

Digitális kultúra emelt

(B) szóbeli mintatételek

köszö Dani :)

- A hivatalos tankönyv (és én is) 0-tól indexeli a listákat, érettségien viszont a feladatokban a listák 1-től vannak indexelve, hacsak a feladat máshogy nem rendelkezik erről.
- Ha kapsz algoritmust, minden esetben lekódolhatod, még ha a feladat nem is kéri. (Kódold is, az segít.)
- Nem minden feladat algoritmizálásra épül. A feladatok kb. az alábbiak szerint oszlanak el egy tételsorban (kb. 25-25-25-25% arányban):
 - o Egyszerű gyakorlati feladat (Pl.: készíts fej vagy írást szimuláló programot, dobókocka dobást szimuláló programot, stb...). Ezek pont olyanok, mint amik a gyakorlati vizsgára kellettek, csak egyszerűbbek. Ilyeneket nem írtam a tételsorba, ezeket gyakoroltatok eleget a gyakorlati vizsgára készülés során.
 - o Algoritmust javító feladatok: Előfordul, hogy egy nem túl bonyolult algoritmust kell javítani, és semmi más dolgod nincs. Ilyenkor több hiba is van az algoritmusban, akár 6-7 is. Általában az alábbi dolgok lehetnek hibásak:
 - Változók értékadása
 - Változók összekeverése
 - Listák indexelése (érettségien pszeudokódban 1-től, de programozáskor 0-tól)
 - Ciklus vagy elágazás feltételek (relációsjel, hossz helyett hossz – 1, stb...)
 - o Algoritmust javító és algoritmust leprogramozó feladat. Ha a feladat nem csak a javítást, hanem az algoritmus leprogramozását is kéri, akkor ebben általában kevesebb hiba van (2-3).
 - o Algoritmust leprogramozó feladat. Kapsz egy nehéz algoritmust, nem hibás, csak le kell kódold, és el kell tudd magyarázni, hogy mit csinál.
- A tételsorom nem fókuszál a leprogramozásra, hiszen arra a gyakorlati rész miatt már készültél. Persze magadnak ezeket bármikor lekódolhatod, de nagyon fontos, hogy úgy menj el vizsgára, hogy pszeudokódot maximálisan tudsz olvasni, értelmezni, átírni. A pszeudokód írást valamelyest kiválthatod azzal, ha megírod az algoritmus programját. Azonban, ha maximum pontra törekszel, akkor az algoritmustkat értelmezned, (át)írnod is tudnod kell.
- Fontos, hogy ezek MINTA tételek. NEM pont ezeket fogod kapni, hasonlóak előfordulhatnak.

Tartalom:

1. tétel: Hibakeresés egy összegzési algoritmusban	4
2. tétel: Pszeudokód kielemezése – Maximumkeresés	5
3. tétel: Pszeudokód írás - Két lista metszetének meghatározása	6
4. tétel: Típusalgoritmus leírása – Másolás.....	7
5. tétel: Hibakeresés egy eldöntési algoritmusban	8
6. tétel: Pszeudokód kielemezése - Páros és páratlan számok szétválasztása	9
7. tétel: Feladatleírás alapján pszeudokód - LKT meghatározása	10
8. tétel: Típusalgoritmus leírása és pszeudokód	11
9. tétel: Hibakeresés egy rendezési algoritmusban (Buborékredezés).....	12
10. tétel: Pszeudokód kielemezése - Fibonacci-sorozat	13
11. tétel: Feladatleírás alapján pszeudokód - Tömb elemeinek átlaga.....	14
12. tétel: Típusalgoritmus leírása és pszeudokód – Összegzés	15
13. tétel: Hibakeresés egy kiválasztási algoritmusban	16
14.tétel: Pszeudokód kielemezése - Prímszámok keresése	17
15. tétel: Feladatleírás alapján pszeudokód - Tömb fordított sorrendbe rendezése	18
16. tétel: Típusalgoritmus leírása és pszeudokód – Kiválogatás	19
17. tétel: Hibakeresés egy megszámlálás algoritmusban	20
18. tétel: Pszeudokód kielemezése - Többször előforduló elemek szűrése	21
19. tétel: Feladatleírás alapján pszeudokód: Legnagyobb és legkisebb elem keresése	22
20. tétel: Típusalgoritmus leírása és pszeudokód: Eldöntés	23

1. tétel: Hibakeresés egy összegzési algoritmusban

Feladat: A pszeudokód a lista elemeinek összegét kalkulálja. Találd meg és javítsd ki a hibát!

