

1. labor kis ZH, 2014. október 3., péntek 8-10

NÉV:

NEPTUN kód:

Az $e=2,7182818\dots$ matematikai konstans előállítható az alábbi képlettel:

$$e = \frac{1}{0!} + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots$$

Írj programot, amely kiszámolja ezt az első 20 taggal! A faktoriális nagy szám lehet. Tárold azt **double** típusú változóban!

max. 10p, megoldási idő: 15 perc

Program például:

```
#include <stdio.h>

int main(void) {
    double e;      /* e tárolása */
    double fact;   /* Faktoriális tárolása */
    int i;         /* Ciklusváltozók */

    e = 0;
    fact = 1;
    for (i = 1; i < 20; i = i + 1) {
        e = e + 1 / fact; /* Hozzáadjuk e-hez */
        fact = fact*i;
    }

    printf("e = %f", e); /* Kiírjuk */

    return 0;
}
```

1. labor kis ZH, 2014. október 8., Szerda 10-12

NÉV:

NEPTUN kód:

Egy program bekér a felhasználótól két pozitív egész számot, és a kisebbiktől a nagyobbikig növekvő lépésközzel kiírja a számokat. A lépésköz kezdetben 1, és minden lépésben eggyel nő. A program akkor is helyesen működik, ha a felhasználó előbb a felső, aztán az alsó határt adja meg (és fordítva is).

Pl. be: 4 23 ki: 4 5 7 10 14 19

Pl. be: 23 4 ki: 4 5 7 10 14 19

max. 10p, megoldási idő: 15 perc

Program például:

```
int main(void) {
    int elso, masodik, also, felso, i, step;

    scanf("%d", &elso);
    scanf("%d", &masodik);

    if (elso > masodik) { also = masodik; felso = elso; }
    else { also = elso; felso = masodik; }

    i = also;
    step = 1;
    while (i <= felso) {
        printf("%d ", i);
        i = i + step;
        step = step + 1;
    }

    return 0;
}
```