2.2.
0–1–2

Zadanie 2.2. (0–2)

Napisz algorytm, który sprawdzi, czy dwie podane liczby są liczbami zaprzyjaźnionymi. Do wykonania zadania wykorzystaj *funkcję suma* z zadania 2.1. Podaj specyfikację algorytmu.

Miejsce na rozwiązanie:

Zadanie 3. Anagramy (0–11)

Anagram oznacza wyraz, wyrażenie lub całe zdanie powstałe przez przestawienie liter bądź sylab innego wyrazu lub zdania, wykorzystujące wszystkie litery (głoski bądź sylaby) materiału wyjściowego.

Najprostszy anagram to poukładanie liter w odwrotnej kolejności, czyli palindrom, np. *kebab – babek*, *keson – nosek*, *arak – kara*, *bark – krab*, *bard – drab*, *mors – srom*. Przykładem jednego z prostych przestawień jest zamiana sylab w wyrazie *ranty*, dająca anagram *tyran*. Przestawiając pojedyncze litery, możemy otrzymać np. anagram *narty*.

Zadanie 3.1. (0–4)

Do sprawdzania, czy wyrazy są swoimi anagramami, najczęściej wykorzystuje się dwie metody: metodę porównywania przez zliczanie liter lub metodę porównywania posortowanych wyrazów.

3.1.
0–1–2
3–4

Wykorzystując metodę porównywania posortowanych wyrazów, napisz algorytm (np. w postaci listy kroków, w pseudokodzie lub w wybranym języku programowania), który sprawdzi, czy dwa podane wyrazy, złożone z wielkich liter alfabetu angielskiego, są anagramami. Sformułuj specyfikację algorytmu.

Uwaga: W zapisie możesz korzystać tylko z instrukcji sterujących, operatorów arytmetycznych: dodawania, odejmowania, mnożenia, dzielenia, dzielenia całkowitego i reszty z dzielenia, operatorów logicznych, porównań, odwoływania się do pojedynczych elementów tablicy i instrukcji przypisywania lub samodzielnie napisanych funkcji i procedur wykorzystujących powyższe operacje. **Zabronione jest używanie funkcji wbudowanych dostępnych w językach programowania związanych z sortowaniem.**

Miejsce na rozwiązanie: