

1. feladat: Torony (50 pont)

Háromféle elemünk (építőköcánk) van, mindegyikből tetszőleges számú. A piros és a zöld elemek magassága egy, a fehéré három. Jelölje $T[i]$ azt, hogy hány különböző i magasságú torony építhető!

A mintán alul egy zöld, felül egy piros kocka van, középen pedig egy fehér tégl.

A. Számítsd ki, hogy hány különböző i magasságú torony épülhet ezen elemekből ($T[i]$)!

i	1	2	3	4	5
$T[i]$	2				

Az 1 magasságú torony vagy egy piros kockából áll, vagy egy zöld kockából. A 3 magasságú torony állhat egyetlen fehér téglából, ...

B. Adj képletet a $T[i]$ kiszámítására!

2. feladat: Nyelv (55 pont)

Nyelvek (programozási nyelvek) formális szabályainak leírására régóta használják a BNF leírást (Backus-Naur Form). A működését egy példán keresztül mutatjuk be:

$\langle \text{mondat} \rangle ::= \langle \text{alany} \rangle \langle \text{állítmány} \rangle | \langle \text{állítmány} \rangle \langle \text{alany} \rangle$

$\langle \text{állítmány} \rangle ::= \text{ugat} | \text{nyávog}$

$\langle \text{alany} \rangle ::= a \langle \text{főnév} \rangle | \langle \text{főnév} \rangle$

$\langle \text{főnév} \rangle ::= \text{kutya} | \text{macska}$

A $::=$ jel a definiálás jele. Tőle balra a definiálandó fogalom szerepel \langle és \rangle jelek között. Jobbra a definíciót találjuk. A definícióban az újabb definiálandó fogalmakat szintén \langle és \rangle jelek közé tesszük, a konstans értékeknél nincsenek ilyen jelek. Ha valami többféle is lehet, akkor közéjük $|$ jelet teszünk, például az állítmány vagy az ugat, vagy a nyávog szó. A fenti szabályok szerint helyes mondatok:

ugat a kutya

macska nyávog

nyávog a kutya

Az utolsó formailag helyes akkor is, ha tartalmilag hibás. Formailag hibás azonban a következő:

a nyávog macska

Módosítjuk a szabályokat

$\langle \text{mondat} \rangle ::= \langle \text{alany} \rangle \langle \text{állítmány} \rangle | a \langle \text{alany} \rangle \langle \text{állítmány} \rangle |$
 $\langle \text{állítmány} \rangle \langle \text{alany} \rangle | a \langle \text{alany} \rangle \langle \text{jelző} \rangle |$
 $\langle \text{jelző} \rangle a \langle \text{alany} \rangle$

$\langle \text{állítmány} \rangle ::= \langle \text{határozó} \rangle \langle \text{ige} \rangle | \langle \text{ige} \rangle \langle \text{határozó} \rangle | \langle \text{ige} \rangle$

$\langle \text{alany} \rangle ::= \langle \text{jelző} \rangle \langle \text{főnév} \rangle | \langle \text{főnév} \rangle$

$\langle \text{ige} \rangle ::= \text{ugat} | \text{nyávog}$

$\langle \text{határozó} \rangle ::= \text{hangosan} | \text{szépen}$

$\langle \text{jelző} \rangle ::= \text{fekete} | \text{mérges}$

$\langle \text{főnév} \rangle ::= \text{kutya} | \text{macska}$

Az alábbiak közül mely mondatok felelnek meg a fenti szabályoknak? Amelyik nem, az miért nem felel meg? Amelyik igen, az a <mondat>, <állítmány>, <alany> szabályok milyen alkalmazásával helyes?

- A. a kutya mérges
- B. a fekete kutya hangosan ugat
- C. mérges kutya
- D. nyávog a fekete macska
- E. a mérges kutya nyávog szépen
- F. szépen ugat a kutya
- G. a kutya ugat hangosan

3. feladat: Jól gondolta? (40 pont)

Bendegúz egy terület legmagasabb pontjának megtalálására a következő algoritmust gondolta ki:

1. Választ egy pontot véletlenszerűen
2. Amíg olyan pontot néz, aminél a 4 szomszédja valamelyike magasabb, addig veszi a pont 4 szomszédja közül a legmagasabbat és oda lép – ha több ilyen is van, akkor véletlenszerűen választ közülük.

Az alábbi területen a 2. sor 1. oszlopából indulva a jelölt útvonalon halad ((2,1), (2,2), (3,2), (4,2), (4,3)), azaz megtalálja a legmagasabb pontot.

0	1	2	3	1	2
5	6	3	4	2	4
4	7	8	5	3	5
5	8	9	8	7	6
2	7	4	5	6	1

Bendegúz elvi algoritmusa erre az 5*6-os területre tetszőleges kezdőpont esetén működik.

