Velký domácí úkol č. 1

Deadline: středa 21. října 23:59.

Odkaz na odevzdávárnu zde

Za každou funkci můžete získat až 6 bodů, za celý úkol dohromady tedy až 30 bodů. Do funkcí v kostře doplňte svou implementaci a soubor du01.py odevzdejte do odevzdávárny pro první úkol. Pracujte samostatně. Pokud si s něčím nejste jistí, zeptejte se v diskuzním fóru, nebo ismejlem. Kdo úkol neodevzdá, automaticky dostává známku X, nenechávejte si tedy práci na poslední chvíli, nemuseli byste vše stihnout.

1) Napište funkci product(number) pro výpočet postupného ciferného součinu. Postupný ciferný součin získáme tak, že z čísla utrhneme poslední cifru, tou jej vynásobíme, a na výsledek opakujeme postup tak dlouho, dokud nezískáme jednociferný výsledek. (Délka referenční implementace: 4 ř.)

Příklad pro vstup 253: 253 → 25 * 3 = 75 → 7 * 5 = 35 → 3 * 5 = 15 → 1 * 5 = 5

2) Napište funkci sequence(length), která vypíše prvních 2 * length členů následující posloupnosti:

0, 0, 2, -3, 4, -6, 6, -9, 8, -12…

Všimněte si, že kladná čísla jsou násobky dvou a záporná čísla násobky tří. (Délka referenční implementace: 3 ř.)

3) Napište funkci asciiart(size), která v textové grafice vykreslí čtverec o hraně 2 * size se symbolem X ve čtverci. (Délka referenční implementace: 6 ř.)

Výstup pro size = 4 (velikost čtverce 2 * 4 = 8:

```
x x x x x x x x
x x           x x
x   x       x   x
x     x x       x
x     x x       x
x   x       x   x
x x           x x
x x x x x x x x
```

4) Napište funkci evaluate(a, b, mode), která vyhodnotí logickou operaci v trojhodnotové logice a vrátí výsledek jako řetězec. Vstupní hodnoty a, b jsou řetězce a nabývají logických hodnot False (0), Maybe (1/2), True (1); mode nabývá hodnoty True pro logický součin (and) a False pro logický součet (or). Výsledek operací si můžete přečíst v tabulkách kartézského součinu. Všimněte si barevně zvýrazněných buněk, celý výpočet lze provést pomocí tří podmínek. (Délka referenční implementace: 4 ř.)

and	0	½	1
0	0	0	0
½	0	½	½
1	0	½	1

or	0	½	1
0	0	½	1
½	½	½	1
1	1	1	1

Příklady:

`evaluate(“True”, “Maybe”, True) → 1 and ½ → ½ → “Maybe”`

`evaluate(“False”, “Maybe”, True) → 0 and ½ → 0 → “False”`

`evaluate(“True”, “Maybe”, False) → 1 or ½ → 1 → “True”`

`evaluate(“False”, “Maybe”, False) → 0 or ½ → ½ → “Maybe”`

5) Napište funkci superTriangle(side) pro vykreslení trojúhelníku sestávajícího z trojúhelníků. Návod: Udělejte si pomocnou funkci triangle(side), která vykreslí trojúhelník o straně side. V superTriangle tuto funkci opakovaně volejte s postupně se zmenšujícím argumentem. Strana menšího trojúhelníku je vůči straně jeho většího brášky vždy poloviční; zmenšujte stranu tak dlouho, dokud nebude zanedbatelně malá (např menší, než 10 jednotek). Potom se přesuňte do druhého vrcholu vnějšího trojúhelníka a postup opakujte. (Délka referenční implementace: 12 ř.)