdni. Az építkezés az 1. napon kezdődik. N munka esetén a munkákat az $1, \dots, N$ számokkal azonosítjuk. Feltesszük, hogy egyetlen egy munkával kezdődik és végződik az építkezés (természetesen irányított kör nincs).

Adja meg, hogy mennyi az összes állási idő! (Az állási idő: két, időben egymást követő munka elvégzése között várakozásra van szükség, ha a soron következő munkát még nem kezdődhet el.) Feltesszük, hogy minden munka a lehető legkorábban kezdődik.

Megjegyzés [SzP23]:

A legkorábbi kezdési idő (a kezdőpontból odáig szükséges leghosszabb út hossza) megnövelhető azzal az értékkel, amelyet így kapunk: az őt közvetlenül követő pontok legkorábbi kezdési idő és a végrehajtási idő különbségeinek a minimuma.

Megjegyzés [SzP24]:

A gráf felépítése után a kezdő- és végpont megkeresésével kell kezdeni a munkát. (Befok=0 / kifok=0)

Megjegyzés [SzP25]:

Valamely kezdőpontból odáig szükséges leghosszabb út hossza.

Megjegyzés [SzP26]:

A súlyozott gráf szélességi bejárása közepette nyilván tartjuk az addig szükséges vállalkozókat (mikor fog végezni). Egy következő ponthoz jutva eldöntjük, van-e olyan vállalkozó, aki már nem dolgozik, ha van, dolgoztatjuk ezzel a munkával, ha nincs, újat veszünk föl.

Megjegyzés [SzP27]:

Mo.:-ötlet:
A gráf felépítése után a kezdő- és végpont megkeresésével kell kezdeni a munkát. (Befok=0 / kifok=0)
Meghatározzuk az egyes munkák legkorábbi és legkésőbbi kezdési idejét: A legkorábbi kezdési idő (a kezdőpontból odáig szükséges leghosszabb út hossza) megnövelhető azzal az értékkel, amelyet így kapunk: az őt közvetlenül követő pontok legkorábbi kezdési idő és a végrehajtási idő különbségeinek a minimuma.
E két értékpár különbség-összege a válasz.

A TERV_x.BE állomány első sora a munkák **N** ($1 \leq N \leq 100$) számát tartalmazza. A második sorban **N** db pozitív egész szám van, ahol az *i*-edik szám a sorban az *i*-edik munka végrehajtásához szükséges napok száma (≤ 100). A harmadik sorban a feltételepárok **M** ($0 \leq M \leq 1000$) száma van. A következő **M** sor mindegyike egy-egy **a b** számpárt tartalmaz ($1 \leq a, b \leq N$); ami azt jelenti, hogy a **b** munkát csak az **a** munka befejezése után lehet elkezdni.

A TERV_x.KI szöveges állományba és a képernyőre kell írni megoldást.

21.

Ismerjük **N** csillag térbeli elhelyezkedését; és egy **K** (távolság-paraméter). A csillagok „hajlamosak” ún. csillaghalmazokba tömörülni. Soroljunk be minden csillagot a saját csillaghalmazába! A halmazba tartozás kritériumai a következők: az **A** és a **B** csillag ugyanabba a halmazba tartozik, ha

1. $Távolság(A,B) < K$ ($Távolság(A,B) := \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2 + (A_z - B_z)^2}$);
2. van olyan **C** csillag, amelyikkel egy halmazba tartozik az **A** és a **B** is.

Készítsen **programot**, amely

- A. meghatározza a **csillaghalmazok** számát;
- B. az egyes csillaghalmazokat; és
- C. az egyes **csillaghalmazok átmérőjét** (a két legtávolabbi csillag távolságát).

A CSILLAG_x.BE szöveges állomány első sorában a csillagok **N** száma, valamint a **K** távolságparaméter található, szóközzel elválasztva. A további **N** sorban az egyes csillagok térbeli koordinátái (*x*-, *y*-, *z*-koordináta sorrendben), szintén egy-egy szóközzel elválasztva egymástól.

A CSILLAG_x.KI szöveges állományba és a képernyőre kell írni megoldást, amelynek első sorába a csillaghalmazok CsHDb száma (az A. részfeladat megoldása) kerül, s az alatta levő CsHDb darab sorba az egyes csillaghalmazokba tartozó csillagok (a beolvasás sorrendje szerinti) sorszámai kerülnek egy-egy szóközzel elválasztva egymástól. (Ez a B. részfeladat.) Majd újabb CsHDb sorban az egyes csillaghalmazok átmérői szerepelnek.

Megjegyzés [szp28]:

A csillagbeolvasás közepette a **K**-nál közelibbetek éllel kapcsoljuk össze. Célszerű a gráfpont értékeként (abban tárolni) a pont sorszámat, és térbeli koordinátáit.

Megjegyzés [szp29]:

A csillaghalmazok a gráf komponensei.

Megjegyzés [szp30]:

A komponensbeli pontokon végig menve a távolság maximumot kiszámolni.