A. Fogalmazd meg, hogy milyen jellegű területekre nem működik ez az algoritmus!

Bendegúz a példán felbátorodva megírta erre a feladatra a programot, ami az (a,b) pontból indul ki a T mátrixban levő N*M-es területen (sorait 1-től N-ig, oszlopait 1-től M-ig indexeljük):

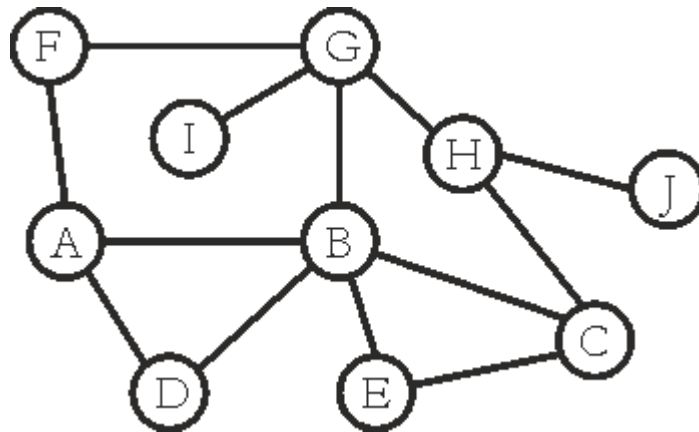
```
vége:=hamis
Ciklus amíg nem vége
  i:=a; j:=b
  Ciklus k=a-1-től a+1-ig
    Ciklus l=b-1-től b+1-ig
      Ha T[k,l]≥T[i,j] akkor i:=k; j:=l
    Ciklus vége
  Ciklus vége
  Ha i=a és j=b akkor vége:=igaz
  a:=i; b:=j
Ciklus vége
```

B. Mit csinál másképpen a fenti programrészlet, mint az elvi algoritmus?

C. Milyen további hiba van a fenti algoritmusban?

4. feladat: Városok (85 pont)

Egy úthálózat köti össze az ábrán szereplő városokat.



A városokat az alábbi stratégia szerint járjuk be:

1. Kezdetben egy várost (pl. az A jelűt) szürkére színezzük, a többit pedig fehérre.
2. Amíg van szürke város, addig vesszük a legújabban szürkére festett várost:
 - Ha nincs fehér szomszédja, akkor feketére festjük.
 - Ha van fehér szomszédja, akkor az abc-sorrendben legelsőt szürkére festjük.

A. Add meg, hogy az egyes városok milyen sorrendben lesznek szürkék, ha a kezdő (azaz az elsőként szürkére festett) város az A, B, C, F!

Egy város távolsága a kezdő várostól azon szakaszok száma, amelyeken keresztül eljutottunk hozzá.

B. Milyen távolságra van a legmesszebb levő város a kezdővárostól és melyek ezek, ha a kezdő város az D, E, H, J?

C. Mely városra a legkisebb a tőle legmesszebb levő város távolsága és mennyi ez a távolság?

5. feladat: Ádám és Éva (60 pont)

Ádám és Éva megadta, hogy egy napon belül mely időszakokban érnek rá. Jelölések:

- adb – Ádám ennyi időszakban ér rá;
- $\text{Ádám}[i].\text{kezd}, \text{Ádám}[i].\text{vég}$ – Ádám i . időszakának kezdete és vége;
- edb – Éva ennyi időszakban ér rá;
- $\text{Éva}[i].\text{kezd}, \text{Éva}[i].\text{vég}$ – Éva i . időszakának kezdete és vége;

Ha például Ádámnak egy szabad időszaka van 12-től 15-ig, Évának kettő szabad időszaka van 8-tól 10-ig és 14-től 20-ig, akkor az így ábrázolható rajzon (Ádám időszakai felül, Éváié alatta):



Az alábbi algoritmus ezek alapján megad „valamit” a kdb változóban és a K tömbben:

```

i:=1; j:=1; kdb:=0
Ciklus amíg i≤adb és j≤edb
  Elágazás
  ①  Ádám[i].vég<Éva[j].kezd esetén i:=i+1
  ②  Éva[j].vég<Ádám[i].kezd esetén j:=j+1
      egyéb esetben kdb:=kdb+1
      K[kdb].kezd:=max(Ádám[i].kezd,Éva[j].kezd)
      K[kdb].vég:=min(Ádám[i].vég,Éva[j].vég)
      Elágazás
  ③  Ádám[i].vég<Éva[j].vég esetén i:=i+1
  ④  Ádám[i].vég>Éva[j].vég esetén j:=j+1
  ⑤  egyéb esetben i:=i+1; j:=j+1
      Elágazás vége
  Elágazás vége
Ciklus vége

```

A. Mi kerül a kdb és a K változóba, ha $adb=1$, $\text{Ádám} = ((8, 10))^1$, $edb=1$, $\text{Éva} = ((9, 11))$?

B. Mi kerül a kdb és a K változóba, ha $adb=3$, $\text{Ádám} = ((8, 10), (12, 14), (18, 19))$, $edb=3$, $\text{Éva} = ((9, 13), (15, 17), (19, 20))$?

