

D0009E - Övningsuppgifter

Jun 23, 2026, 2 min read

Övningsuppgifter

En övningsuppgift per föreläsning med tillhörande lösningar.

Föreläsning 2 – Multiplikationstabell

Koncept: [For-loopar](#)

- Skriv ett program som skriver ut en multiplikationstabell för ett tal, t.ex. 3:ans tabell från 0 till 9.
- Skriv programmet så att det är enkelt att ändra 3:an till ett annat tal.
- Lösning: [simple_mul_table.py](#)

Föreläsning 3 – Multiplikationstabell med funktioner

Koncept: [Funktioner](#), [Parametrar](#)

- Modifiera F2-lösningen: fråga användaren efter vilket tal, flytta utskriften till en funktion med två parametrar (tabell och rad).
- Lösning: [simple_mul_table_functions_interactive.py](#)

Föreläsning 4 – Rekursiv multiplikationstabell

Koncept: [Rekursion](#)

- Modifiera F3-lösningen: inga upprepade kodsnuttar – använd rekursion och villkor istället.
- Extrauppgift: låt programmet fråga efter start- och stoppvärde.

- Lösning: [simple_mul_table_recursive.py](#)

Föreläsning 5 – sumTo och sumFromTo

Koncept: [Rekursion](#), [Returvärden](#)

1. [sumTo\(n\)](#) – returnera summan av alla tal 0..n, rekursivt (returnera, skriv inte ut).
2. [sumFromTo\(start, stop\)](#) – returnera summan av start..stop, rekursivt.
3. [sumToUI\(\)](#) och [sumFromToUI\(\)](#) – UI-funktioner som frågar användaren, anropar respektive funktion, och skriver ut resultatet.

- Lösning: [sumTo_sumFromTo.py](#)

Föreläsning 6 – Iteration

Koncept: [For-loopar](#), [While-loopar](#)

- Lös uppgifterna från F4 och F5 igen, men med iteration (for/while) istället för rekursion.
- Lösning: [simple_mul_table_iterative.py](#)

Föreläsning 7 – prefixLength

Koncept: [Strängar](#)

- Skriv funktionen [prefixLength\(s1, s2\)](#) som räknar och returnerar hur många inledande tecken som är gemensamma mellan strängarna s1 och s2.
- Lösning: [prefixLength.py](#)

Föreläsning 8 – Listor

Koncept: [Listor](#), [Rekursion](#)

1. [sumRec\(lst\)](#) – summera alla tal i lst rekursivt, returnera summan.
2. [sumIter\(lst\)](#) – summera alla tal i lst iterativt, returnera summan.
3. [selectLargest\(lst1lst\)](#) – ta en lista av listor, returnera alias till listan med störst summa.

- Lösning: `select_largest.py`

Föreläsning 9 – Histogram

Koncept: `Kataloger`

1. `histogram(str)` – returnera dictionary med tecken→frekvens.
2. `histogram2(str)` – returnera lista av par (tecken, frekvens), utan dictionary.
3. `printHist(hist)` – skriv ut tabellformad utskrift med tecknet i vänster kolumn och frekvensen i höger.

- Lösning: `histogram.py`
-