Pszeudokód:

összeg := 0

LISTA := [1, 2, 3, 4, 5, 6]

Ciklus i := 1-től LISTA hosszág:

összeg := összeg + LISTA[i]

KIÍR összeg

Megoldás

HIBA: túlmegy az index a ciklusban, vagy 0-tól LISTA hosszág, vagy 1-től LISTA-1 hosszág

javított kód:

összeg:=0

LISTA:=[1, 2, 3, 4, 5, 6]

Ciklus i:=1-től LISTA hossza-1-ig:

összeg:= összeg + LISTA[i]

KIÍR összeg

2. tétel: Pszeudokód kielemezése – Maximumkeresés

Feladat: Magyarázd el, hogyan működik ez az algoritmus.

Pszeudokód:

MAX := LISTA[0]

Ciklus i := 1-től LISTA hossza - 1-ig:

HA LISTA[i] > MAX:

MAX := LISTA[i]

KIÍR "A legnagyobb elem: ", MAX

Megoldás

egy ciklus végigmegy a listán, melynek első eleme meg van adva kezdő maximumértéknek, ezzel összehasonlítja az összes elemet, amennyiben a vizsgált elem nagyobb, mint az addigi legnagyobb, akkor azt teszi legnagyobbnak. Így a ciklus és a lista végén megkapjuk a lista legnagyobb értékét, amit kiírnunk a képernyőre.

3. tétel: Pszeudokód írás - Két lista metszetének meghatározása

Feladat: Írj pszeudokódot, amely meghatározza két számokból álló lista metszetét.

Pszeudokód:

LISTA1:=[1, 3, 5, 7, 9]

LISTA2:=[2, 3, 5, 6, 7, 8]

METSZET:=[]

ciklus i: 1-től LISTA1 hossza-ig:

 ciklus j:= 1-től LISTA2 hossza-ig:

 HA LISTA1[i]=LISTA2[j] és LISTA1[i] nincs benne METSZET-ben:

 Hozzáad METSZET-hez: LISTA1[i]

kiír: METSZET

4. tétel: Típusalgoritmus leírása – Másolás

Feladat: Magyarázd el a Másolás típusalgoritmusát és írd hozzá pszeudokódot.

Megoldás

A másolás algoritmus egy sorozat/lista elemeit átmásolja egy másik sorozatba/listába és közben valamilyen feltételnek megfelelően egyes elemeken valamilyen átalakítást végez.

pl. Egy szövegben a szóközöket le akarok cserélni alsóvonásra:

Pszeudokód:

SZÖVEG: „Milyen csodálatos, kellemes tavaszi idő van.”

SZÖVEG2: „”

ciklus i: 1-től SZÖVEG hossza-ig:

HA SZÖVEG[i] = „ ”:

SZÖVEG2 = SZÖVEG2 + „_”

KÜLÖNBEN:

SZÖVEG2 = SZÖVEG2 + SZÖVEG[i]

KIÍR: SZÖVEG2

5. tétel: Hibakeresés egy eldöntési algoritmusban

Feladat: Az algoritmus eldönti, hogy a `KeresettElem` megtalálható-e a listában.
Hibás-e az algoritmus?

Pszudokód:

Talált := HAMIS

Ciklus i := 0-tól LISTA hossza - 1-ig:

HA LISTA[i] = KeresettElem:

Talált := IGAZ

Kilépés a ciklusból

HA Talált = HAMIS:

KÍR "Az elem nem található."

Megoldás

A ciklus része teljesen jól működik (a pszudokód alapból 1-től indexel, de 0-tól hossz-1-ig is teljesen jó), viszont ha a Talált változó értéke IGAZ, nem ír ki semmit a képernyőre, így ha tartalmazza a lista a keresett elemet, nem kapunk vissza semmit. Valamint a ciklus elején meg kéne adni a listát, valamint a keresett értéket is. A javított megoldás:

Pszudokód:

LISTA = [1, 2, 3, 4, 5, 6]

KeresettElem = 2

Ciklus i:= 1-től LISTA hossza-ig:

HA LISTA[i] = KeresettElem:

Talált:= IGAZ

KÍR „Az elem szerepel a listában.”

