

Megoldások

A feladatsor létrejöttében közreműködtek: Bakonyi Viktória, Busa Máté, Csertán András, Deák Bence, Gáspár Attila, Horváth Gyula, Németh Zsolt, Noszály Áron, Sárközi Gergely, Sente Péter, Zsakó László

17

Ötlet: Zsakó László

Kidolgozó: Zsakó László

Témák: implementáció (elágazás, ciklus, tömb)

A feladat erősen implementációs jellegű, a megoldáshoz elegendő a feladatszövegben definiált algoritmus lekódolása. Gyakori buktatók voltak a második üres sor hiánya vagy az algoritmus nem pontos megvalósítása (pl. $N > 17$ -ig ismételni $N > 0$ helyett). Érdekeséggéppen meggondolhatjuk, hogy az algoritmus $O(\log(N))$ futási idejű, mivel lépésenként legalább 1-el csökken az N tízes számrendszerbeli hossza.

Zene**Ötlet:** Nikházy László**Kidolgozó:** Deák Bence**Témák:** prefix összegek, bináris keresés

Világos, hogy az $[1, T_1]$ időpontokban az első, $[T_1+1, T_1+T_2]$ -ben a második, $[T_1+T_2+1, T_1+T_2+T_3]$ -ban a harmadik zene szól, és így tovább.

Általánosan: $[T_1 + \dots + T_{i-1} + 1, T_1 + \dots + T_i]$ -re a válasz i .

Állítsuk elő T prefix összegeit: $S_i = T_1 + \dots + T_i$! Ezek szerint az $[S_{i-1} + 1, S_i]$ időpontokban az i -edik zeneszám szól. A feladat szerint a lejátszás periódusa S_N , vagyis bármely $p > S_N$ időpontban ugyanaz a dal szól, mint a $p - S_N$ időpontban.

Először egy adott P_i időponthoz határozzuk meg azt a $1 \leq p \leq S_N$ értéket, amire $P_i \equiv p \pmod{S_N}$! Ekkor a válasz az a legnagyobb j index lesz, amire $p \leq S_j$. Megkereséséhez használhatunk bináris keresést. A bináris keresés komplexitása $O(\log(N))$, tehát a teljes algoritmus aszimptotikus futási ideje $O(N + K \cdot \log(N))$ lesz.

Kérjük a tisztelt kollégákat, hogy a dolgozatokat az egységes értékelés érdekében szigorúan az alábbi útmutató szerint pontozzák, a megadott részpontoszámokat ne bontsák tovább! Vagyis ha egy részmegoldásra pl. 3 pontot javasolunk, akkor arra vagy 0, vagy 3 pont adható. (Természetesen az útmutatótól eltérő megoldások is lehetnek jók.)

1. feladat: Sípálya (25 pont)

Egy hójelentés N sípályán mért hóréteget tartalmazza.

Készíts programot (`sipalya.pas,...`), amely beolvassa a sípályák számát ($1 \leq N \leq 20$) és az egyes pályákon a hóréteg vastagságát ($0 \leq V(i) \leq 100$), majd

A. megadja, hogy melyik sípályán a legnagyobb a hóréteg;

B. megad egy sípályát, ahol a hóréteg legalább 100 cm vastag;

C. megadja azokat a sípályákat, ahol nem lehet síelni (azaz a hóréteg vastagsága 0)!

Példa

Bemenet:

Kimenet:

Pályák száma: 7

Legnagyobb: 3

1. pálya: 0

Van 100 cm hó: 2

2. pálya: 150

Nem lehet síelni 3 pályán: 1 4 6

3. pálya: 200

4. pálya: 0

5. pálya: 57

6. pálya: 0

7. pálya: 30

Értékelés:

Sehol sincs hó (3,0,0,0 → bármi, -, 3 db 1 2 3)

0+0+1 pont

Utolsó pálya a leghavasabb, a második hónélküli (3,10,0,100 → 3, 3, 1 db: 2)

2+2+2 pont

Mindenhol lehet síelni (3,110,120,100 → 2, 1 vagy 2 vagy 3, 0 db)

2+2+2 pont

Általános eset (6,0,110,0,0,120,100 → 5, 2 vagy 5 vagy 6, 3 db: 1 3 4)

2+2+2 pont

Általános eset (6,0,50,0,110,120,50 → 5, 4 vagy 5, 2 db: 1 3)

2+2+2 pont

2. feladat: Pénz (30 pont)

Bergengóciában N-féle pénzermét használnak: P(1), P(2), ..., P(N) forintost.

Írj programot (`penz.pas,...`), amely megadja, hogy mely 1 és M forint közötti összegek fizethetők ki egyetlen pénzermével, melyek legfeljebb 2 pénzermével és melyek legfeljebb 3 pénzermével!

A program olvassa be a pénzermék számát ($1 \leq N \leq 10$), és a kifizetendő összeget ($1 \leq M \leq 1000$)! Ezután olvassa be a pénzermék értékét ($1 \leq P_i \leq M$)!

A program írja ki a legfeljebb 1, 2, majd a 3 pénzermével kifizethető összegeket!

