

C++ - Wyszukiwanie w tablicy

© Copyright by 3bird Projects 2022, <http://edukacja.3bird.pl>

Ogólne

Dwie główne metody wyszukiwania elementów w tablicy to **find()** i **count()**. W obu metodach możemy wykorzystać instrukcję warunkową IF lub pętlę WHILE.

Uwaga: Nigdy nie wolno kopiować kodu z PDF-a, gdyż zawiera on niewidoczne znaki końca linii i tzw. twarde odstępy. Kod należy przepisać ze zrozumieniem.

Kod - find() → if

```
#include <iostream>
#include <algorithm> // Potrzebne do find()
using namespace std;

int main(void) {
    string tablicaOdpowiedziTak[] = {"Tak", "tak", "T", "t", "Yes", "yes", "y", "Y", "taa", "yee",
    "ta", "Ta"};
    string szukane;

    cout << "Wpisz szukane słowo (Tak / Nie): ";
    // Pierwszy argument przekazuje wartość drugiemu (także naciśnięcie ENTER):
    getline(cin, szukane);
    // Find porównuje każdy z elementów tablicy do szukanej wartości i zwraca "1" lub "0".
    // Begin = pierwszy element pierwszej przeszukiwanej tablicy;
    // End = ostatni element ostatniej przeszukiwanej tablicy (tu jest akurat tą samą).
    // Jeśli znajdzie, zatrzymuje się (znaleziony element nie jest tożsamy z ostatnim, więc całość
    // jest prawdą). Jeśli nie znajdzie i dojdzie do końca, wskaże na ostatni element (zwróci go), a
    // ostatni sprawdzany element będzie tożsamy z ostatnim elementem tablicy, co znaczy, że
    // nie znalazł (całość więc zwróci 0).
    // Find() - jeśli jest w stanie bezproblemowo przeszukać wszystkie elementy tablicy - zawsze
    // zwraca 1 bez względu na to, czy znajdzie czy nie (to jest tylko informacja, że udało się
    // przeszukanie). Dopiero porównanie z ostatnim elementem w tablicy daje odpowiedź, czy
    // znaleziono.
    bool jestObecne = find(begin(tablicaOdpowiedziTak), end(tablicaOdpowiedziTak), szukane)
    != end(tablicaOdpowiedziTak);

    if (jestObecne) {
        cout << "Wyrażenie JEST w tablicy!" << endl ;
    }
    else {
        cout << "Wyrażenie NIE jest obecne w tablicy!" << endl;
    }
    cout << "\n\nNaciśnij ENTER, aby zakończyć...";
    system("pause > nul");
}
```

Kod - find() → while

```
#include <iostream>
#include <algorithm> // Potrzebne do find()
using namespace std;

int main(void) {
    const string tablicaOdpowiedziTwierdzacych[] = {"Tak", "tak", "T", "t", "Yes", "yes", "y", "Y",
    "taa", "yee", "ta", "Ta"};
    string szukane;

    while (true) {
        cout << "Czy chcesz kontynuować (Tak/Nie): ";
        // Pierwszy argument przekazuje wartość drugiemu (także naciśnięcie ENTER):
        getline(cin, szukane);
        // Find porównuje każdy z elementów tablicy do szukanej wartości i zwraca "1" lub "0".
        // Begin = pierwszy element pierwszej przeszukiwanej tablicy;
        // End = ostatni element ostatniej przeszukiwanej tablicy (tu jest akurat tą samą).
        // Jeśli znajdzie, zatrzymuje się (znaleziony element nie jest tożsamy z ostatnim, więc
        // całość jest prawdą). Jeśli nie znajdzie i dojdzie do końca, wskaże na ostatni element
        // (zwróci go), a ostatni sprawdzany element będzie tożsamy z ostatnim elementem
        // tablicy, co znaczy, że nie znalazł (całość więc zwróci 0).
        // Find() - jeśli jest w stanie bezproblemowo przeszukać wszystkie elementy tablicy -
        // zawsze zwraca 1 bez względu na to, czy znajdzie czy nie (to jest tylko informacja, że
        // udało się przeszukanie). Dopiero porównanie z ostatnim elementem w tablicy daje
        // odpowiedź, czy znaleziono.
        bool jestObecne = find(begin(tablicaOdpowiedziTwierdzacych),
        end(tablicaOdpowiedziTwierdzacych), szukane) != end(tablicaOdpowiedziTwierdzacych);
        if (jestObecne == 0) {
            break;
        }
    }
    cout << "\n\nNaciśnij ENTER, aby zakończyć...";
    system("pause > nul");
}
```

Kod - count() → if

```
#include <iostream>
#include <string> // Potrzebne do stoi()
#include <algorithm> // Potrzebne do count()

using namespace std;

int main(void) {
    const int tablicaLiczba[] = {20, 43, 36, 22, 14, 51};
    int szukanaLiczbaInt; // Liczba jako integer
    string szukanaLiczbaStr; // Liczba jako string
```

```

cout << "Wpisz szukaną dodatnią liczbę całkowitą: " ;
// Pierwszy argument przekazuje wartość drugiemu (także naciśnięcie ENTER),
// dlatego właśnie używamy tego, z nie "cin > zmienna":
getline(cin, szukanaLiczbaStr);
// Zamieniamy liczbę jako string na liczbę jako integer:
szukanaLiczbaInt = stoi(szukanaLiczbaStr);
// Count() zwraca ile razy dany element występuje w tablicy:
bool znaleziona = count(begin(tablicaLiczb), end(tablicaLiczb), szukanaLiczbaInt) > 0;

if (znaleziona) {
    cout << "\n\nLiczba została znaleziona w tablicy!\a" << endl;
}
else {
    cout << "\n\nLiczba NIE została znaleziona w tablicy." << endl;
}
cout << "\n\nNaciśnij ENTER, aby zamknąć..." << endl;
system("pause > nul");
}

```

Kod - count() → while

```

#include <iostream>
#include <string> // Potrzebne do stoi()
#include <algorithm> // Potrzebne do count()

using namespace std;

int main(void) {
    const int tablicaLiczb[] = {20, 43, 36, 22, 14, 51};
    int szukanaLiczbaInt; // Liczba jako integer
    string szukanaLiczbaStr; // Liczba jako string

    while (true) {
        cout << "Wpisz szukaną dodatnią liczbę całkowitą: " ;
        // Pierwszy argument przekazuje wartość drugiemu (także naciśnięcie ENTER),
        // dlatego właśnie używamy tego, z nie "cin > zmienna":
        getline(cin, szukanaLiczbaStr);
        // Zamieniamy liczbę jako string na liczbę jako integer:
        szukanaLiczbaInt = stoi(szukanaLiczbaStr);
        // Count() zwraca ile razy dany element występuje w tablicy:
        bool znaleziona = count(begin(tablicaLiczb), end(tablicaLiczb), szukanaLiczbaInt) > 0;

        if (znaleziona) {
            cout << "\n\nLiczba została znaleziona w tablicy!\a" << endl;
        }
        else {

```

```
    cout << "\n\nLiczba NIE została znaleziona w tablicy." << endl;
}
cout << "Kontynuować? (Tak / Nie): ";
string kontynuacja;
getline(cin, kontynuacja);
if (kontynuacja != "Tak") {
    break;
}
}
cout << "\n\nNaciśnij ENTER, aby zamknąć..." << endl;
system("pause > nul");
}
```

Ostatnia aktualizacja: 26 grudnia 2022.