Kilépés a ciklusból

HA Talált := HAMIS:

KÍR „Az elem nem található.”

6. tétel: Pszeudokód kielemezése - Páros és páratlan számok szétválasztása

Feladat: Magyarázd el, hogyan választja szét az algoritmus a páros és páratlan számokat.

Pszeudokód:

PÁROSOK := []

PÁRATLANOK := []

Ciklus AMÍG van még elem a LISTA-ban:

 HA LISTA elem MOD 2 = 0:

 PÁROSOK hozzáad LISTA elem

 KÜLÖNBEN:

 PÁRATLANOK hozzáad LISTA elem

KIÍR "Páros számok: ", PÁROSOK

KIÍR "Páratlan számok: ", PÁRATLANOK

Megoldás

Egy ciklus végigmegy a lista minden elemén, és megnézi, hogy 2-vel való osztás esetén van-e maradék. Ez gyakorlatban annyit tesz, hogy ha van maradék (vagy 1, vagy az eredmény), akkor páratlan a nézett szám, ha nincs, akkor páros. Ennek megfelelően hozzáadja az értéket vagy a PÁROSOK, vagy a PÁRATLANOK listához, majd a ciklus lefutása után kiírja mindkét listát.

7. tétel: Feladateleírás alapján pszeudokód - LKT meghatározása

Feladat: Írj pszeudokódot, amely kiszámítja két szám legkisebb közös többszörösét.

Pszeudokód:

Eljárás LNKO(a, b):

Amíg $b \neq 0$:

$r := a \bmod b$

$a := b$

$b := r$

visszaad a

Eljárás LKT(a, b):

$lnko := LNKO(a, b)$

$lkt := (a * b) / lnko$

Visszaad lkt

$a := 12$

$b := 20$

KÍR: „A legkisebb közös többszörös:”, LKT(a, b)

8. tétel: Típusalgoritmus leírása és pszeudokód

Feladat: Magyarázd el az Unio algoritmusát és írd hozzá egy pszeudokódot.

pszeudokód:

LISTA1 := [1, 3, 5, 7, 9]

LISTA2 := [0, 2, 4, 6, 8, 9, 3]

UNIO := []

Ciklus i := 1-től LISTA1 hossza-ig:

HA LISTA1[i] nincs benne UNIO-ban:

METSZET hozzáad: LISTA1[i]

Ciklus l := 1-től LISTA2 hossza-ig:

HA LISTA2[l] nincs benne UNIO-ban:

UNIO hozzáad: LISTA2[l]

KIÍR: UNIO

Az algoritmus két lista elemeinek az unióját adja vissza úgy, hogy mindkét listán külön-külön végigmegy, és amennyiben az adott lista jelenlegi eleme nem szerepel az UNIO nevezetű listában, hozzáadja ahhoz. Ez ebben a formában matematikailag nem felel meg a halmaz definíciójának, ahhoz még egy rendezésre lenne szükség az UNIO listán.

9. tétel: Hibakeresés egy rendezési algoritmusban (Buborékredezés)

Feladat: A pszeudokód célja egy lista buborékredezéssel való rendezése. Találd meg a hibát!

Pszeudokód:

Ciklus i := 0-tól LISTA hossza - 1-ig:

Ciklus j := 0-tól i-ig:

HA LISTA[j] > LISTA[j+1]:

CSERE LISTA[j] és LISTA[j+1]

Megoldás:

A külső ciklus rossz irányba megy, a végétől kell visszafele haladnia, a belsőnek pedig az elejétől a végéig, de csak a még nem rendezett részen.

javítva:

Pszeudokód:

LISTA := [2, 5, 6, 76, 3, 9, 0]

Ciklus i := LISTA hossza-1-től 1-ig:

Ciklus j:= 0-tól i-1-ig:

HA LISTA[j] > LISTA[j+1] :

CSERE LISTA[j] és LISTA[j+1]

KÍR: LISTA

10. tétel: Pszeudokód kielemezése - Fibonacci-sorozat

Feladat: Magyarázd el, hogyan generálja ez az algoritmus a Fibonacci-sorozat első N elemét.