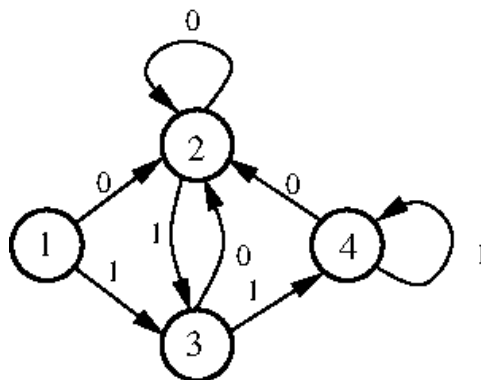
C. Rajzold le a fenti ábrának megfelelő módon, hogy Ádám és Éva időszakai milyen elrendezésében hajtja végre az algoritmus az ①,②,③,④,⑤ sorszámokkal jelzett feltételekhez tartozó értékadásokat!

D. Fogalmazd meg általánosan, mi kerül a kdb és a K változóba az algoritmus végrehajtása után!

6. feladat: Automata (50 pont)

Tekintsük azt az automatát, amelynek négy állapota van, az 1,2,3,4 számokkal azonosítva. Az automata a 0 és 1 bemeneti jeleket tudja fogadni. Ha az automata az A állapotban van és az x bemeneti jelet kapja, akkor az egyértelműen meghatározott $S(A, x)$ állapotba megy át. Az automatánk S átmenetfüggvényét az alábbi táblázat adja meg.

állapot	1	2	3	4
bemenet =0	2	2	2	2
bemenet =1	3	3	4	4



Például, ha az automata a 2. állapotban van és a 01011 bemeneti jelsorozatot kapja, akkor a 4. állapotba megy át.

¹ $\text{Ádám} = ((8, 10))$ jelentése: $\text{Ádám}[1].\text{kezd}=8$, $\text{Ádám}[1].\text{vég}=10$.

Adj meg egy-egy bemeneti 0-1 jelsorozatot, amelyek hatására az automata az i . állapotból a j . állapotba megy át! Fogalmazd meg általánosan is (figyelem: egyes állapotpárookra többféle eset lehet)! Például $i=3$ és $j=2$ esetén minden olyan jelsorozat, aminek a végén 0 van.

- A. $i=1, j=2$
- B. $i=1, j=3$
- C. $i=2, j=3$
- D. $i=3, j=3$
- E. $i=3, j=4$
- F. $i=4, j=4$

7. feladat: Operátorok (60 pont)

Programozási nyelvek leírásakor a kifejezésekben szerepelő műveleteket operátoroknak, azokat a dolgokat pedig, amiken a műveleteket el lehet végezni, operandusoknak nevezzük. Ahhoz, hogy összetett kifejezéseket ki tudjunk értékelni, szükség van az operátorok tulajdonságaira. Az alábbi táblázat egy példát mutat tulajdonságok megadására, a hagyományos matematikai műveletekkel.

Precedencia	Operátor	Asszociativitás
1	$-a$	
2	$a*b$	balról jobbra
	a/b	
3	$a+b$	balról jobbra
	$a-b$	

A táblázat első oszlopa az operátor precedenciáját adja meg, minél magasabb egy operátor precedenciája (kisebb számérték), annál hamarabb kell elvégeznünk a műveletet. Például a $3+a*5$ zárójelek segítségével $3+(a*5)$, ugyanis a szorzásnak magasabb a precedenciája, mint az összeadásnak. A harmadik oszlop azt adja meg két operandusú műveletek esetén, hogy azonos precedenciájú műveleteket balról jobbra vagy jobbról balra kell elvégeznünk. Például a $3/5*4$ kifejezést $(3/5)*4$ -ként kell elvégeznünk, nem pedig $3/(5*4)$ -ként, hiszen balról jobbra szerepel a táblázatban.

Egy ismeretlen programozási nyelvben az operátorokat nagy- az operandusokat kisbetűkkel jelölik, a tulajdonságaik az alábbi táblázatban szerepelnek.

Precedencia	Operátor	Asszociativitás
1	$P a$	
2	$a Q b$	balról jobbra
	$a R b$	
3	$a S b$	jobbról balra
	$a V b$	
4	$a W b$	balról jobbra
5	$X a$	
6	$a Y b$	jobbról balra
	$a Z b$	

Egy kifejezés teljes zárójelezésében minden operandus vagy egy kisbetű, vagy egy zárójeles kifejezés, de a kifejezés jelentése nem változik. Például a $x+a*(b/c)+d$ teljes zárójelezése: $(x+(a*(b/c)))+d$

A. Add meg a következő kifejezések teljes zárójelezését az ismeretlen programozási nyelvben!

A1. a Q b W c

A2. a R P b Y c

A3. a S P b V c

A4. a S (b W c) V d

A5. (a Y b V c) S P d Q e

B. A következő kifejezésekben szerepelhetnek felesleges zárójelpárok. Hagyd el ezeket!

B1. (a W (b R c)) Q d

B2. (a Y b) Z ((c W d) S (P e))

B3. ((a R b) Q (c Z d)) Y ((e V f) S (g W h))

B4. ((a Y b) S (P c R d)) Q (e Z P (f W g Y h))