

## Povinné úlohy:

1. Vytvorte program, ktorý pomocou cyklu *for* vypíše nasledujúcu postupnosť:  
5  
4  
3  
2  
1
2. Vytvorte program, ktorý si od používateľa v cykle vypýta 5 čísel a následne vypíše ich súčet.
3. Vytvorte program, ktorý vypíše násobky čísla 5 v rozsahu od 28 po 100 vrátane.
4. Vytvorte program, ktorý vypíše pod seba čísla od *n* do *m*, pričom čísla (*n*, *m*) zadá používateľ. Ošetríte program tak, aby bol funkčný pre oba prípady:  $n < m$  aj  $n > m$ .
5. Vytvorte pomocou programu nasledujúcu tabuľku:

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

Pomôcky: Použite cyklus FOR vnorený do vonkajšieho cyklu FOR.

Použite premennú *x* na iteráciu vo vnorenom cykle a *y* na iteráciu vo vonkajšom cykle.

Pre výpis bez zalomenia riadku použite príkaz `print(x*y, end="")`

Po skončení vnoreného cyklu zalomte riadok napr. prázdny príkazom `print()`

6. Vytvorte program, ktorý načíta číslo *N* a:
  - a) vypíše počet všetkých jeho deliteľov.
  - b) Ak ide o prvočíslo, vypíše je číslo *N* je prvočíslo.
  - c) Vypíše, či ide o dokonalé číslo (t.j. súčet jeho deliteľov okrem seba samého je rovný danému číslu, napr.  $28 = 1+2+4+7+14$ ).

## Voliteľná úloha (+1 za aktivitu):

- Vytvorte program, ktorý bude simulovať [hod kockou](#). Použite funkciu `randint()` z modulu `random`. Používateľ si bude môcť vybrať nasledujúce kocky: 6-strannú (1d6), 10-strannú (1d10), 20-strannú (1d20) a 100-strannú (1d100). Pomocou cyklu `for` umožnite viacnásobné hody (napr.  $3d6 == 3 \times \text{hod šestštrannou kockou}$ ). Vstup nech je presne v tvare "*NdK*", kde *N* je počet hodov a *K* počet strán kocky. Na rozdelenie ([parsovanie](#)) vstupného reťazca `ret` použite metódu `ret.find("d")` (prezentácia INF4-VH9, snímka 8), ktorá nájde deliaci index `d_i`. Ten následne použite na rozdelenie, napr. nasledovne:  
`N=ret[0:d_i]`  
`K=ret[d_i+1:len(ret)]`