Pszeudokód:

FIBONACCI := [0, 1]

Ciklus i := 2-től N-ig:

 ÚjElem := FIBONACCI[i-1] + FIBONACCI[i-2]

 FIBONACCI hozzáad ÚjElem

KIÍR FIBONACCI

Megoldás

Az algoritmus működéséhez szükséges egy legalább 2 elemű lista. Amennyiben ez megvan, a következő elemet az azt megelőző kettő (első alkalommal az 1 és a 0) összeadásával kapja.

11. tétel: Feladateleírás alapján pszeudokód - Tömb elemeinek átlaga

Feladat: Írj pszeudokódot, amely kiszámítja egy számokból álló tömb elemeinek átlagát.

Pszeudokód:

LISTA := [1, 2, 54, 2, 35, 25]

osszeg = 0

Ciklus i := 1-től LISTA hossza-ig:

 osszeg := osszeg + LISTA[i]

avg := osszeg / LISTA hossza

KÍR avg

12. tétel: Típusalgoritmus leírása és pszeudokód – Összegzés

Feladat: Magyarázd el az összegzés algoritmusának működését és készíts hozzá egy pszeudokódot.

Megoldás

Az összegzés algoritmus egymás után összeadja egy tömb/lista elemeit.

példa:

Pszeudokód:

összeg:=0

LISTA:=[2, 3, 5, 7, 235, 865, 42]

Ciklus I := 1-től LISTA hossza-ig:

 összeg = összeg + LISTA[i]

KÍR összeg

13. tétel: Hibakeresés egy kiválasztási algoritmusban

Feladat: Az algoritmus a legnagyobb elem indexét próbálja megkeresni a listában.
Javítsd ki a hibát!

Pszudokód:

```
LEGNAGYOBB := LISTA[0]
INDEX := 0
Ciklus i := 1-től LISTA hosszáig:
    HA LISTA[i] > LEGNAGYOBB:
        LEGNAGYOBB := i
KIÍR "A legnagyobb elem indexe: ", INDEX
```

Megoldás

A cikluson belül helyesen hasonlítja az aktuális elemet az addigi maximumhoz, de a feltétel beteljesülésekor az INDEX változó helyett a maximum értéknek adja meg az aktuális indexet. Javítva:

Pszudokód:

```
LEGNAGYOBB := LISTA[0]
INDEX := 0
Ciklus i := 1-től LISTA hosszáig:
    HA LISTA[i] > LEGNAGYOBB:
        INDEX := i
        LEGNAGYOBB := LISTA[i]
KIÍR "A legnagyobb elem indexe: ", INDEX
```


14.tétel: Pszeudokód kielemezése - Prímszámok keresése

Feladat: Magyarázd el, hogyan azonosítja az algoritmus a prímszámokat 2 és N között.

Pszeudokód:

```
PRÍMEK := []  
Ciklus i := 2-től N-ig:  
    Prím := IGAZ  
    Ciklus j := 2-től i DIV 2-ig:  
        HA i MOD j = 0:  
            Prím := HAMIS  
    Szakítsd meg a ciklust  
HA Prím:  
    PRÍMEK hozzáad i  
KIÍR PRÍMEK
```

Megoldás

Az algoritmus 2-től kezdve (mivel tudjuk, hogy az 1 nem prímszám, de az általános szabály nem alkalmazható rá) egyesével végigfuttatja minden számon a j ciklust. A j ciklus az adott i szám kettővel való egészrészes osztással kapott számig minden egész számmal megnézi, hogy azzal való elosztást követően van-e maradék, tehát oszthatóak-e. Amennyiben az osztási maradék 0, a két szám osztható egymással, így az aktuálisan vizsgált szám nem prímszám. Ezt a megállapítást követően a j ciklus véget ér. Ha viszont a j ciklus végig tud futni, akkor a Prím nevű boolean változó igaz marad, ezáltal hozzáadjuk a listához.

15. tétel: Feladateleírás alapján pszeudokód - Tömb fordított sorrendbe rendezése

Feladat: Írj pszeudokódot egy tömb elemeinek fordított sorrendbe történő rendezéséhez.

Pszeudokód:

LISTA1 := [1, 2, 3, 4, 5, 6]

LISTA2 := []

Ciklus i:= LISTA1 hossza-1-tol 0-ig, -1 lépésként -1:

LISTA2 hozzáad: LISTA1[i]

KÍR LISTA2

16. tétel: Típusalgoritmus leírása és pszeudokód – Kiválogatás

Feladat: Magyarázd el a kiválogatás algoritmusát, és írd hozzá egy pszeudokódot, amely kiválogatja a páratlan számokat egy tömbből.

