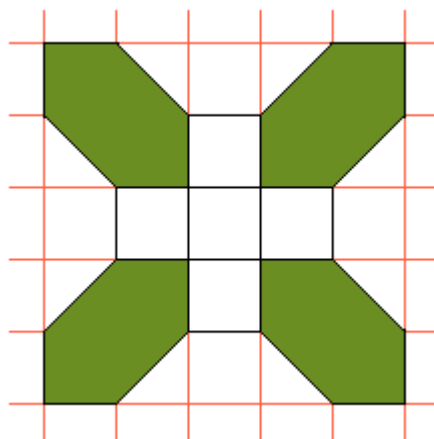


Zadania 3 etapu konkursu miniLOGIA 16

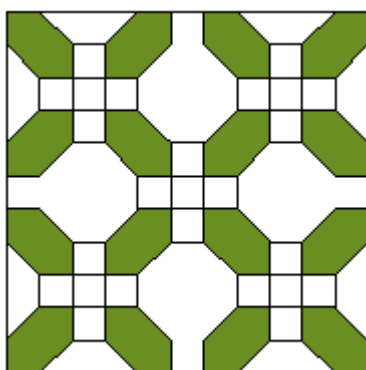
– przedmiotowego konkursu informatycznego dla uczniów
szkół podstawowych województwa mazowieckiego
22 marca 2018 roku

Zadanie 1

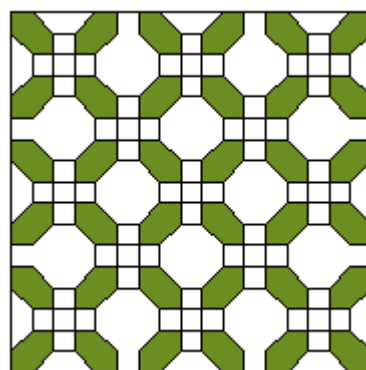
Napisz jednoparametrową procedurę/funkcję **posadzka**, po wywołaniu której na środku ekranu powstanie posadzka, jak na przykładowych rysunkach. Parametr określa liczbę kafelków w najniższym rzędzie i może przyjmować wartości od **1** do **8**. Długość boku posadzki wynosi **480**.



Rysunek pomocniczy kafelka



Logo – posadzka 2, Python – posadzka(2)



Logo – posadzka 3, Python – posadzka(3)

Zadanie 2

Michał przygotowuje Własny System Kodów Paskowych (WSKP) i dlatego szuka liczb, których suma cyfr jest większa od **a** i mniejsza od **b**. Przegląda kolejne liczby począwszy od 1 do 100 000 (włącznie). Napisz trójparametrową funkcję **jaka**. Pierwszym parametrem funkcji jest lewy koniec badanego przedziału (**a**), drugim prawy koniec (**b**). Wynikiem funkcji jest **n**-ta kolejna liczba (określona przez trzeci parametr), której suma cyfr jest większa niż **a** i mniejsza niż **b**. Jeśli taka liczba nie istnieje, to wynikiem jest -1. Przyjmij, że trzeci parametr **n** wynosi co najwyżej **100**, natomiast **a < b < 40**. Przykłady:

Logo:

wynikiem **jaka 1 10 4** jest **5**,

wynikiem **jaka 1 3 100** jest -1.

Python:

wynikiem **jaka(1, 10, 4)** jest **5**,

wynikiem **jaka(1, 3, 100)** jest -1.

W pierwszym przykładzie kolejne liczby spełniające warunek to 2, 3, 4, 5. W drugim przykładzie istnieje tylko 15 liczb spełniających warunek.

Zadanie 3

Robot porusza się od lewego do prawego końca korytarza. Został tak zaprogramowany, że wykonuje tylko dwie podstawowe czynności: idzie naprzód o 1 krok lub zawraca, gdy nie może pójść naprzód. Na początku robot stoi w lewym końcu korytarza, jest skierowany w kierunku prawego i jest naładowany.

Po uruchomieniu porusza się do drugiego końca, a potem zawraca. Wykonując czynności (ruch lub zawracanie) traci energię, więc co jakiś czas musi zostać doładowany. Doładowanie nie jest jednak pełne, każde kolejne umożliwia wykonanie o jedną czynność mniej niż poprzednie.

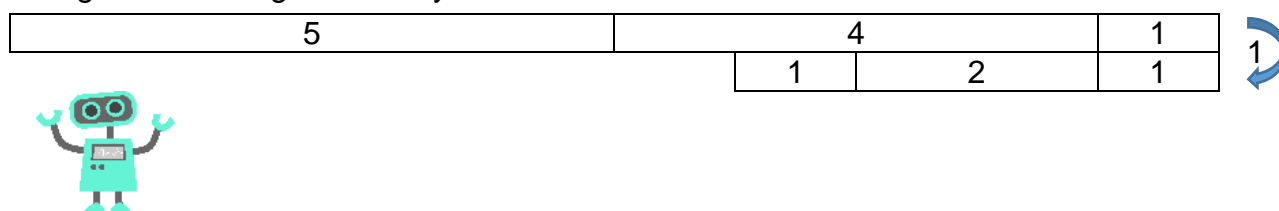
Napisz dwuparametrową funkcję **robot**. Pierwszy parametr określa liczbę czynności, jaką robot może wykonać na pełnym naładowaniu, drugi długość korytarza wyrażoną w krokach robota. Oba parametry są liczbami naturalnymi od 1 do 1000. Wynikiem funkcji jest odległość robota od lewego końca po wykonaniu wszystkich możliwych czynności.