Példa:

Bemenet:

Kimenet:

Érmék száma: 2

1 érmevel: 1 5

Maximális összeg: 100

2 érmevel: 1 2 5 6 10

1. érme: 1

3 érmevel: 1 2 3 5 6 7 10 11 15

2. érme: 5

Értékelés:

N=1, P(1)=1, M=100 → 1, 1 2, 1 2 3	1+2+3 pont
N=2, P(1)=5, P(2)=10, M=100 → 5 10, 5 10 15 20, 5 10 15 20 25 30	1+2+3 pont
N=2, P(1)=2, P(2)=5, M=100 → 2 5, 2 4 5 7 10, 2 4 5 6 7 9 10 12 15	1+2+3 pont
N=2, P(1)=2, P(2)=3, M=100 → 2 3, 2 3 4 5 6, 2 3 4 5 6 7 8 9	1+2+3 pont
N=3, P(1)=2, P(2)=5, P(3)=10, M=100 → 2 5 10, 2 4 5 7 10 12 15 20, 2 4 5 6 7 9 10 12 14 15 17 20 22 25 30	1+2+3 pont

3. feladat: Nyelv (20 pont)

Egy programozási nyelven az elágazások az IF szóval kezdődnek és a FI szóval végződnek. Minden program legalább egy, legfeljebb 100 szóból áll

Készíts programot (`nyelv.pas`, ...), amely beolvas szavanként egy szöveget, majd megadja, hogy az elágazások egymásba ágyazása helyes-e! Ha nem helyes, akkor megadja az első hiba okát is. (Pl. hibás egymásba ágyazás az alábbi: ... IF ... FI ... FI ... IF ...)

Megjegyzés: a többi szó bármi lehet, ellenőrzésükkel nem kell foglalkozni.

Bemenet:

Kimenet:

Szavak száma: 7

Hibás: FI utasítás hiányzik

1. szó: IF

2. szó: ALMA

3. szó: IF

4. szó: FI

5. szó: FI

6. szó: IF

7. szó: BARACK

Értékelés:

Helyes program (nincs IF, FI)	2 pont
Helyes program (egyetlen IF FI pár)	2 pont
Helyes program (több IF FI pár egymás után)	2 pont
Helyes program (több IF FI pár egymás belsejében)	2 pont
FI előtt nincs IF (FI ...)	2 pont
FI előtt nincs IF, de utána van – darabszámuk jó (FI ... IF)	2 pont
FI előtt kevés IF van (IF ... FI ... FI ...)	3 pont
IF-nek nincs FI párja (IF ...)	2 pont
Több IF, mint FI (IF ... IF ... FI)	3 pont

Elérhető összpontszám: 75 pont + 25 pont az 1. fordulóból

Kérjük a tisztelt kollégákat, hogy az egységes értékelés érdekében az alábbi eljárást alkalmazzák:

1. Az értékelő gépen hozzák létre a \NT2 könyvtárat.
2. Másolják be a \NT2 könyvtárba az NT2.EXE állományt, amely jelszóval védett önkibontós ARJ állomány (a tesztadatokat és az értékelő programot tartalmazza), és indítsák el az NT2 .EXE -g<jelszó> paranccsal (a jelszót a <> jelek nélkül kell beírni). A NT2.EXE állományt és a jelszót mindenkihez időben eljuttatjuk.
3. Minden versenyző számára hozzanak létre egy külön könyvtárat, és ezekbe másolják be, majd fordítsák le a versenyzők programjait (a feladatleírásban szereplő néven).
4. Egy versenyző értékelése:
 - A. Az aktuális könyvtár legyen a versenyző könyvtára.
 - B. Adják ki az \NT2\T3 parancsot, amely lefuttatja a versenyző programjait minden tesztesetre. Ha a végrehajtás megszakad, vagy meg kell szakítani, mert letelt a 60 másodperc, akkor ismét a \NT2\T3 parancsot kell kiadni, mindaddig, amíg az "ÉRTÉKELÉS BEFEJEZŐDÖTT" üzenet meg nem jelenik a képernyőn. (A futtató tudja, hogy honnan kell folytatnia.). Ezt követően automatikusan elindul a megoldásokat értékelő program, amely összesítést készít a versenyző könyvtárban EREDMENY.TXT néven, és az eredményt a képernyőre is kiírja.

1. feladat: Falu (18 pont)

Ismerjük egy megye települései (falvak, városok) közötti utak hosszát. Zsákfalunak nevezzük azt a falut, ahova csak egyetlen út vezet (és onnan tovább már nem lehet menni, csak visszafelé). A településeket sorszámmal azonosítjuk.

Készíts programot (falu.pas, ...), amely megadja:

- A. a zsákfalvak számát;
- B. azt a települést, ahova a legtöbb út vezet szomszédos településről;
- C. az egymáshoz legközelebbi 2, nem szomszédos települést (ha több ilyen van, akkor bármelyik megadható)

A falu.be szöveges állomány első sorában a települések ($2 \leq N \leq 1000$) és az utak száma van ($1 \leq M \leq 100000$), egy szóközzel elválasztva. A következő M sor mindegyikében három egész szám van, egy-egy szóközzel elválasztva: egy-egy út két végpontjának sorszáma és a köztük levő út hossza.

A falu.ki szöveges állomány első sorába a zsákfalvak számát; a második sorba a legtöbb utas település sorszáma (ha több van, bármelyik megadható), a harmadikba pedig a két legközelebbi település sorszáma (szóközzel elválasztva, ha több egyforma távolság is van, bármelyik településpár megadható) kell írni!

