

C++ - TABLICE

Każda linia kodu z wyjaśnieniem w komentarzu

Opracowanie na podstawie materiałów „Cpp3U-Tablice” oraz „Cpp3U - Tablice - Cwiczenia”. Kod został zachowany zgodnie z materiałem źródłowym, a do każdej linii dopisano komentarz wyjaśniający jej działanie.

CZĘŚĆ I. Wprowadzenie do tablic

Tablica przechowuje wiele wartości tego samego typu. Elementy tablicy mają indeksy, a pierwszy indeks ma wartość 0.

1. Zamiast wielu osobnych zmiennych

```
int a1, a2, a3, a4, a5;           // Tworzymy pięć osobnych zmiennych całkowitych.
int tab[5];                       // Tworzymy jedną tablicę o nazwie tab, która ma miejsce na 5 liczb całkowitych
.
```

2. Indeksy elementów tablicy

```
tab[0];                           // Odwołujemy się do pierwszego elementu tablicy; indeksowanie zaczyna się od 0
.
tab[1];                           // Odwołujemy się do drugiego elementu tablicy.
tab[2];                           // Odwołujemy się do trzeciego elementu tablicy.
```

3. Tworzenie tablicy

```
int tab[5];                       // Deklarujemy tablicę pięciu liczb całkowitych o indeksach od 0 do 4.
```

4. Przypisywanie wartości

```
tab[0] = 10;                      // Do pierwszego elementu tablicy wpisujemy wartość 10.
tab[1] = 20;                      // Do drugiego elementu tablicy wpisujemy wartość 20.
```

5. Wczytywanie danych do tablicy

```
for (int i = 0; i < 5; i++) {     // Pętla przechodzi po indeksach tablicy od 0 do 4.
    cin >> tab[i];                // Wczytujemy liczbę z klawiatury i zapisujemy ją w elemencie o indeksie i.
}                                  // Kończymy blok pętli for.
```

6. Wypisywanie elementów tablicy

```
for (int i = 0; i < 5; i++) {     // Pętla przechodzi kolejno przez wszystkie 5 elementów tablicy.
    cout << tab[i] << endl;       // Wypisujemy aktualny element tablicy i przechodzimy do nowej linii.
}                                  // Kończymy blok pętli for.
```

CZĘŚĆ II. Przykłady z materiału

Przykład 1 - wczytaj i wypisz 5 liczb

```
#include <iostream>           // Dołączamy bibliotekę wejścia i wyjścia potrzebną do cin, cout i endl.
using namespace std;         // Umożliwiamy używanie cin, cout i endl bez przedrostka std::.
int main() {                 // Rozpoczynamy główną funkcję programu.
    int tab[5];              // Tworzymy tablicę pięciu liczb całkowitych.
    for (int i = 0; i < 5; i++) { // Przechodzimy po indeksach od 0 do 4.
        cin >> tab[i];       // Wczytujemy kolejną liczbę do elementu tablicy o indeksie i.
    }                        // Kończymy pętlę wczytującą dane.
    for (int i = 0; i < 5; i++) { // Ponownie przechodzimy po wszystkich elementach tablicy.
        cout << tab[i] << endl; // Wypisujemy wartość elementu tab[i].
    }                        // Kończymy pętlę wypisującą dane.
    return 0;               // Kończymy program poprawnie.
}                           // Kończymy funkcję main.
```

Przykład 2 - suma elementów

```
#include <iostream>           // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std;         // Używamy standardowej przestrzeni nazw.
int main() {                 // Rozpoczynamy główną funkcję programu.
    int tab[5];              // Tworzymy tablicę pięciu liczb całkowitych.
    int suma = 0;            // Tworzymy zmienną suma i ustawiamy jej wartość początkową na 0.
    for (int i = 0; i < 5; i++) { // Przechodzimy przez wszystkie elementy tablicy.
        cin >> tab[i];       // Wczytujemy kolejną liczbę do tablicy.
        suma += tab[i];      // Dodajemy wczytaną wartość do zmiennej suma.
    }                        // Kończymy pętlę wczytującą i sumującą.
    cout << suma << endl;    // Wypisujemy sumę wszystkich pięciu elementów.
    return 0;               // Kończymy program poprawnie.
}                           // Kończymy funkcję main.
```

