

3. 14. října 2026

Tuto středu jsme napsali několik programků:

14 rozklad na součin prvočísel

15 rozklad na součin prvočísel – funkce vracející seznam prvočísel z rozkladu

16 malá násobilka

17 pythagorejské trojice – hledání hrubou silou

18 faktoriály – funkce vracející $n!$

19 kombinační čísla pozor na efektivitu a spolehlivost; zůstat v oboru celých čísel je lepší
vypis Pascalova trojúhelníku

Domácí cvičení

Posílám opět nějaké hračky k vyzkoušení do příště:

1. Využijte funkci *kombin(n)* z dnešního semináře vracející hodnotu kombinačního čísla n nad k a vypočtete součet všech kombinačních čísel v j -tém řádku Pascalova trojúhelníku pro $j = 1, 2, \dots, 5$.

2. Využijte funkci *faktorial(n)* z dnešního semináře, která vrací hodnotu $n!$ a vypočtete následující součet: $1 + 1/1! + 1/2! + 1/3! + 1/4! + \dots + 1/20!$.

3. Pokuste se součet z předchozí úlohy vypočítat efektivněji – bez použití funkce *faktorial(n)*.

4. Pohrajte si s níže uvedenou funkcí *den_tydne(d, m, r)* vracející k danému datu d. m. r den týdne:

a) který den týdne jste se narodili, b) na který den týdne letos připadne Štědrý den (24. 12. 2026),
c) připadne letos státní svátek 17. 11. 2026 na pracovní den?

Těším se na další programko,

Zdeněk Halas

Výpočet dne týdne dle gregoriánského kalendáře

```
def den_tydne(d, m, r):
    dnu_v_mesici = [0, 0, 3, 3, 6, 1, 4, 6, 2, 5, 0, 3, 5]

    if r % 4 == 0:      # přestupné roky
        if r % 100 != 0 or r % 400 == 0:
            for i in range(3, 13):
                dnu_v_mesici[i] = (dnu_v_mesici[i] + 1) % 7

    nazev_dne = ["neděle", "pondělí", "úterý", "středa", "čtvrtek", "pátek", "sobota"]

    r = r - 2001
    i = r + r//4 - r//100 + r//400 + dnu_v_mesici[m] + d

    return nazev_dne[i % 7]
```

```
den = 21
mesic = 10
rok = 2026
```

```
print( den, mesic, rok, den_tydne(den, mesic, rok) )
```