

Kurs języka Ruby

Lista 1.

Zadanie 1. Zaprogramuj

1. funkcję *sloownie*, której argumentem jest liczba naturalna, a wynikiem string ze słownym zapisem liczby, np. wynikiem *sloownie(123)* jest *"sto dwadzieścia trzy"*. Możesz przyjąć, że argument jest nie większy niż dziewięć tysięcy dziewięćset dziewięćdziesiąt dziewięć.
2. jednoargumentową funkcję *pascal* wypisującą na ekran kolejne wiersze trójkąta Pascala. Argumentem jest liczba wierszy trójkąta do wypisania. Na przykład po wywołaniu *pascal(3)* powinno się pojawić

```
1
1 1
1 2 1
```

3. jednoargumentową funkcję *wielkaliczba*, która dla zadanego argumentu (liczby naturalnej) wypisuje na ekran tę liczbę ale powiększoną, np. po wywołaniu *wielkaliczba(123)* powinno się pojawić

```
  x      x      xxxx
 xx     x  x    x   xx
x  x      x      xx
  x      x      x   xx
  x     xxxxxx   xxxx
```

Oczywiście, kształt cyfr może się różnić od tych w przykładzie;

4. jednoargumentową funkcję *podzielniki(n)*, gdzie n jest liczbą naturalną, zwracającą tablicę dzielników pierwszych liczby n bez powtórzeń, np. *podzielniki(1025)* powinno zwrócić `[5, 41]`. Co prawda 5 dwukrotnie dzieli 1025, jednak w wyniku nie powinno być powtórzeń;
5. dwuargumentową funkcję *godzina*, której argumentami są dwie liczby naturalne oznaczające godzinę i minutę. Wynikiem działania funkcji ma być string zwracający tę godzinę w potocznej formie. Przykładowo *godzina(12,45)* powinna zwrócić *"za kwadrans pierwsza"*.

Uwaga: można przyjąć, że argumenty są zawsze poprawne i nie jest konieczna kontrola argumentów.

Najwygodniej jest umieścić wszystkie rozwiązania w jednym pliku, a na końcu umieścić kilka testów prezentujących możliwości zaimplementowanych funkcji. Za każde zadań można otrzymać 2 punkty, jednak za całą listę nie można otrzymać więcej niż 4 punkty (a więc do oceny można przedstawić co najwyżej 2 funkcje).

Marcin Młotkowski