

```

function quicksort(array)
  if length(array) > 1
    pivot := select any element of array
    left := first index of array
    right := last index of array
    while left ≤ right
      while array[left] < pivot
        left := left + 1
      while array[right] > pivot
        right := right - 1
      if left ≤ right
        swap array[left] with array[right]
        left := left + 1
        right := right - 1
    quicksort(array from first index to right)
    quicksort(array from left to last index)

```

Rysunek 1: Algorytm Quick Sort

6. Sortowanie - dobór funkcji do porównywania elementów

Program 6.1: Algorytm Quick Sort

Szablon programu należy uzupełnić o definicję funkcji `quick_sort(...)`, która realizuje algorytm sortowania przedstawiony na rysunku 1. Algorytm Quick Sort: <https://en.wikipedia.org/wiki/Quicksort>. Implementację Quick Sort można zrealizować w sposób klasyczny sugerowany w algorytmie lub za pomocą wskaźników.

- **Wejście**
1
- **Wyjście**
Posortowane rosnąco wartości zadane w kodzie.
- **Przykład:**
Wejście: 1
Wyjście: -31 0 1 2 2 4 65 83 99 782

Program 6.2: Sortowanie słów

Szablon programu należy uzupełnić o definicję funkcji `sort_words(...)`, sortuje słowa alfabetycznie. Realizując zadania od 6.2 do 6.4, można użyć funkcji bibliotecznych związanych z funkcją `qsort()`. Zestaw słów do sortowania jest już zadany w kodzie.

- **Wejście**
2

- **Wyjście**
Posortowane alfabetycznie słowa zadane w kodzie.
- **Przykład:**
Wejście: 2
Wyjście: ant bat bear cat dog frog leopard man monkey rabbit

Program 6.3: Sortowanie leksykograficzne struktur

Szablon programu należy uzupełnić o definicję funkcji `struct_lexi_sort(...)`, sortuje leksykograficznie strukturę według wybranego jej elementu. Problem wiąże się z pojęciem uporządkowania leksykograficznego: https://en.wikipedia.org/wiki/Lexicographic_order

- **Wejście**
3
- **Wyjście**
Posortowana leksykograficznie struktura zadana w kodzie.
- **Przykład:**
Wejście: 3
Wyjście: BAT nietoperz Man człowiek ant mrowka bear niedzwiedz cat kot dog pies frog zaba leopard lampart monkey małpa rabbit krolik

Zadanie 6.4. Złożone wymagania dotyczące sortowania

Szablon programu należy uzupełnić o definicję funkcji `struct_name_sort(...)`, Która sortuje bazę zawierającą Imię Nazwisko i Wiek w ten sposób, że na początek listy wstawia kobiety sortując je po wieku, a następnie mężczyzn sortując ich po nazwiskach.

- **Wejście**
4
- **Wyjście**
Posortowane dane zgodnie z opisanymi powyżej wymaganiami.
- **Przykład:**
Wejście:
4
Wyjście:
Anna Stone 15 Katarzyna Michalak 20 Kamila Munk 25 Anastazja Kamiska 28 Małgorzata Kowal 33 Ernest Goja 27 Emil Kant 22 Stefan Kowal 36 Kamil Mist 29 Tomasz Nod 18 Karol Wodecki 27