



## A 2016/2017 tanévi Országos Középiskolai Tanulmányi Verseny első fordulójának feladatai

### INFORMATIKA II. (programozás) kategória

#### 1. feladat: Vállalkozás (27 pont)

Egy vállalkozás 1 napos munkákra kapott megbízást, 1 nap 1 munkát tud elvégezni. Ismerjük minden megbízás határidejét és az elvégzett munkáért kapott fizetséget.

Add meg, hogy az alábbi megrendelések (a számpárok első tagja a határidő, a második a fizetés) esetén maximum mennyit kereshet a vállalkozás és ehhez mely munkákat, mely napokon kell elvégeznie.

A. 1:(3,60), 2:(4,40), 3:(1,10), 4:(3,30), 5:(7,70), 6:(4,20)

B. 1:(4,400), 2:(8,200), 3:(1,200), 4:(4,700), 5:(9,300), 6:(6,700), 7:(7,900), 8:(7,400), 9:(10,100), 10:(2,500)

C. 1:(1,30), 2:(2,420), 3:(2,450), 4:(4,650), 5:(5,930), 6:(1,100), 7:(3,860), 8:(3,50), 9:(4,300), 10:(2,930)

#### 2. feladat: Sorrend (30 pont)

Egy könyvespolcon egy könyvsorozat N tagja helyezkedik el, tetszőleges sorrendben. A sorbarendezés egy lépésében egy tetszőleges könyvet levehetünk a könyvespolcra és egy másik helyre visszatehetjük. (Nem számít külön lépésnek, hogy emiatt a könyv régi és új helye közötti könyveket egy hellyel eltoljuk, hogy a könyvek megint egymás mellett legyenek.)

Add meg, hogy az alábbi könyvsorozatok minimum hány lépésben rendezhetők növekvő sorrendbe és mely könyveket kell ehhez levenni a polcra! Több megoldás esetén bármelyik megadható.

A. 8 könyv, sorrendjük: 1,5,2,6,3,8,4,7

B. 10 könyv, sorrendjük: 8,2,5,4,1,6,3,9,7,10

C. 12 könyv, sorrendjük: 11,1,10,4,3,2,8,7,12,6,9,5

D. 20 könyv, sorrendjük: 2,17,5,9,6,16,8,3,4,10,1,12,7,18,13,20,14,15,11,19

#### 3. feladat: Rendezés (33 pont)

Egy síterepen az egyes csomópontok között lehetnek síelési útvonalak. Add meg az alábbi útvonalak alapján a csomópontok egy olyan sorrendjét, amiben minden csomópont megelőz minden olyan csomópontot, ahova síelési útvonalon el lehet jutni belőle! (Több megoldás esetén bármelyiket.) Add meg azt is, hogy melyik csomópontba hányféleképpen lehet eljutni olyan csomópontból, ahova nem vezet útvonal! (Amely csomópontba nem vezet útvonal, ott ez a szám 0.)

A. (5,2), (1,3), (1,4), (2,3)

B. (9,5), (6,5), (10,6), (10,9), (3,2), (3,10), (8,3), (1,8), (7,8), (4,1), (1,10)

4. feladat: Titkosítás (40 pont)

Egy titkosítási eljárás úgy működik, hogy összekeverik a kapott számok kettes számrendszerbeli biteit. A keverést egy vektor írja le, amely  $i$ . elemében megadjuk, hogy a titkos kód  $i$ . bitjét a szám hányadik bitjéből kell venni (balról 0-tól sorszámozva a biteket). Például, ha a keverést a  $(2,1,3,0)$  vektor írja le, akkor a 0000, 1100, 0111 számok titkos kódjai a 0000, 0101, 1110 számok lesznek.

Add meg a keverést leíró vektort, ha a számok és a kódjuk az alábbi (a számok és a kódok sorrendje nem feltétlenül azonos, több megoldás esetén bármelyik megoldható):

A. Számok: 0011, 0110, 1100, 1010, 1011. Kódjuk: 1110, 1010, 1100, 1001, 0011.

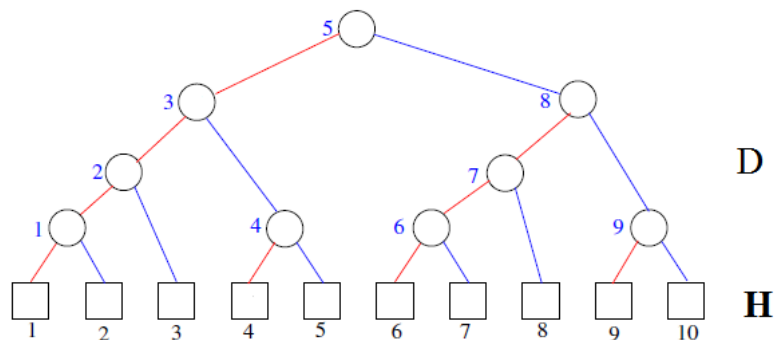
B. Számok: 1010, 0101, 0110, 1001. Kódjuk: 0011, 1100, 0101, 1010.