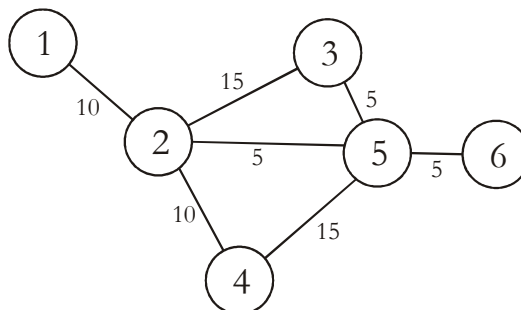
Példa:

falu.be

```
6 7
1 2 10
2 3 15
2 4 10
2 5 5
3 5 5
4 5 15
5 6 5
```

falu.ki

```
2
2
3 6
```


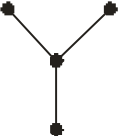
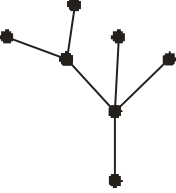
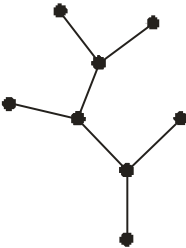


Értékelés:

Két falu	1+1+1 pont
Nincs zsákfal, egy legtöbb utas, egy legközelebbi	1+1+1 pont
Egy zsákfal, több legtöbb utas, minden távolság egyforma	1+1+1 pont
Több zsákfal, a többiek útszáma egyforma, több legközelebbi	1+1+1 pont
Általános eset (közepes teszt)	1+1+1 pont
Általános eset (nagy teszt)	1+1+1 pont

2. feladat: Fa (18 pont)

Minden fát leírhatunk egy karaktorsorozattal. Ebben a leírásban X betűk és zárójelek fognak szerepelni. Az X ágat jelent, az ágak végi elágazásokat pedig zárójelbe tesszük.

			
X	X(X)(X)	X(X(X)(X))(X)(X)	X(X(X)(X(X)(X)))(X)

Írj programot (`fak.pas`, ...), amely megadja:

- A. a fa magasságát (a földtől milyen messze van a legmesszebb levő ágvég);
- B. a fa elágazásai számát (a törzs nem számít elágazásnak);
- C. egy helyen a legnagyobb elágazásszámot!

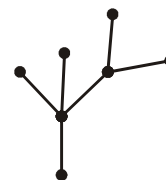
A `fak.be` szöveges állomány egyetlen sorában a fát leíró szöveg van (hossza legfeljebb 10000 karakter).

A `fak.ki` szöveges állomány első sorába a fa magasságát, a második sorába a fa elágazásai számát, a harmadik sorába pedig a legnagyobb elágazásszámot kell kiírni!

Példa:

`fak.be`
X (X) (X) (X (X) (X))

`fak.ki`
3
5
3

Értékelés:

Nincs elágazás	1+1+1 pont
Egyetlen kétfelé ágazás, az egyik ág a hosszabb	1+1+1 pont
Egyetlen sokfelé ágazás	1+1+1 pont
Több elágazás, középső ág a leghosszabb	1+1+1 pont
Több elágazás, szélső ág a leghosszabb (nagy teszt)	1+1+1 pont

3. feladat: Pakolás (18 pont)

Egy raktárban egyetlen hosszú sorban ládák vannak. Minden láda kocka alakú, de méretük különböző lehet. A ládák egymásra rakásával akarnak helyet felszabadítani. A biztonsági előírás szerint több ládát is lehet egymásra rakni, de minden ládát csak nálánál nagyobbra lehet helyezni. Továbbá, az i -edik helyen lévő ládát csak akkor lehet rárakni a j -edik helyen lévő torony tetejére, ha az i -edik és j -edik helyek között már nincs láda (j lehet akár kisebb, akár nagyobb, mint i). Minden ládát legfeljebb egyszer lehet mozgatni.

Készíts programot (`pakol.pas`, ...), amely kiszámítja, hogy legkevesebb hány toronyba lehet a ládákat összepakolni!

A `pakol.be` szöveges állomány első sorában a ládák N ($2 \leq N \leq 30000$) száma van. A második sor pontosan N pozitív egész számot tartalmaz egy-egy szóközzel elválasztva, a ládák méreteit. A második sorban lévő számok mindegyike 1 és 30000 közötti érték.

A `pakol.ki` szöveges állomány első és egyetlen sora egy egész számot tartalmazzon, azt a legkisebb M számot, hogy a bementben megadott ládasor összepakolható M számú toronyba!

Példa:

<code>pakol.be</code>	<code>pakol.ki</code>
10	2
1 2 4 6 7 5 3 2 5 3	

Értékelés:

Növekvő sorozat	2 pont
Csökkenő sorozat	2 pont
Kis méretű ládák	2 pont
Általános eset	2 pont
Véletlen kis teszt	2 pont
Véletlen közepes teszt	2 pont
Véletlen közepes teszt	2 pont
Véletlen nagy teszt	2 pont
Véletlen nagy teszt	2 pont

4. feladat: Játék (21 pont)

Tekintsük azt az egyszemélyes játékot, amelyet N sorból és M oszlopból álló négyzetácsos táblán játszanak. Minden mező vagy üres, vagy csapda. Egy bábút kell mozgatni a táblán. A bábu kezdetben a tábla bal felső sarkában van, és a jobb alsó sarokba kell eljuttatni az alábbi lépés-szabályt betartva:

- Csak olyan mezőre lehet lépni, ahova még nem lépett a bábu.
- Csapda mezőre nem lehet lépni.
- Csak a négy szomszédos mező valamelyikére lehet lépni.
- Egy lépésben csak jobbra, vagy lefelé lehet lépni.

