



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Zadanie Dukaty – miniLOGIA 16 (2017/18), etap 2

Treść zadania

Bajtek planuje systematycznie oszczędzać. Będzie codziennie rano dokładać jednego dukata do skarbonki, a co piąty dzień dołoży również kolejne dwa dukaty. W momencie, gdy w skarbonce uzbiera się co najmniej 50 dukatów Bajtek będzie z niej pobierał $\frac{3}{4}$ dukatów na zakup ulubionych książek. Dukat jest najmniejszą jednostką, dlatego Bajtek pobierze ze skarbonki maksymalnie dużo dukatów, ale nie więcej niż zaplanował. Zdefiniuj jednoparametrową funkcję **dukaty**, która pomoże Bajtkowi kontrolować zawartość skarbonki. Wynikiem funkcji jest zawartość skarbonki w dniu podanym jako parametr, po dokonanych operacjach.

Przykłady:

Python:

wynikiem **dukaty(3)** jest **3**,

wynikiem **dukaty(6)** jest **8**,

wynikiem **dukaty(36)** jest **13**.

Logo:

wynikiem **dukaty 3** jest **3**,

wynikiem **dukaty 6** jest **8**,

wynikiem **dukaty 36** jest **13**.

Omówienie rozwiązania

Zadanie dukaty należy do grupy zadań symulacyjnych. Analizując treść zadania, należy zwrócić uwagę na kluczowe elementy związane z oszczędzaniem. Zawartość skarbonki zmienia się w następujący sposób:

- codziennie przybywa jeden dukat,
- co piąty dzień kolejne dwa dukaty,
- jeśli w skarbonce jest co najmniej 50 dukatów pobieramy nie więcej niż $\frac{3}{4}$ dukatów, ale maksymalnie dużo, pamiętając o tym, że dukaty są niepodzielne.

Symulacja, rozpisana na kolejne dni i wykonywane operacje, będzie wyglądać tak:

Dzień	Dodane dukaty	Stan po dodaniu dukatów	Pobrane dukaty	Stan po pobraniu dukatów
0				0
1	1	1	0	1
2	1	2	0	2
3	1	3	0	3
4	1	4	0	4
5	1 + 2 = 3	7	0	7
6	1	8	0	8



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

...				
10	$1 + 2 = 3$	14	0	14
...				
35	$1 + 2 = 3$	49	0	49
36	1	50	$(3 * 50) \text{ div } 4 = 37$	$50 - 37 = 13$

Każdego kolejnego dnia zmienia się zawartość skarbonki. Zapis algorytmu może być następujący:

1. $\text{stan} \leftarrow 0$
2. dla każdego kolejnego dnia, od 1 do ostatniego dnia
 - $\text{stan} \leftarrow \text{stan} + 1$
 - jeśli $\text{kolejny} \bmod 5 = 0$
 - $\text{stan} \leftarrow \text{stan} + 2$
 - jeśli $\text{stan} \geq 50$
 - $\text{stan} \leftarrow \text{stan} - (3 * \text{stan}) \text{ div } 4$
3. wynik stan

uwaga: *mod* – reszta z dzielenia całkowitego, *div* – dzielenie całkowite

Rozwiązanie w języku Python

```
1. def dukaty(dzien):
2.     stan = 0
3.     for kolejny in range(1, dzien + 1):
4.         stan = stan + 1
5.         if kolejny % 5 == 0:
6.             stan = stan + 2
7.         if stan >= 50:
8.             stan = stan - (3 * stan) // 4
9.     return stan
```

Rozwiązanie w języku Logo

```
1. oto dukaty :dzien
2. niech "stan 0
3. powtórz :dzien [
4.     zwiększ "stan
5.     jeśli reszta npw 5 = 0 [ (zwiększ "stan 2) ]
6.     jeśli :stan >= 50 [ niech "stan :stan - ilorazc (3 * :stan) 4 ]
7. ]
8. wy :stan
9. już
```

W obu językach programowania rozwiązania są identyczne z dokładnością co do składni. Warto zwrócić uwagę na brak operatorów operacji modulo i ilorazu całkowitego w języku Logo. Operatory zastąpione są funkcjami, odpowiednio **reszta** dla operacji modulo oraz **ilorazc** dla ilorazu całkowitego.

Warto zauważyć, że stan skarbonki uprawniający do wypłaty to: 50 ($49 + 1$), 51 ($48 + 3$) lub 52 ($49 + 3$). Różnica stanu skarbonki i kwoty do wypłaty jest stała i wynosi 13:



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

$$50 - 3 * 50 \text{ div } 4 = 13$$

$$51 - 3 * 51 \text{ div } 4 = 13$$

$$52 - 3 * 52 \text{ div } 4 = 13$$

Zatem stan skarbonki po wypłacie można zapisać tak:

```
stan = 13 #Python  
niech "stan 13 ;Logo
```

Nigdzie w treści zadania nie napisano, że dukaty do skarbonki trafiają po jednej sztuce i stan skarbonki należy analizować po dodaniu pojedynczego dukata tak, aby wypłaty dokonać w momencie, gdy jej zawartość wyniesie dokładnie 50 (a nie co najmniej 50 – 50, 51 lub 52). Dlatego symulacje przeprowadzone w inny sposób nie dawały poprawnego wyniku.

Testy

Testowanie rozwiązania warto rozpocząć od wartości, dla których jest łatwo ręcznie policzyć wynik.

Logo	Python	
dukaty 1	dukaty(1)	1
dukaty 20	dukaty(20)	28
dukaty 5	dukaty(5)	7
dukaty 40	dukaty(40)	19
dukaty 37	dukaty(37)	14
dukaty 89	dukaty(89)	49
dukaty 1009	dukaty(1009)	29
dukaty 1234	dukaty(1234)	45
dukaty 4	dukaty(4)	4
dukaty 3000	dukaty(3000)	18

Testy zostały tak dobrane, żeby sprawdzić wszystkie możliwe przypadki zachodzące podczas symulacji:

- pierwszy i czwarty dzień,
- pierwszy dzień, w którym dołożono trzy dukaty,
- wielokrotność piątego dnia, bez dotychczasowego opróżniania skarbonki,
- dzień przed opróżnieniem skarbonki,
- następny dzień po opróżnieniu skarbonki,
- odległe dni w symulacji, gdzie w międzyczasie wielokrotnie opróżniano skarbonkę; w tych dniach dokładano do skarbonki 1 lub 3 dukaty.