

1. Sortowanie

1 Sortowanie struktur

Szablon programu należy uzupełnić o definicję trzech funkcji:

1. `cmp_person()` – komparator, porównujący struktury `Person` najpierw według wieku (malejąco); jeżeli wiek jest ten sam to według imienia i wreszcie nazwiska
2. `read_person_array()` – czytaj dane do tablicy struktur `Person` (wiek, imię, nazwisko); funkcja zwraca liczbę przeczytanych struktur
3. `print_person_array()` – wydrukuj tablicę osób (w kolejności wiek, imię, nazwisko)

- **Wejście**

```
1
person 1
person 1
...
```

- **Wyjście**

Posortowana (według kryteriów jak wyżej) lista osób – struktur `Person`

- **Przykład:**

Wejście:

```
2
1
23 Dominik Adamczyk
27 Natalia Adamiak
23 Dominik Bartczak
22 Szymon Bednorz
24 Marcin Bereta
```

Wyjście:

```
27 Natalia Adamiak
24 Marcin Bereta
23 Dominik Adamczyk
23 Dominik Bartczak
22 Szymon Bednorz
```

2 Sortowanie linii według zadanych kryteriów

Szablon programu należy uzupełnić o definicję trzech funkcji:

1. `cmp_lines()` – komparator, porównujący linie tekstu zawierające wiek, imię, nazwisko w ten sposób, że na początek listy wstawia kobiety sortując je po wieku, a następnie mężczyzn sortując ich po nazwiskach. Wskazówka: imiona kobiet kończą się literą 'a'
2. `read_lines()` – czytaj linie tekstu do tablicy znakowej; każda linia zawiera wiek, imię, nazwisko; funkcja zwraca liczbę przeczytanych linii
3. `print_lines()` – wydrukuj listę osób

- **Wejście**

```
2
line 1
line 2
...
```

- **Wyjście**

Linie posortowane według powyższych kryteriów

- **Przykład:**

Wejście:

```
2
23 Dominik Adamczyk
27 Natalia Adamiak
24 Marcin Chudy
29 Anna Cichocka
22 Natalia Deyna
24 Marcin Bereta
```

Wyjście:

```
22 Natalia Deyna
27 Natalia Adamiak
29 Anna Cichocka
23 Dominik Adamczyk
24 Marcin Bereta
24 Marcin Chudy
```