Készíts programot (`jatek.pas`, ...), amely kiszámítja, hogy hányféle képen lehet eljuttatni a bábút a bal felső sarokból a jobb alsóba!

A `jatek.be` szöveges állomány első sorában két egész szám van, a sorok N és oszlopok M száma ($1 \leq N \leq 10$, $1 \leq M \leq 20$). A további N sor mindegyike M egész számot tartalmaz egy-egy szóközzel elválasztva. Minden szám vagy 0, vagy 1. A sorban az i -edik szám 1, akkor a megfelelő mező csapda, egyébként a mező üres.

A `jatek.ki` szöveges állomány egyetlen sora egy egész számot tartalmazzon, azt, hogy hány féleképpen lehet eljuttatni a bábút a bal felső sarokból a jobb alsóba!

Példa:

<code>jatek.be</code>	<code>jatek.ki</code>
5 6	7
0 0 0 0 0 0	
0 1 0 0 1 0	
0 0 1 0 0 0	
1 0 1 0 0 0	
0 0 0 0 1 0	

Értékelés:

A megoldás 0	2 pont
Nincs csapda	2 pont
Egy csapda van	2 pont
Sok csapda van	2 pont

Kicsi véletlen bemenet	2 pont
Közepes véletlen bemenet	2 pont
Közepes véletlen bemenet	2 pont
Nagy véletlen bemenet	2 pont
Nagy véletlen bemenet	2 pont
Nagy véletlen bemenet	3 pont

Elérhető összpontszám: 75 pont + 25 pont az 1. fordulóból

1. feladat: Lövészverseny (25 pont)

Az időjárás előrejelzésben ismerjük előre N ($2 \leq N \leq 100$) nap várható minimális és maximális hőmérsékletét. Készíts programot (`idojaras.pas`, `idojaras.c`, ...), amely beolvassa N értékét és a $2 \cdot N$ db hőmérsékletet, majd megadja:

A. azt a K napos időtartamot (ha van), amelyben az előrejelzés szerint folyamatosan fagy lesz;

B. azt a két szomszédos napot, ahol a legnagyobbat változik a hőmérséklet;

C. azokat a napokat, ahol a napi minimális hőmérséklet a napi átlaghőmérsékletek átlaga fölötti!

Példa:

Bemenet:

Napok száma?5

Fagy hány napon keresztül?2

1. nap minimuma, maximuma: -9 -2

2. nap minimuma, maximuma: -1 4

3. nap minimuma, maximuma: -5 -4

4. nap minimuma, maximuma: -6, -1

5. nap minimuma, maximuma: 5 8

Kimenet:

Folyamatos fagy: 3 4

Legnagyobb változás: 4 5

Átlag fölöttiek: 2 5

Értékelés:

Nincs folyamatos fagy, első 2 nap változik felfelé, nincs átlag fölötti (3,2,-5 5,-5 5,-5 3
→nincs,1 2,nincs) 1+1+1 pont

Folyamatos fagy az elején, utolsó 2 nap változik lefelé, egy átlag fölötti (3,2,-10 -5,-1 -1,
-15 -15 →1 2,2 3,2) 1+1+1 pont

Folyamatos fagy a végén, első 2 nap változik lefelé, több átlag fölötti (3,2,-1 -1,-15 -15,-5 -1
→2 3,1 2,1 3) 1+1+1 pont

Folyamatos fagy közepén, utolsó 2 nap változik felfelé, több átlag fölötti (5,3,-1 8,-5 -5,
-10 -10,-5 -3,5 10 →2 4,4 5,1 5) 1+1+2 pont

Általános eset (6,3,-1 8,-5 -5,-10 -10,-5 -3,5 10,5 10 →2 4,4 5,5 6) 2+2+2 pont

Általános eset (7,3,-1 8,-5 -5,-10 -10,-5 -3,5 10,5 10,-10 -5 →2 4,6 7,1 5 6) 2+2+2 pont

2. feladat: Tördelés (25 pont)

Wikipédián található az alábbi leírás a webcímekekről (most csak azt engedjük meg, ami ebben a leírásban szerepel):

Egy tipikus, egyszerű webcím így néz ki:

<http://hu.wikipedia.org:80/wiki>

Ennek részei:

- A `http` (vagy `https`) a használandó protokoll. A protokoll neve után kettőspont (`:`) írandó.
- A `hu.wikipedia.org` a célgép tartományneve. Ez elé két perjel (`//`) írandó.
- A `80` a célgép azon hálózati portszáma, amin kérésünket várja; ez elé kettőspont (`:`) írandó. Ezt a részt gyakran teljesen elhagyhatjuk, például esetünkben a `http` protokoll alapértelmezett portszáma a `80`.
- A `/wiki` a kért elérési út a célgépen. Ez a rész mindig a perjellel (`/`) kezdődik.