Megoldás

A kiválogatás algoritmus egy tömb elemein megy végig és az előre megszabott feltétel alapján kiválogatja azokat az elemeket, amelyekre a feltétel igaz. Pl:

Pszeudokód:

LISTA := [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]

PÁRATLANOK := []

Ciklus i:= 1-től LISTA hossza-ig:

HA LISTA[i] MOD 2 = 1 (ÉS LISTA[i] nem szerepel PÁRATLANOK-ban):

PÁRATLANOK hozzáad: LISTA[i]

KIÍR PÁRATLANOK

17. tétel: Hibakeresés egy megszámlálás algoritmusban

Feladat: A kód a nullák számát próbálja megszámlálni egy listában. Javítsd ki a hibát!

Pszudokód:

NULLÁK := 0

LISTA := [0, 1, 0, 1, 1, 0, 1]

Ciklus i := 0-tól LISTA hosszáig:

HA LISTA[i] = 0:

NULLÁK := NULLÁK + 1

KÍR "Nullák száma: ", NULLÁK

Megoldás

A ciklus túl fog futni. Ezt vagy úgy lehet orvosolni, hogy 0-tól LISTA hossza-1-ig, de az érettségis pszeudokód általában egytől indexel, így a helyes megoldás:

Pszudokód:

NULLÁK := 0

LISTA := [0, 1, 0, 1, 1, 0, 1]

Ciklus i := 1-től LISTA hossza-ig:

HA LISTA[i] = 0:

NULLÁK := NULLÁK + 1

KÍR "Nullák száma: ", NULLÁK

18. tétel: Pszeudokód kielemezése - Többször előforduló elemek szűrése

Feladat: Magyarázd el, hogyan szűri ki az algoritmus a többször előforduló elemeket.

Pszeudokód:

ELEMEK := [1, 2, 2, 3, 4, 4, 4, 5]

EGYEDI := []

Ciklus AMÍG van még elem az ELEMEK-ben:

 HA az ELEM nem szerepel az EGYEDI-ben:

 EGYEDI hozzáad ELEM

KIÍR "Egyedi elemek: ", EGYEDI

Megoldás

Az algoritmusban szereplő ciklus végigmegy az eredeti, ELEMEK nevű tömbön és megnézi, hogy az aktuális elem már szerepel-e a másik, EGYEDI nevű tömbben. Ha még nincs, hozzáadja az EGYEDI tömbhöz, így ha még egyszer előfordulna a másik listában, akkor a feltétel eredménye már HAMIS lenne.

19. tétel: Feladateleírás alapján pszeudokód: Legnagyobb és legkisebb elem keresése

Feladat: Írj pszeudokódot, amely megtalálja és kiírja egy tömb legnagyobb és legkisebb elemét.

Pszeudokód:

LISTA := [2, 45, 24, 78, 1, 35, 0, 764]

MIN := LISTA[1]

MAX := LISTA[1]

Ciklus i := 2-től LISTA hossza-ig:

HA LISTA[i] < MIN:

MIN = LISTA[i]

HA LISTA[i] > MAX:

MAX = LISTA[i]

KIÍR „A tömb legkisebb és legnagyobb eleme:”, MIN, MAX

20. tétel: Típusalgoritmus leírása és pseudokód: Eldöntés

Feladat: Magyarázd el az eldöntés algoritmusát, és írd hozzá egy pseudokódot, amely eldönti, hogy egy adott szám szerepel-e egy listában.

Megoldás

Az eldöntés algoritmus egy előre megszabott feltétellel hasonlít össze egy változót, vagy egy tömb elemeit. Eredményként IGAZ vagy HAMIS értéket ad vissza. pl:

Pseudokód:

LISTA := [1, 2, 3, 4, 5, 6, 7, 8, 9]

Keresett := 4

Szerepel := HAMIS

Ciklus i := 1-től LISTA hossza-ig:

HA LISTA[i] = Keresett:

Szerepel = IGAZ

Szakítsd meg a ciklust.

HA Szerepel:

KIÍR: „A keresett szám szerepel a listában.”

KÜLÖNBEN:

KIÍR: „A keresett szám nem szerepel a listában.”