

Zestaw 1D

Zadanie 1:

4,5pkt

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

- nazwę kontynentu,
- nazwę kraju,
- powierzchnię kraju.

Napisz funkcję wyszukującą wszystkie kraje położone na kontynencie, którego nazwa rozpoczyna się od podanej litery i których powierzchnia jest większa od podanej wartości (nazwa kontynentu oraz powierzchnia podawane z klawiatury). Implementacja optymalna.

Zadanie 2:

5,5pkt

Napisz funkcję sortującą, jednocześnie, kraje alfabetycznie według nazwy kontynentu (kryterium podstawowe) i alfabetycznie według nazwy kraju (kryterium dodatkowe). Należy zastosować algorytm sortowania przez proste wybieranie. Implementacja optymalna.

Przykładowe dane do Zadania 2 i 3:

Europa	Polska	312.685;
Azja	Chiny	9597.000;
Afryka	Egipt	1001.450;
AmerykaPl	Meksyk	1972.550;
AmerykaPn	Kanada	9976.140;
Azja	Indie	3287.262;
Azja	Wietnam	331.688;
Europa	Norwegia	324.220;
AmerykaPl	Kolumbia	1141.748;
Europa	Niemcy	357.021;

Wzorzec zawartości pliku programu:

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

using namespace std;

int const N = 10;
struct Kraj {
};
Kraj dane[N]={}; //tablicę należy zainicjalizować danymi podanymi w zadaniu

void zadanie1(){

}

void zadanie2(){

}

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

}

int main(int argc, char *argv[])
{
cout<<"Zadanie 2"<<endl;
zadanie1();
cout<<"Zadanie 2"<<endl;
zadanie2();

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