A legtöbb böngésző nem is igényli, hogy a „`http://`” részt begépeljük egy weblap eléréséhez, hiszen az esetek döntő többségében úgyis ezt használjuk. Egyszerűen begépelhetjük a lap címét, például: „`hu.wikipedia.org/wiki/Bit`”. A főlap megtekintéséhez általában elég a tartomány nevét beírni, például „`hu.wikipedia.org`”.

A webcím a példákban szereplőtől eltérő jeleket (pl. szóközt, relációkat, ...) nem tartalmazhat. A webcímek egyéb részeket is tartalmazhatnak, http esetében például az elérési út után, egy kérdőjel (?) mögé helyezve keresési kérdés szerepelhet, ami egy get metódusú HTML űrlapból származik.

Példa:

<http://hu.wikipedia.org/w/wiki.phtml?title=Bit&action=history>

Készíts programot (`url.pas`, `url.c`, ...), amely egy beolvasott webcímet (legfeljebb 100 karakteres) a fenti szempontok szerint ellenőriz! Hay valamely szempont szerint hibás, akkor megadja, hogy m a hiba a szövegben.

Példa:

Bemenet: `http://www.njszt.hu\tehetseg`

Kimenet: Hibás karakter: \

Értékelés:

Jó tesztek

http://valami.hu	1 pont
https://valami.hu/valami	1 pont
valami.hu	1 pont
http://valami.hu:80/valami	2 pont
https://valami.hu/valami?a=b	2 pont
http://valami.hu/valami?a=&c=b	2 pont

Hibás tesztek

http után nincs : (<code>http/valami.hu/</code>)	2 pont
http: után nincs // (<code>http:valami.hu</code>)	2 pont
nincs http, de van : vagy / (<code>://valami.hu</code>)	2 pont
csak egy / van a http: után (<code>https:/valami.hu</code>)	2 pont
két / között nincs semmi (<code>http://valami.hu//</code>)	2 pont
hibás karakter (<code>http:valami.hu\valami</code>)	2 pont
a : nem a célgép tartományneve mögött van (<code>https://valami.hu/valami:80</code>)	2 pont
a : után nem port azonosító szám jön (<code>http://valami.hu:valami</code>)	2 pont

3. feladat: Elszigetelt falu (25 pont)

Egy megyében N ($1 \leq N \leq 100$) falu van. A falvakat M ($3 \leq M \leq 1000$) út köti össze, ismerjük minden út hosszát. A legelszigeteltebb falunak azt nevezzük, amelytől a legközelebbi szomszédja a lehető legtávolabb van.

Készíts programot (`falu.pas`, `falu.c`, ...), amely megadja a legelszigeteltebb falut! Ha több megoldás van, akkor bármelyik kiírható.

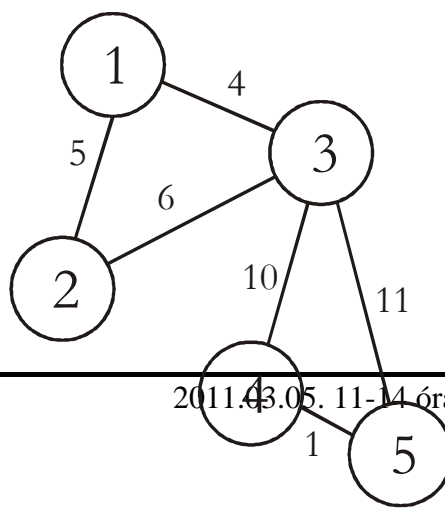
Példa:

Bemenet:

Falvak száma?5

Utak száma?6

1. út kezdete, vége, hossza?1 2 5
2. út kezdete, vége, hossza?2 3 6



3. út kezdete, vége, hossza? 3 1 4
4. út kezdete, vége, hossza? 3 4 10
5. út kezdete, vége, hossza? 4 5 1
6. út kezdete, vége, hossza? 3 5 11

Kimenet:

A legelszigeteltebb: 2

Értékelés:

Két falu (2,1,1 2 5→1 vagy 2)	1 pont
Van falu, ahova nem vezet út (3,1,1 2 5→3)	2 pont
Minden távolság egyforma (3,3,1 2 5,2 3 5,3 1 5→1 vagy 2 vagy 3)	3 pont
Három falu, egyértelmű megoldás (3,3,1 2 5,2 3 5,3 1 4→2)	3 pont
Négy falu, két lehetséges megoldás (4,4,1 2 1, 2 3 4, 3 4 2, 4 1 4→3 vagy 4)	4 pont
Négy falu, egyértelmű megoldás (4,4,1 2 1,2 3 4, 3 4 4, 4 1 1→3)	4 pont
Négy falu, egyértelmű megoldás (4,5,1 2 1,2 3 4, 3 4 4, 4 1 1,2 4 1→3)	4 pont
A példa permutálva (5,6,2 3 5,3 4 6,4 2 4,4 5 10,5 1 1,4 1 11→3)	4 pont

Elérhető összpontszám: 75 pont + 25 pont a 2. fordulóból

Kérjük a tisztelt kollégákat, hogy a dolgozatokat az egységes értékelés érdekében szigorúan az alábbi útmutató szerint pontozzák, a megadott részpontszámokat ne bontsák tovább! Vagyis ha egy részmegoldásra pl. 3 pontot javasolunk, akkor arra vagy 0, vagy 3 pont adható. (Természetesen az útmutatótól eltérő megoldások is lehetnek jók.)