C. Számok: 00110, 00101, 10000, 10010, 01001. Kódjuk: 01100, 10001, 01010, 00100, 10010

D. Számok: 01011, 00110, 10010, 11010. Kódjuk: 11010, 10100, 10001, 10011.

5. feladat: Mit csinál? (25 pont)

Az alábbi algoritmus a 10 elemű H logikai vektor alapján számítja ki egy függvényértéket és tölti ki a 9 elemű D vektort. Az alábbi ábrán a H vektor elemeit négyzetekkel, a D vektorét pedig körökkel jelöljük:



A. Mi lesz az  $\text{Épít}(1,10)$  függvény értéke és a D vektor tartalma, ha  $H=(\text{hamis}, \text{igaz}, \text{hamis}, \text{igaz}, \text{igaz}, \text{hamis}, \text{igaz}, \text{hamis}, \text{igaz}, \text{igaz})$ ?

B. Fogalmazd meg általánosan is, hogy H-tól függően mi az  $\text{Épít}(1,10)$  függvény értéke és a D vektor tartalma!

```

Épít (bal, jobb) :
    Ha bal<jobb
        akkor s=egészszerés ((bal+jobb)/2)
            b:=Épít (bal, s); j:=Épít (s+1, jobb)
            D(s):=j; Épít:=b+j
        különben ha H(bal) akkor Épít:=1
            különben Épít:=0
    
```

Eljárás vége.

6. feladat: Mit csinál? (25 pont)

Nyolc ember (0-tól 7-ig sorszámozva) mindegyike 1 és 10 közötti minősítést szerzett. Szeretnének ebből valamilyen értéket kiszámolni, de senki sem szeretné, ha az ő minősítését egynél több másik ember ismerné. Kitaláltak egy algoritmust, amelyet ha mindenki végrehajt, akkor a kívánt érték a feltételeknek megfelelően kiszámolható. Az algoritmusban adott rendszer szerint üzenetet kell küldeniük egymásnak és az általuk tudott számot közölni a partnerrel, majd a saját és a kapott számból egy új értéket kell kiszámolni.

Az összes ember programja így néz ki:

Program:

```
i:=1; P:=Azonositom()
Ciklus amíg i*2<8
  Ha P mod (i*2)<i akkor Kuld(P+i,A); B:=Fogad(P+i)
  különben Kuld(P-i,A); B:=Fogad(P-i)
  Ha B<A akkor A:=B
  i:=i*2; Ki: A
Ciklus vége
```

Eljárás vége

A Kuld(kinek, mit) művelet nem blokkoló, hatására a kinek pufferébe kerül a mit érték (egy egész) és tovább folytatódik a program. A Fogad(kitol) blokkoló, azaz amíg a kitol nem hajtott végre Kuld(en, x) műveletet (ahol en a Fogad végrehajtója), addig várakozik. Az Azonositom() művelet megadja a végrehajtó ember azonosítóját (0..7).

A. Mik lesznek a 8 programban a kiírt számok, ha kezdetben a 8 program A változója: 7, 9, 4, 10, 10, 6, 2, 2?

Program	0.	1.	2.	3.	4.	5.	6.	7.
1. kiírás:								
2. kiírás:								
3. kiírás:								

B. Fogalmazd meg általánosan, mi lesz az egyes programok által kiszámított utolsó A érték!

### 7. feladat: Függvény (20 pont)

Egy képzeletbeli programozási nyelven az  $x$ -ekhez és  $y$ -okhoz  $z$ -ket rendelő  $f$  függvény típusát  $f :: x \rightarrow y \rightarrow z$  alakban írjuk le, az a típusú elemekből álló lista típusát  $[a]$ -val, az  $a$  és  $b$  típusú elemekből álló párt pedig  $(a, b)$ -vel jelöljük. Az üres listát  $[]$  jelöli. Hosszabb listákat a  $(:)$  művelettel állíthatunk elő egy elemből és egy listából. Például:  $1 : [2, 3, 4] == [1, 2, 3, 4] == 1 : (2 : (3 : (4 : [])))$ . A függvények működése különbözhet az argumentumok szerkezetétől függően (ezt mintaillesztésnek hívjuk).

Írtunk egy függvényt:

```
valami :: [a] -> [b] -> [(a,b)]
valami as [] = []
valami [] bs = []
valami (a:as) (b:bs) = (a,b) : valami as bs
```

A. Mi a valami  $[1, 2, 3, 4] ['a', 'b', 'c', 'd', 'e', 'f']$  értéke?

B. Mit csinál a függvény, általánosan megfogalmazva?

A függvényt így módosítottuk:

```
valami :: [a] -> [b] -> [(a,b)]
valami as [] = []
valami [] (b:bs) = (0,b) : valami [] bs
valami (a:as) (b:bs) = (a,b) : valami as bs
```

C. Mi lesz most a valami  $[1, 2, 3, 4] ['a', 'b', 'c', 'd', 'e', 'f']$  értéke?

D. Mit csinál a függvény, általánosan megfogalmazva?

**Összpontszám: 200 pont**