

Napište program v **Pythonu** (nebo v jiném programovacím jazyce, pokud chcete), který řeší vám přidělené úlohy.

Podmínky

- Váš program musí **přesně** řešit zadanou úlohu. To mimo jiné znamená, že funguje a naprosto dodržuje, co je vstupem a výstupem.
- S řešením úloh vám **nesmí** pomáhat žádná umělá inteligence. Spolupráce s jiným žákem je fajn, pokud o ní vím dopředu a odsouhlasím ji. Nebudu podporovat párování lidí, z nichž jeden ví, co dělá, a druhý v tom plave.
- K řešení používejte ideálně pouze části Pythonu, které jsme si vysvětlili. **Použití jakýchkoli funkcí nebo datových struktur, které výrazně zjednodušují úlohu, neberu!** Příkladem takových funkcí je třeba `sort`, která seřadí seznam, nebo `reverse`, která ho otočí.
- **Součástí vašeho programu budou komentáře!** Těmi vysvětlujete, co děláte a hlavně proč.
- Program odevzdáte do Classroomu s koncovkou `.py` (tj. **ne** jako textový soubor a už vůbec ne jako obrázek). Program, který bude obsahovat chyby, kvůli kterým nejde spustit, hodnotím rovnou 0 %. **Program vám vrátím k přepočítání maximálně dvakrát, pokud je odevzdaný včas!**
- Pokud vám jakákoli část zadání nebo obecně cokoli o úkolu není jasné, **pište/přijďte a ptejte se**. Pozdní odevzdání se slovy, „Nechápu, co se po mně chtělo.“ neberu.

Úlohy

1. Napište funkci, která dostane jako parametr list (seznam) čísel typu `int` (celé číslo) a vrátí součet **všech kladných čísel v tomto seznamu** jako `int`. Pokud je seznam prázdný, vrátí 0.
Příklad: pro seznam `[2, -1, 3, -2, 5]` vaše funkce vrátí 10.
2. Napište funkci, která dostane parametrem string, odebere jeho první a poslední prvek a výsledek vrátí opět jako string. Můžete počítat s tím, že obdrženy string má aspoň dva prvky.
Hint: `for` cyklus prochází string prvek po prvku stejně jako třeba seznam.
Příklad: pro string `"čuně"` vaše funkce vrátí `"un"`.
3. Napište funkci, která dostane parametrem kladné číslo (jako `int`) a vrátí součin všech kladných čísel, která jsou mu menší nebo rovna (opět jako `int`).
Nepoužívejte funkci range!
Příklad: pro číslo 7 vaše funkce vrátí 5040.

4. Napište funkci, která dostane parametrem rok (po Kristu, jako int) a vrátí století, do kterého je tento rok zařazen (opět jako int). Pamatujte, že třeba rok 1705 je v 18. století, ale rok 400 je stále ve čtvrtém.
Příklad: pro rok 1609 vaše funkce vrátí 17.
5. Napište funkci, která najde „jehlu v kupce sena“. Dostane parametrem list (seznam) stringů, který obsahuje **přesně jeden** string "jehla". Vrátí **pozici/pořadí** v zadaném seznamu, kde se tento string nachází (jako int).
Příklad: pro seznam ["bla", "ano", "jejda", "jehla", "blb"] vaše funkce vrátí 3.
6. Napište funkci, která dostane tři parametry – hodiny, minuty, vteřiny – jako tři celá čísla (int). Tyto parametry značí množství času, které uběhlo od půlnoci. Vaše funkce vrátí **počet vteřin** (jako int), který uběhl od půlnoci. Nepředstavuje-li trojice parametrů denní čas (například je-li počet hodin větší než 23), vrátí -1.
Příklad: pro vstup 4, 20, 11 vaše funkce vrátí 15611 a pro vstup 20, 69, 420 vrátí -1.
7. Napište funkci, která dostane dva parametry – prvek (jakéhokoli typu) a list. Funkce vrátí True, pokud prvek leží v obdrženém seznamu, jinak vrátí False.
Příklad: pro prvek 7 a seznam ["nic", 2, 3.5, "zas nic"] vrátí funkce False.
8. Napište funkci, která dostane jeden parametr – seznam (list) stringů – a vrátí seznam dvojic (tuple) všech stringů v seznamu spolu s jejich délkou.
Příklad: pro seznam ["nic", "slovo", "kočka"] má vaše funkce vrátit [("nic", 3), ("slovo", 5), ("kočka", 5)].