

```
1 //deklaracja potrzebnych bibliotek
2 #include <iostream>
3 #include <fstream>
4 #include <sstream>
5 #include <vector>
6 #include <string>
7 #include <chrono>
8 using namespace std;
9 //tworzenie funkcji szukającej powtarzającego się elementu
10 int pow(vector<int> tablica) {
11     for (int i = 0; i <= tablica.size(); i++) { //funkcje szukające
12         powtarzającego się elemetu
13         for (int j = 0; j <= tablica.size(); j++) {
14             if (i == j) { break; }
15             if (tablica[i] == tablica[j]) { //jesli znalezlismy
16                 powtarzajacy element to go zwracamy
17                 return tablica[i];
18             }
19         }
20     }
21 }
22 vector<int> pobranie_danych(string nazwa) { // funkcja pobierajaca dane z
23     pliku do wektora
24     ifstream indata(nazwa); //otwarcie pliku
25     if (!indata.is_open()) {
26         cout << "Blad otwarcia pliku " << nazwa << endl; // w razie
27         bledu otwarcia pliki
28         return {};
29     }
30     int tmp;
31     vector<int> data;
32     while (!indata.fail() && !indata.eof()) // do konca danych
33     {
34         indata >> tmp;
35         data.push_back(tmp);
36     }
37     indata.close(); //zamkniecie pliku
38     return data;
39 }
40 void zapisanie_danych(string nazwa, int pow, double czas) { // funkcja
41     zapisujaca dane z wektora do pliku
42     ofstream do_pliku(nazwa);
43     do_pliku << "Powtarzajacy element to " << pow << endl; // zapisanie
44     elementow do pliku
45     do_pliku << "Czas wykonywania programu: " << czas << " sekundy";
46 }
```

```
47     vector<int> dane = pobranie_danych("dane.txt");
48     cout << "Powtarzający element to " << pow(dane) << endl;
49     auto koniec = chrono::high_resolution_clock::now(); //koniec pomiaru ↗
        czasu
50     auto miara_czasu = chrono::duration_cast<chrono::nanoseconds>(koniec ↗
        - start);
51     double czas = miara_czasu.count() * 1e-9;
52     zapisanie_danych("zapis.txt", pow(dane), czas);
53
54     return 0;
55 }
56
57 // Uruchomienie programu: Ctrl + F5 lub menu Debugowanie > Uruchom bez ↗
    debugowania
58 // Debugowanie programu: F5 lub menu Debugowanie > Rozpocznij ↗
    debugowanie
59
60 // Porady dotyczące rozpoczynania pracy:
61 // 1. Użyj okna Eksploratora rozwiązań, aby dodać pliki i zarządzać ↗
    nimi
62 // 2. Użyj okna programu Team Explorer, aby nawiązać połączenie z ↗
    kontrolą źródła
63 // 3. Użyj okna Dane wyjściowe, aby sprawdzić dane wyjściowe ↗
    kompilacji i inne komunikaty
64 // 4. Użyj okna Lista błędów, aby zobaczyć błędy
65 // 5. Wybierz pozycję Projekt > Dodaj nowy element, aby utworzyć nowe ↗
    pliki kodu, lub wybierz pozycję Projekt > Dodaj istniejący element, ↗
    aby dodać istniejące pliku kodu do projektu
66 // 6. Aby w przyszłości ponownie otworzyć ten projekt, przejdź do ↗
    pozycji Plik > Otwórz > Projekt i wybierz plik sln
67
```