1. feladat: Kártya (25 pont)

Egy kártyajátékban az egyes lapoknak számértékük van. Minden lapot egy színnel és egy figurával adunk meg. A színek: piros, zöld, tök, makk. A figurák: 7-es, 8-as, 9-es, 10-es, alsó, felső, király, ász. A számot tartalmazó figurák annyit érnek, amennyi a ráírt szám. Az alsó 2-t, a felső 3-at, a király 4-et, az ász 11-et ér. A piros lapoknál az értéket duplán kell számítani.

Készíts programot (`kartya.pas`,...), amely beolvassa egy játékos N ($1 \leq N \leq 4$) kártyáját, majd megadja, hogy a lapok összesen hány pontot érnek!

Példa

Bemenet:

Kimenet:

Kártyák száma? 3

A kártyák értéke: 22

1. kártya színe? piros

1. kártya figurája? alsó

2. kártya színe? tök

2. kártya figurája? 7-es

3. kártya színe? tök

3. kártya figurája? ász

Értékelés:

Egyetlen lap: tök alsó $\rightarrow 2$

2 pont

Egyetlen lap: makk 8-as $\rightarrow 8$

2 pont

Egyetlen lap: zöld király $\rightarrow 4$

3 pont

Egyetlen lap: piros 10-es $\rightarrow 20$

3 pont

Két lap: tök felső, makk király $\rightarrow 7$

3 pont

Két lap: zöld 7-es, zöld 10-es $\rightarrow 17$

3 pont

Két lap: piros 7-es, piros ász $\rightarrow 36$

3 pont

Két lap: makk 7-es, piros ász $\rightarrow 29$

3 pont

Négy lap: piros ász $\rightarrow 88$

3 pont

2. feladat: Bábu (25 pont)

Egy játéktáblán a 0. időegységben L bábu van. Mindegyiket elindítjuk valamerre. Egy időegység alatt mindegyik a neki megfelelő távolságra mozdul el, a tábla széléről visszafordulnak. Lehetséges, hogy előbb-utóbb két bábu összeütközik: ugyanarra a helyre lépnének vagy átlépnének egymáson.

Írj programot (`babu.pas`, ...), amely megadja, hogy K időegységen belül mikor ütközik legelőször két bábu!

A program olvassa be a tábla szélességét ($1 \leq N \leq 100$), a bábuk számát ($1 \leq L \leq 10$) és az időtartamot ($1 \leq K \leq 100\,000$)! Ezután olvassa be a bábuk kezdő helyét ($1 \leq S_i \leq N$) és mozgás irányát ($X_i \in \{J, B\}$ – jobbra, balra)!

A program írja ki az első ütközés időpontját! Ha K időegységen belül nincs ütközés, akkor -1-et kell kiírni!

Példa:

A tábla hossza: 10

A bábuk száma: 2

**Az időtartam: 10**

1. bábu helye: 3, iránya: B

2. bábu helye: 8, iránya: B

Ütközés időpont: 5

Magyarázat: A bábuk helyzete időegységenként:

1: 2, 7

2: 1, 6

3: 2, 5

4: 3, 4

5: 4, 3

Az 5. időegységre az 1-es és a 2-es bábu egymáson átlépett volna, azaz az 5. időegységben ütköztek.

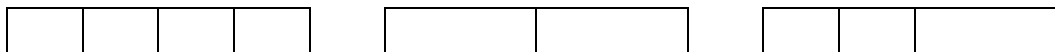
Értékelés:

Egyetlen bábu (10,1,10,1 J \Rightarrow NINCS) 1 pontKét bábu, szembe mennek, nem érnek el egymáshoz (10,2,1,1 J,10 B \Rightarrow NINCS) 2 pontKét bábu, szembe mennek, egy helyre lépnének (10,2,10,1 J 5 B \Rightarrow 2) 2 pontKét bábu, egy irányba mennek, egy helyre lépnének (10,2,10,3 B,5 B \Rightarrow 3) 2 pontKét bábu, ellenkező irányba mennek, egy helyre lépnének (10,2,20,2 B,8 J \Rightarrow 6) 3 pontKét bábu, ellenkező irányba mennek, átlépnének egymáson (10,2,20,2 B,9 J \Rightarrow 6) 3 pontKét bábu, szembe mennek, átlépnének egymáson (10,2,10,1 J 6 B \Rightarrow 3) 3 pont4 bábu, 2 ütközés (10,4,10,1 B,7 J,8 J,10 J \Rightarrow 1) 3 pontVéletlen közepes teszt (100,3,100,32 B,45 J,87 B \Rightarrow 21) 3 pontVéletlen nagy teszt (100,5,100,12 J,98 B,45 J,44 B,92 B \Rightarrow 16) 3 pont3. feladat: Járdakövezés (25 pont)

Egy N ($1 \leq N \leq 80$) egység hosszú járdát 1 és 2 méretű lapokkal szeretnénk kikövezni. Hányféleképpen lehet ezt megtenni?

Készíts programot (lapok.pas, ...), amely kiszámítja, hogy egy N egység hosszú járdát hányféleképpen lehet kikövezni 1 és 2 méretű lapokkal!

