



# Zadania trzeciego etapu konkursu Logia17

– przedmiotowego konkursu informatycznego  
dla uczniów gimnazjów województwa mazowieckiego  
9 marca 2017 roku

## Zadanie 1 (litera).

Napisz jednoparametrową funkcję `litera`, której parametrem jest niepusta lista słów składających się z małych liter alfabetu łacińskiego. Wynikiem funkcji jest ta litera, która występuje najczęściej. Jeśli jest więcej niż jedna taka litera, to wynikiem funkcji jest uporządkowana rosnąco lista liter mających tę własność.

*Przykłady:*

w Logo – wynikiem `litera [ala ma kota]` jest "a", wynikiem `litera [julka lubi psy]` jest [l u],

w Pythonie – wynikiem `litera(['ala', 'ma', 'kota'])` jest 'a', wynikiem `litera(['julka', 'lubi', 'psy'])` jest ['l', 'u'].

## Zadanie 2 (abc).

Jaś przygotował naszyjnik dla Małgosi. Małgosia poprosiła go o usunięcie niektórych koralików tak, aby wszystkie niebieskie koraliki były przed czerwonymi i zielonymi, a wszystkie czerwone – przed zielonymi. Pozostałe koraliki mogą występować w dowolnych miejscach.

Napisz jednoparametrową funkcję `abc`, której wynikiem jest minimalna liczba koralików do usunięcia. Parametrem funkcji jest niepuste słowo złożone z małych liter alfabetu łacińskiego opisujące kolory kolejnych koralików w oryginalnym naszyjniku. Kolejne znaki oznaczają kolory: n – niebieski, c – czerwony, z – zielony, a pozostałym literom przypisane są inne kolory. Liczba koralików jest nie większa niż 10 000.

*Przykłady:*

w Logo – wynikiem `abc "nncnbnfbbbbbcccczcz"` jest 2 (wystarczy usunąć trzeci i przedostatni koralik),  
wynikiem `abc "zzzznnnnz"` jest 4 (wystarczy usunąć wszystkie niebieskie koraliki),

w Pythonie – wynikiem `abc('nncnbnfbbbbbcccczcz')` jest 2 (wystarczy usunąć trzeci i przedostatni koralik),  
wynikiem `abc('zzzznnnnz')` jest 4 (wystarczy usunąć wszystkie niebieskie koraliki).

## Zadanie 3 (planeta).

Na pewnej planecie domy mają adresy będące parami liczb całkowitych dodatnich – ich współrzędnych. Rozmiar planety to maksymalna możliwa wartość współrzędnej. Na planecie o rozmiarze  $N$  jest  $N \times N$  adresów, od (1, 1) do (N, N).

Po planecie można poruszać się tylko w kierunkach północ↔południe oraz wschód↔zachód (nie po skosie), także w kółko. Odległość między domami jest sumą minimalnych różnic współrzędnych. Na przykład dla planety o rozmiarze 10 dom o adresie (6, 7) jest oddalony od domu o adresie (7, 9) o  $1 + 2 = 3$ , a dom o adresie (1, 4) jest oddalony od domu o adresie (8, 9) o  $3 + 5 = 8$ , a nie  $7 + 5 = 12$ .

Na planecie wyróżniamy osiedla, tj. rozłączne zbiory domów. Osiedla mogą składać się z jednego lub większej liczby domów. Dom należy do osiedla, gdy jego odległość od pewnego domu tego osiedla jest nie większa niż 5. Jeśli odległość od danego domu do każdego innego na planecie jest większa niż 5, to taki dom stanowi jednodomowe osiedle.

Napisz dwuparametrową funkcję `planeta`, której pierwszym parametrem jest rozmiar planety, a drugim lista adresów domów. Wynikiem jest liczba domów w osiedlu składającym się z największej liczby domów. Na planecie stoją co najmniej 2 domy i nie więcej niż 5 000. Rozmiar planety jest nie mniejszy niż 2 i nie większy niż 5 000. Adresy domów nie powtarzają się.

*Przykłady:*

w Logo – wynikiem `planeta 12 [[3 1][1 1][1 3][2 12][9 5][8 6]]` jest 4,

wynikiem `planeta 100 [[6 6][6 11][11 6][11 11][80 80]]` jest 4,

w Pythonie – wynikiem `planeta(12, [[3, 1], [1, 1], [1, 3], [2, 12], [9, 5], [8, 6]])` jest 4,

wynikiem `planeta(100, [[6, 6], [6, 11], [11, 6], [11, 11], [80, 80]])` jest 4.

## Zadanie 4 (zawody).

Zawodnicy (co najmniej dwóch, co najwyżej dwudziestu sześciu) przed startem otrzymują identyfikatory oznaczone kolejnymi wielkimi literami alfabetu łacińskiego (A, B, C, ... itd.), których nie zmieniają podczas zawodów. Zawody składają się z kolejnych rund. Sekwencja zawodników w kolejnych rundach zostaje zachowana, ale zawodników ubywa. Po każdej rundzie odpada zawodnik, który w danej rundzie uzyskał najmniej punktów. Wyniki każdej rundy zapisywane są w liście zawierającej liczby punktów uzyskanych przez kolejnych zawodników w tej rundzie (zakładamy, że każdy zawodnik ma inną liczbę punktów).

Napisz jednoparametrową funkcję `zawody`, której parametrem jest poprawna lista opisująca przebieg zawodów, tj. lista wyników kolejnych rund. Wynikiem funkcji jest litera oznaczająca zawodnika, który zwyciężył, czyli wygrał ostatnią rundę.

*Przykłady:*

w Logo – wynikiem `zawody [[8 9]]` jest "B", wynikiem `zawody [[4 0 2 1][1 2 3][2 1]]` jest "C" (po pierwszej rundzie odpadł zawodnik B, po drugiej A, a w ostatniej lepszy był C niż D),

w Pythonie – wynikiem `zawody([[8, 9]])` jest 'B', wynikiem `zawody([[4, 0, 2, 1], [1, 2, 3], [2, 1]])` jest 'C' (po pierwszej rundzie odpadł zawodnik B, po drugiej A, a w ostatniej lepszy był C niż D).