

emter

INFORMATIKA – tétel

2019

ELIGAZÍTÁS:

- 1 pont hivatalból;
- Az 1-4 feladatokban (a pszeudokód programrészletekben): (1) a **kiír** '**\n**' utasítás újsorba ugratja a képernyőn a kurzort; (2) a „/” operátor osztási hányadost ad meg, a „%” operátor pedig osztási maradékot.
- Az 5-9 feladatok esetében használj alprogramot, valahányszor célszerűnek találod. Törekedj hatékony megoldásra az algoritmusok időigénye tekintetében. **Lásd el beszédes kommentekkel programjaidat.**
- A bemeneti adatok helyesnek tekinthetők.

FELADATOK:

1. Legyen az alábbi pszeudokód programrészlet:

```
y = 0
└amíg x > 0 végezd
| y = y * 10 + (9 - x % 10)
| x = x / 10
└─
kiír y
```

Mit ír ki a fenti programrészlet, ha $x=2019$? [1 pont]

897

2. Legyen az alábbi pszeudokód programrészlet:

```
x[1] = 1
h = 2
└minden i = 2, n végezd
| x[i] = 2 * x[i-1] + 1
| h = h * 2
└─
kiír x[n], '\n', h-1
```

Mit ír ki a fenti programrészlet, ha $n=5$? [1 pont]

31

31

3. Legyen az alábbi pszeudokód programrészlet:

```
i = 1
j = 1
k = 1
└amíg i<=n ÉS j<=n végezd
| └ha k%2 == 1 akkor
| | j = j + k
| | különben
| | i = i + k
| └─
| k = k + 1
└─
kiír k
```

Mit ír ki a fenti programrészlet, ha $n=10$? [1 pont]

7

4. Legyen az alábbi pszeudokód programrészlet:

```
x = 1
y = 1
z = 1
k = 1
┌amíg x<=n ÉS y<=n ÉS z<=n végezd
| ┌ha k%3 == 0 akkor
| | x = x + k
| | különben
| | ┌ha k%3 == 1 akkor
| | | y = y + k
| | | különben
| | | z = z + k
| | └─
| └─
| k = k + 1
└─
kiír k
```

Mit ír ki a fenti programrészlet, ha $n=2019$? [1 pont]

110

5. Legyen egy n érték ($0 \leq n \leq 100$), valamint egy n elemű egész számsorozat. Írj Pascal vagy C/C++ programot, amely billentyűzetről beolvassa az n értéket és az n elemű számsorozatot, majd kiírja a képernyőre, hogy a számsorozat melyik végéhez van közelebb nullás érték. Lehetséges válaszok: BAL, JOBB, EGYENLŐ, NINCS [1 pont]

```
#include <iostream>
using namespace std;
int main()
{
    int n, i, x, jobbról, balról, elso, utolso;
    elso = utolso = 0;
    cin >> n;
    for( i = 1; i <= n; ++i )
    {
        cin >> x;
        if( x == 0 )
        {
            if( elso == 0 )
            {
                elso = i;
            }
            utolso = i;
        }
    }
    if( elso == 0 )
    {
        cout << "NINCS";
        return 0;
    }
    balról = elso;
    jobbról = n - utolso + 1;
    if( balról < jobbról )
    {
        cout << "BAL";
    }
    else if( balról > jobbról )
    {
```

```

        cout << "JOBB";
    }
    else
    {
        cout << "EGYENLO";
    }
    return 0;
}

```

6. Legyen egy n érték ($1 \leq n \leq 100$), valamint egy n elemű egész számsorozat. A számsorozatot az alábbi módszer szerint járjuk be: indulunk az első elemtől, lépegetünk jobbra ($n-1$) lépést az n -edik elemig, majd lépegetünk balra ($n-2$) lépést a második elemig, majd újra jobbra, aztán megint balra, és így tovább mindaddig, míg végül már nulla lépést kellene, hogy tegyünk. Írj Pascal vagy C/C++ függvényt, amely paraméterként megkapja a számsorozat hosszát, illetve a tömböt, amely ezt tárolja, és visszatéríti azon elemek összegét, amelyeket a bejárás érint (minden eleme annyiszor kerül bele az összegbe, ahányszor érintettük). [1 pont]

```

int bejarasmenti_osszeg ( int *a, int n ){
    int i, j, k, osszeg = a[0];
    for ( i = 0, j = n-1, k = 0 ; k < n ; ++k ){
        if ( k % 2 == 0 ){
            osszeg += k * a[i];
            ++i;
        }
        else {
            osszeg += k * a[j];
            --j;
        }
    }
    return osszeg;
}

```

7. Írj Pascal vagy C/C++ programot, amely a `matrix.txt` állományból beolvassa az n értéket ($3 \leq n \leq 51$), valamint egy $n \times n$ méretű mátrixot (elemei egész számok), majd kiírja a következő útvonalmenti elemek összegét: indulunk a bal-felső sarokból, lépünk 1-et jobbra, majd 2-t lefele, aztán 3-t jobbra, majd újra lefele 4-et, és így tovább mindaddig, míg kilépünk a mátrixból. [1 pont]

```

#include <stdio.h>
int main()
{
    int a[52][52], n, i, j, k;
    FILE *fin;
    fin = fopen("matrix.txt", "rt");
    fscanf(fin, "%i", &n);
    for ( i = 0 ; i < n ; ++i ){
        for ( j = 0 ; j < n ; ++j ){
            fscanf(fin, "%i", &a[i][j]);
        }
    }
    fclose(fin);
    int lepes = 1;
    int osszeg = 0;
    int irany = 0;
    i = j = k = 0;
    while ( true ){
        osszeg += a[i][j];
        i += irany;
        j += !irany;
        ++k;
        if ( i >= n || j >= n ){ break; }
        if ( k == lepes ){
            irany = !irany;
            k = 0;
        }
    }
}

```



```
int n, i, j;
scanf("%i", &n);
printf("%i\n", n*n-1);
for ( i = 1 ; i < n ; ++i ){
    printf("%i %i\n", i, i+1);
}
for ( i = 1, j = 1 ; i < n ; ++i, j+=n ){
    printf("%i %i\n", j, j+n);
}
for ( i = 1 ; i < n*(n-1) ; ++i ){
    if ( i % n == 0 ){ continue; }
    printf("%i %i\n", i, i+n+1);
}
return 0;
}
```