Példa: N=4 egység hosszú járdát 5-féleképpen lehet kikövezni, a kikövezési lehetőségei:



Bemenet:

Kimenet:

A járda hossza? 4

A kikövezések száma: 5

Értékelés:

N=1 \rightarrow 1 2 pontN=2 \rightarrow 2 2 pont

$N=3 \rightarrow 3$	2 pont
$N=5 \rightarrow 8$	3 pont
$N=6 \rightarrow 13$	3 pont
$N=10 \rightarrow 89$	3 pont
$N=20 \rightarrow 10946$	3 pont
$N=50 \rightarrow 20365011074$	3 pont
$N=80 \rightarrow 37889062373143906$	4 pont

Elérhető összpontszám: 75 pont + 25 pont az 1. fordulóból2

Kérjük a tisztelt kollégákat, hogy az egységes értékelés érdekében az alábbi eljárást alkalmazzák:

1. Az értékelő gépen hozzák létre a \NT2 könyvtárat.
2. Másolják be a \NT2 könyvtárba az NT2.EXE állományt, amely jelszóval védett önkibontós ARJ állomány (a tesztadatokat és az értékelő programot tartalmazza), és indítsák el az NT2.EXE -g<jelszó> paranccsal (a jelszót a <> jelek nélkül kell beírni). A NT2.EXE állományt és a jelszót mindenkihez időben eljuttatjuk.
3. Minden versenyző számára hozzanak létre egy külön könyvtárat, és ezekbe másolják be, majd fordítsák le a versenyzők programjait (a feladatleírásban szereplő néven).
4. Egy versenyző értékelése:
 - A. Az aktuális könyvtár legyen a versenyző könyvtára.
 - B. Adják ki a \NT2\T2 parancsot, amely lefuttatja a versenyző programjait minden tesztetere. Ha a végrehajtás megszakad, vagy meg kell szakítani, mert letelt a 60 másodperc, akkor ismét a T2 parancsot kell kiadni, mindaddig, amíg az "ÉRTÉKELÉS BEFEJEZŐDÖTT" üzenet meg nem jelenik a képernyőn. (A futtató tudja, hogy honnan kell folytatnia.). Ezt követően automatikusan elindul a megoldásokat értékelő program, amely összesítést készít a versenyző könyvtárában EREDMENY.TXT néven, és az eredményt a képernyőre is kiírja.

1. feladat: Kép (20 pont)

Ugyanarról a területről két időpontban készítettünk fényképet. A fényképek négy széléről le szeretnénk vágni azt a részt, amelyek egyformák.

Készíts programot (kep.pas, ...), amely megadja hogy a kép 4 széléről maximum mekkora téglalapok vághatók le!

A kep.be szöveges állomány első sorában a fényképek sorai és oszlopai száma van ($1 \leq N, M \leq 1000$), egy szóközzel elválasztva. A következő N sorban az első kép, az azt követő N sorban a második kép képpontjai vannak. Minden sor M képpont leírását tartalmazza, egymástól egy-egy szóközzel elválasztva. A képpontokat egy 0 és 255 közötti fényességértékkel adjuk meg.

A kep.ki szöveges állomány első sorába a legnagyobb balról, alulról, jobbról, illetve felülről levágható téglalap szélességét kell írni!

Példa:

kep.be

```
8 10
1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 3 3 3 3 3
2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 5 5 5 5
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0
-----
1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 3 3 3 3 3
2 2 2 2 2 2 2 2 2 2
2 2 2 2 2 2 5 5 5 5
1 1 1 1 1 1 1 1 1 1
1 3 1 1 3 1 1 1 1 1
1 1 1 1 1 1 5 1 1 1
0 0 0 0 0 0 0 0 0 0
```

kep.ki

```
1 1 3 2
```

Értékelés:

Egyetlen pont	1 pont
Egy sorban több pont	2 pont
Egy oszlopban több pont	2 pont
Téglalap bal felső és jobb alsó sarka	3 pont

Téglalap bal alsó és jobb felső sarka	3 pont
Téglalap belsejében is vannak különbségek	3 pont
Téglalap szélei, de nem a sarka	3 pont
Véletlen nagy teszt	3 pont

2. feladat: Játék (20 pont)

Egy játéktáblán a 0. időegységben L bábu van. Mindegyiket elindítjuk valamerre. Egy időegység alatt mindegyik a neki megfelelő távolságra mozdul el, a tábla szélére érve megállnak. Lehetséges, hogy előbb-utóbb két bábu összeütközik: ugyanarra a helyre lépnének vagy átlépnének egymáson.

Írj programot (`tabla.pas`, ...), amely megadja, hogy K időegységen belül mikor ütközik legelőször két bábu!

A `tabla.be` szöveges állomány első sorában a játéktábla sorai és oszlopai száma ($1 \leq N, M \leq 100$), a bábuk száma ($1 \leq L \leq 10$) és az időtartam ($1 \leq K \leq 100\,000$) van, egyetlen szóközzel elválasztva. A következő L sor egy-egy bábu leírását tartalmazza: a kezdő helyét ($1 \leq S_i \leq N$, $1 \leq O_i \leq M$) és mozgás irányát ($X_i \in \{F, L, J, B\}$ – fel, le, jobbra, balra), egy-egy szóközzel elválasztva.

A `tabla.ki` szöveges állomány egyetlen sorába az első ütközés időpontját kell írni! Ha K időegységen belül nincs ütközés, akkor -1-et kell kiírni!

