

enter

INFORMATIKA – tétel

2025

ELIGAZÍTÁS:

- 1 pont hivatalból;
- Az 1-4 feladatokban (a pszeudokód programrészletekben): (1) a kiírások szóközzel elválasztva történnek; (2) a „/” operátor osztási hányadost ad meg, a „%” operátor pedig osztási maradékot.
- Az 5-9 feladatok esetében használj alprogramot, valahányszor célszerűnek találod. Maximális pontszám **hatékony megoldásra** kapható (elsődleges szempont az algoritmusok időigénye). Lásd el **beszédes kommentekkel** programjaidat.
- A bemeneti adatok helyesnek tekinthetők. Minden egész szám eltárolható az `int` tartományban.

FELADATOK:

1. Legyen az alábbi pszeudokód programrészlet:

```
x = 0
y = n - 1
amíg x <= y végezd
| i = (x + y) / 2
| kiír a[i], ' '
| ha a[i] > 0 akkor
| | x = i + 1
| |különben
| | y = i - 1
| L
L
```

Mit ír ki a fenti programrészlet, ha $n = 7$ és az a tömb tartalma a következő $[-9, 5, -1, 0, 3, 7, 10]$? [1 pont]

- a) 0 5 -1
- b) 0 -9 -1
- c) 0 3 7 10
- d) 0 5 -9 -1

2. Legyen az alábbi pszeudokód programrészlet:

```
x = 1
y = n*m
s = -2
amíg x < y végezd
| i = x / m
| j = (x % m) + 1
| s = s + (i/j + i%j)
| x = x + 2
| L
kiír s
```

Mit ír ki a fenti programrészlet, ha $n = 2$ és $m = 3$? [1 pont]

- a) -1
- b) 0
- c) 1
- d) 2

3. Legyen az alábbi pszeudokód programrészlet:

```
x = 0
minden i = 0, 4 végezd
| ha a[i] == b[i] akkor
| | x = x + 1
| különben
| | x = x * 2
|
kiír x
```

Mit ír ki a fenti programrészlet, ha $a = [3, 5, 7, 9, 11]$, $b = [3, 4, 7, 8, 11]$? [1 pont]

- a) 3
- b) 4
- c) 7
- d) 8
- e) 10
- f) 11

4. Legyen az alábbi pszeudokód programrészlet:

```
x = t[n - 1]
minden i = n - 1, 1, -1 végezd
| t[i] = t[i - 1]
|
t[0] = x
```

Mi lesz a t tömb tartalma a futás után, ha $t = [1, 2, 3, 4]$, $n = 4$? [1 pont]

5. Írj Pascal vagy C/C++ *programot*, amely bekéri egy diák matematika és magyar nyelv tantárgyból elért pontszámát (egész számok), majd meghatározza, hogy felvételt nyert-e az egyetemre (IGEN/NEM). A felvételhez a matematika pontszámának legalább 70-nek és a magyarnyelv pontszámának legalább 60-nak kell lennie, vagy a két pontszám összegének el kell érnie a 140-et. [1 pont]

Példa bemenet:

55 90

Kimenet:

IGEN

6. Írj Pascal vagy C/C++ *eljárást* (*void-függvényt*), amely paraméterként kap egy n értéket ($1 < n < 1000$), valamint egy $n \times 2$ méretű kétdimenziós tömböt, ahol minden sorban két elem található, amelyek egy GPS koordináta szélességi fokának, illetve hosszúsági fokának értékét jelölik. Határozzuk meg (és írassuk ki) azoknak a soroknak a sorszámát, illetve értékeit, amelyekben valid GPS koordináták találhatók. Egy GPS koordinátát validnak tekintünk, ha szélességi fokának értéke -90 és $+90$ között, hosszúsági fokának értéke pedig -180 és $+180$ között található. [1 pont]

Lehetséges C/C++ függvényfejléc:

```
void valid_sorok( double x[][2], int n )
```

Példa bemenet:

$n = 10$

```
x = [[53.7446, -0.33525], [-97.828, -74.01042], [102.506, 194.0182],
      [50.75767, -1.5443], [40.85158, 122.74754], [-104.233, 120.89694],
      [48.06919, 11.37703], [27.85, 112.9], [111.332, 185.29], [-20.87306, -
      48.29694]]
```

Kimenet:

```
1: 53.7446, -0.33525
4: 50.75767, -1.5443
5: 40.85158, 122.74754
7: 48.06919, 11.37703
8: 27.85, 112.9
10: -20.87306, -48.29694
```

7. Írj Pascal vagy C/C++ *programot*, amely beolvasson a *szamok.txt* állományból egy n értéket ($1 < n < 1000000$), valamint egy n elemű pozitív egész számokat tartalmazó sorozatot, majd kiírja a képernyőre a számsorozat leghosszabb szigorúan növekvő szakaszának (egymásutáni elemek) a hosszát. [1 pont]

Példa bemenet (*szamok.txt*):

```
10
5 4 7 7 9 11 13 19 19 17
```

Kimenet:

```
5
```

8. Írj Pascal vagy C/C++ *eljárást* (*void-függvényt*), amely paraméterként kap egy n értéket ($1 < n < 10000$), valamint egy n elemű pozitív egész számokat tartalmazó tömböt! A függvény feladata, hogy külön sorokba kiírja azokat a csoportokat, amelyekben a számok ugyanazokból a számjegyekből állnak, tetszőleges sorrendben. Az azonos csoportba tartozó számok számjegy-gyakoriságának teljesen meg kell egyeznie. A számok legfeljebb 10 számjegyből álló pozitív egész számok lehetnek. [1 pont]

Lehetséges C/C++ függvényfejléc:

```
void kiir_hasonlo_csoportok( int x[], int n )
```

Példa bemenet:

```
n = 11
x = [112, 121, 211, 31, 13, 123, 321, 312, 456, 654, 111]
```

Kimenet:

```
111
112 121 211
123 321 312
31 13
456 654
```

9. Egy boldog szám olyan természetes szám, amelyre az alábbi műveletsort ismételve végül 1-et kapunk:

- A szám számjegyeit négyzetre emeljük és összeadjuk.
- Az így kapott számra megismételjük a műveletet.
- Ha ez a folyamat végül 1-et ad, akkor a szám boldog, különben nem boldog (például ha egy végtelen ciklusba kerül).

Például:

```
19 → 12 + 92 = 82
82 → 82 + 22 = 68
68 → 62 + 82 = 100
100 → 12 + 02 + 02 = 1 (tehát boldog szám)
```

Írj Pascal vagy C/C++ *eljárást* (*void-függvényt*), amely két pozitív, 10000000000-nál kisebb egész számot kap paraméterként (például a és b , ahol $a < b$), és kiírja az összes olyan számpárt az $[a, b]$ intervallumban, ahol mindkét szám boldog és a számjegyeik összege azonos. Például a (19, 91) páros mindkét tagja boldog és számjegyösszegük azonos ($1+9 = 9+1 = 10$). [1 pont]