Przykład 1:

Dla wartości pierwszego naładowania równej 5 i długości korytarza 10.

Robot po pierwszym naładowaniu idzie 5 kroków, po drugim naładowaniu 4 kroki, po trzecim 1 krok, zawraca i idzie 1 krok, po czwartym naładowaniu 2 kroki, a po ostatnim 1 krok.

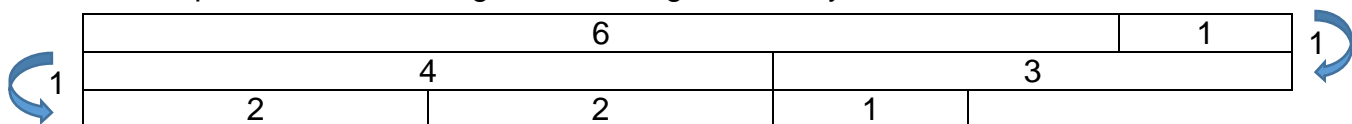
Odległość od lewego końca wynosi 6.



Przykład 2:

Dla wartości pierwszego naładowania równej 6 i długości korytarza 7.

Robot po pierwszym naładowaniu idzie 6 kroków, po drugim naładowaniu 1 krok, zawraca, a potem 3 kroki, po trzecim 4 kroki, po czwartym zawraca i idzie 2 kroki, po piątym idzie 2 kroki, a po ostatnim 1. Odległość od lewego końca wynosi 5.



Logo:

wynikiem **robot 5 10** jest 6,
wynikiem **robot 6 7** jest 5.

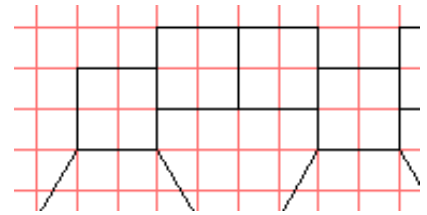
Python:

wynikiem **robot(5,10)** jest 6,
wynikiem **robot(6,7)** jest 5.

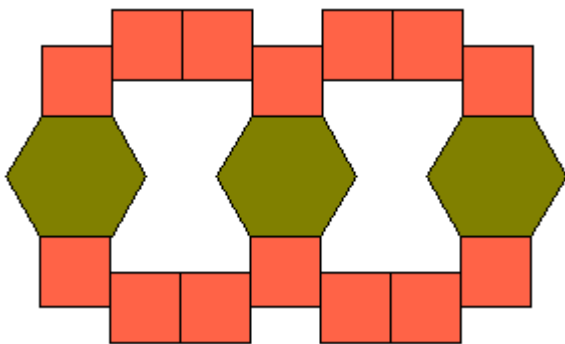


Zadanie 4

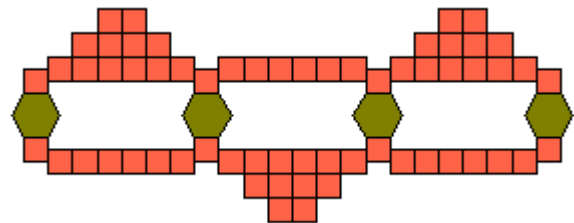
Napisz dwuparametrową procedurę/funkcję **ogniwa**, po wywołaniu której na środku ekranu powstanie rysunek łańcuszka złożonego z dwukolorowych ogniów powstałych z kwadratowych i sześciokątnych koralików. Pierwszy parametr określa liczbę sześciokątnych koralików w łańcuszku i może przyjmować wartości od **3** do **10**. Drugi parametr określa liczbę kwadratowych koralików w najdłuższym rzędku ogniwa i może przyjmować wartości od **2** do **14**. Szerokość rysunku wynosi **700**.



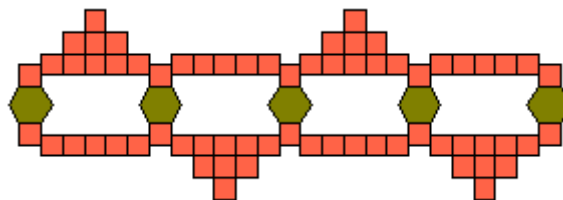
Rysunek pomocniczy



Logo – ogniwa 3 2
Python – ogniwa (3, 2)



Logo – ogniwa 4 6
Python – ogniwa (4, 6)



Logo – ogniwa 5 5
Python – ogniwa (5, 5)