Példa:

<code>tabla.be</code>	<code>tabla.ki</code>
7 10 3 100	3
4 3 J	
2 6 F	
4 8 B	

Magyarázat: A bábuk helyzete időegységenként:

1: (4, 4), (1, 6), (4, 7)
 2: (4, 5), (1, 6), (4, 6)
 3: (4, 6), (1, 6), (4, 5)

A 3. időegységre az 1-es és a 3-as bábu egymáson átlépett volna, azaz a 3. időegységben ütköztek.

Értékelés:

Egyetlen bábu	1 pont
Két bábu, nincs ütközés	1 pont
Két bábu, szembe mennek, egy helyre lépnének	2 pont
Két bábu, szembe mennek, átlépnének egymáson	2 pont
Két bábu, egyik ellép a másik elől	2 pont
Két bábu, egyik a szélén áll, mire a másikkal ütközik	3 pont
4 bábu, 2 ütközés	3 pont
Véletlen közepes teszt	3 pont
Véletlen nagy teszt	3 pont

3. feladat: Ütemezés (20 pont)

Egy vállalkozó alkatrészek gyártásával foglalkozik. Minden alkatrészen kétféle műveletet kell elvégeznie, **A** és **B** műveletet. Mindkét művelet elvégzésére egy-egy munkagépe van, amelyek egymástól függetlenül tudnak dolgozni, de egy alkatrészen egyszerre csak egyik mű-

velet végezhető. Az alkatrészen a két műveletet tetszőleges sorrendben el lehet végezni. Minden legyártandó alkatrésze ismert, hogy mennyi időt igényel az **A**, valamint a **B** művelet elvégzése.

Készíts programot (`utemez.pas`, ...), amely kiszámítja, hogy legkevesebb mennyi idő alatt lehet legyártani az összes alkatrészt!

A `utemez.be` szöveges állomány első sorában az alkatrészek N ($2 \leq N \leq 1000$) száma van. Az alkatrészeket az $1, \dots, N$ számokkal azonosítjuk. A második és a harmadik sor pontosan N egész számot tartalmaz egy-egy szóközzel elválasztva, a legyártandó alkatrészek elvégzendő **A**, illetve **B** műveletek idejét. A második sorban az i -edik szám az i -edik alkatrészen végzendő **A** művelet ideje. A harmadik sorban az i -edik szám pedig az i -edik alkatrészen végzendő **B** művelet ideje. A második és harmadik sorban lévő számok mindegyike **1** és **5000** közötti érték.

A `utemez.ki` szöveges állomány első sora egy egész számot tartalmazzon, azt a legkisebb **T** időt, amely alatt a két gép le tudja gyártani az összes alkatrészt!

Példa:

<code>utemez.be</code>	<code>utemez.ki</code>
3	50
4 20 15	
11 30 3	

Értékelés:

A két összeg egyenlő	1 pont
Kis méret	1 pont
Közepes méret, egyszerű eset	2 pont
Közepes méret, nem egyszerű eset	2 pont
Közepes méret, véletlen bemenet	2 pont
Nagy méret, egyszerű eset	3 pont
Nagy méret, nem egyszerű eset	3 pont
Véletlen közepes teszt, egyszerű eset	3 pont
Véletlen nagy teszt, nem egyszerű eset	3 pont

4. feladat: Járdakövezés (15 pont)

Egy N egység hosszú járdát 1, 2 és 3 méretű lapokkal szeretnénk kikövezni. Hányféleképpen lehet ezt megtenni?

Készíts programot (`kovezes.pas`, ...), amely kiszámítja, hogy egy N egység hosszú járdát hányféleképpen lehet kikövezni 1, 2 és 3 méretű lapokkal!

A `kovezes.be` szöveges állomány egyetlen sorában a járda hossza ($1 \leq N \leq 70$) van.

A `kovezes.ki` szöveges állomány egyetlen sora a lehetséges kikövezések számát tartalmazza!

Példa:

<code>kovezes.be</code>	<code>kovezes.ki</code>
4	7
<div style="display: inline-block; border: 1px solid black; width: 100px; height: 20px; margin-right: 5px;"></div> <div style="display: inline-block; border: 1px solid black; width: 100px; height: 20px; margin-right: 5px;"></div>	<div style="display: inline-block; border: 1px solid black; width: 100px; height: 20px; margin-right: 5px;"></div> <div style="display: inline-block; border: 1px solid black; width: 100px; height: 20px; margin-right: 5px;"></div>
<div style="display: inline-block; border: 1px solid black; width: 100px; height: 20px; margin-right: 5px;"></div> <div style="display: inline-block; border: 1px solid black; width: 100px; height: 20px; margin-right: 5px;"></div>	<div style="display: inline-block; border: 1px solid black; width: 100px; height: 20px; margin-right: 5px;"></div> <div style="display: inline-block; border: 1px solid black; width: 100px; height: 20px; margin-right: 5px;"></div>

--	--	--	--

Értékelés:

N=1 → 1	1 pont
N=2 → 2	1 pont
N=3 → 4	1 pont
N=5 → 13	2 pont
N=6 → 24	2 pont
N=10 → 274	2 pont
N=20 → 121415	2 pont
N=50 → 10562230626642	2 pont
N=70 → 2073693258389777176	2 pont

Elérhető összpontszám: 75 pont + 25 pont az 1. fordulóból