22 października 2018

Kurs języka Ruby

Lista 3.

Zadanie 1. Bloki z jednym parametrem można traktować jak definicje jednoargumentowych funkcji. Korzystając z tej obserwacji zaprogramuj dwie procedury. Pierwsza z nich `calka(a,b,&b)` powinna obliczać numerycznie całkę oznaczoną na przedziale $[a, b]$ funkcji zadanej jako blok. Dokładność obliczeń może być ustalona. Druga funkcja to `wykres(a, b, &blok)`, która za pomocą znaków ASCII naszkicuje wykres funkcji danej jako blok. Można przyjąć arbitralny rozmiar terminala.

Implementacje poniższych funkcji powinne być w postaci jednego wyrażenia. Jest to możliwe używając tylko zakresów, operacji na tablicach i bloków. W przypadku bardzo długich wyrażeń akceptowane będzie podzielenie rozwiązania na podwyrażenia.

Zadanie 2. Zaprogramuj

- jednoargumentową funkcję `pierwsza(n)`, która zwraca tablicę liczb pierwszych nie większych niż n.
- jednoargumentową funkcję `doskonale(n)`, która zwraca tablicę liczb doskonałych nie większych niż n, na przykład

```
doskonale(1000)
==> [6, 28, 496, 8128]
```

Zadanie 3. Zaprogramuj

- jednoargumentową funkcję `rozklad(n)` która oblicza rozkład liczby n na czynniki pierwsze i zwraca jako wynik tablicę tablic $[[p_1, w_1], [p_2, w_2], \ldots, [p_k, w_k]]$ taką, że $n = p_1^{w_1} * p_2^{w_2} * \ldots * p_k^{w_k}$ oraz $p_1, \ldots, p_k$ są różnymi liczbami pierwszymi. Na przykład

```
rozklad(756)
==> [[2, 2], [3, 3], [7, 1]]
```

- jednoargumentową funkcję `zaprzyjaznione(n)`, która zwraca tablicę par liczb zaprzyjaźnionych nie większych niż n, na przykład

```
zaprzyjaznione(1300)
==> [[220, 284], [1184, 1210]]
```

Zadanie 4. Zaprogramuj funkcję `najkrotsza_sciezka(odleglosci, lista_miast)` wyznaczającą najkrótszą trasę między miastami. Argument `odleglosci` zawiera odległości między miastami. Zakładamy, że odległość między miastem A i B jest taka sama jak między B i A. `lista_miast` to lista miast jakie chcemy odwiedzić. Wynikiem jest trasa, tj. lista miast w takiej kolejności, aby trasa była najkrótsza.

Zakładamy, że

- trasa kończy się tam, gdzie się rozpoczęła;
- `lista_miast` zawiera tylko miasta będące w `odleglosci`.

Aby rozwiązać to zadanie, przejrzyj dokumentację klasy ***Array***, niektóre metody mogą się okazać przydatne.

Za każde zadanie można otrzymać do 4 pkt, maksymalnie za całą listę: 8 pkt.

Marcin Młotkowski