

Zestaw 1B

Ocena:

ocena 3.0	Ocena 4.0	Ocena 5.0
Zadanie 1 lub Zadanie 2a	Zadanie 2a, Zadanie 2b	Zadanie 2 (a i b), Zadanie3

Zadanie 1:

Napisz funkcję sortującą niemalejąco N liczb całkowitych.

Należy zastosować algorytm sortowania przez proste wybieranie (inaczej przez selekcję). Implementacja optymalna.

Zadanie 2:

Mając dane N elementów, z których każdy zawiera następujące informacje:

- nazwę producenta,
- nazwę produktu.

a) Napisz funkcję wyszukującą wszystkie produkty podanego producenta (nazwa producenta podawana z klawiatury). Implementacja optymalna.

b) Napisz funkcję sortującą produkty alfabetycznie według ich nazwy. Należy zastosować algorytm sortowania przez proste wstawianie. Implementacja optymalna.

Zadanie 3:

Napisz funkcję sortującą elementy z Zadania 2 jednocześnie według dwóch kryteriów:

- alfabetycznie według nazwy produktu (kryterium podstawowe),
- alfabetycznie według nazwy producenta (kryterium dodatkowe).

Należy zastosować prosty algorytm sortowania, inny niż w Zadaniu 2. Implementacja optymalna.

Przykładowe dane do Zadania 2 i 3:

ePaper	teczkaA4
ABC	notesA560
ePaper	zeszyt100l
Goose	pioro
Hen	pioro
PP	notesA560
ABC	zeszyt160k
ABC	zeszyt160l
PP	teczkaA4
Goose	pioro

Wzorzec zawartości pliku programu:

```
#include <cstdlib>
#include <iostream>

using namespace std;

void zadanie1(){
    int const N = 10;
    int dane[N] = {7,3,5,1,9,3,8,2,6,4};
}

void zadanie2a(){
    int const N = 10;
    struct Produkt {
    };
    Produkt dane[N]={}; //tablicę należy zainicjalizować danymi podanymi w zadaniu
}

void zadanie2b(){
    int const N = 10;
    struct Produkt {
    };
    Produkt dane[N]={}; //tablicę należy zainicjalizować danymi podanymi w zadaniu
}

void zadanie3(){
    int const N = 10;
    struct Produkt {
    };
    Produkt dane[N]={}; //tablicę należy zainicjalizować danymi podanymi w zadaniu
}

int main(int argc, char *argv[])
{
    zadanie1();
    zadanie2a();
    zadanie2b();
    zadanie3();

    system("PAUSE");
    return EXIT_SUCCESS;
}
```