Przykład 3 - największy element

```
#include <iostream>           // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std;         // Używamy standardowej przestrzeni nazw.
int main() {                 // Rozpoczynamy funkcję main.
    int tab[5];              // Tworzymy tablicę pięciu liczb całkowitych.
    for (int i = 0; i < 5; i++) { // Przechodzimy po indeksach od 0 do 4.
        cin >> tab[i];       // Wczytujemy kolejne wartości do tablicy.
    }                        // Kończymy pętlę wczytującą.
    int maxL = tab[0];        // Przyjmujemy pierwszy element jako początkowo największy.
    for (int i = 1; i < 5; i++) { // Sprawdzamy kolejne elementy, zaczynając od indeksu 1.
        if (tab[i] > maxL) {   // Sprawdzamy, czy aktualny element jest większy od dotychczasowego maksimum.
            maxL = tab[i];     // Jeśli tak, zapisujemy aktualny element jako nowe maksimum.
        }                    // Kończymy instrukcję warunkową if.
    }                        // Kończymy pętlę wyszukującą maksimum.
    cout << maxL << endl;    // Wypisujemy największą znalezioną wartość.
    return 0;               // Kończymy program poprawnie.
}                           // Kończymy funkcję main.
```

CZĘŚĆ III. Ćwiczenia - 20 zadań

Zadanie 1 - wczytaj 10 liczb do tablicy i wypisz je

```
#include <iostream>                // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std;              // Umożliwiamy używanie cin, cout i endl bez std::.
int main() {                      // Rozpoczynamy główną funkcję programu.
    int tab[10];                 // Tworzymy tablicę 10 liczb całkowitych.
    for (int i = 0; i < 10; i++) cin >> tab[i]; // Wczytujemy 10 liczb do kolejnych elementów tablicy.
    for (int i = 0; i < 10; i++) cout << tab[i] << endl; // Wypisujemy wszystkie elementy tablicy.
    return 0;                   // Kończymy program poprawnie.
}                                // Kończymy funkcję main.
```

Zadanie 2 - oblicz sumę 10 liczb

```
#include <iostream>                // Dołączamy bibliotekę iostream.
using namespace std;              // Używamy standardowej przestrzeni nazw.
int main() {                      // Rozpoczynamy program.
    int tab[10], suma = 0;        // Tworzymy tablicę 10 liczb oraz zmienną suma o wartości początkowej 0.
    for (int i = 0; i < 10; i++) { // Przechodzimy po wszystkich 10 elementach tablicy.
        cin >> tab[i];           // Wczytujemy kolejną liczbę do tablicy.
        suma += tab[i];          // Dodajemy aktualną wartość tablicy do sumy.
    }                             // Kończymy pętlę.
    cout << suma << endl;         // Wypisujemy obliczoną sumę.
    return 0;                    // Kończymy program poprawnie.
}                                // Kończymy funkcję main.
```

Zadanie 3 - znajdź największą z 10 liczb

```
#include <iostream>                // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std;              // Używamy standardowej przestrzeni nazw.
int main() {                      // Rozpoczynamy program.
    int tab[10];                 // Tworzymy tablicę 10 liczb całkowitych.
    for (int i = 0; i < 10; i++) cin >> tab[i]; // Wczytujemy wszystkie wartości do tablicy.
    int maxL = tab[0];           // Przyjmujemy pierwszy element jako początkowe maksimum.
    for (int i = 1; i < 10; i++) { // Sprawdzamy elementy od indeksu 1 do 9.
        if (tab[i] > maxL) maxL = tab[i]; // Jeśli aktualny element jest większy, zapisujemy go jako nowe maksimum.
    }                             // Kończymy pętlę wyszukiującą.
    cout << maxL << endl;         // Wypisujemy największą wartość.
    return 0;                    // Kończymy program poprawnie.
}                                // Kończymy funkcję main.
```

Zadanie 4 - znajdź najmniejszą z 10 liczb

```
#include <iostream>                // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std;              // Używamy standardowej przestrzeni nazw.
int main() {                      // Rozpoczynamy program.
    int tab[10];                 // Tworzymy tablicę 10 liczb całkowitych.
    for (int i = 0; i < 10; i++) cin >> tab[i]; // Wczytujemy 10 liczb do tablicy.
    int minL = tab[0];           // Przyjmujemy pierwszy element jako początkowe minimum.
    for (int i = 1; i < 10; i++) { // Sprawdzamy pozostałe elementy tablicy.
        if (tab[i] < minL) minL = tab[i]; // Jeśli aktualny element jest mniejszy, zapisujemy go jako nowe minimum.
    }                             // Kończymy pętlę wyszukiującą.
    cout << minL << endl;         // Wypisujemy najmniejszą wartość.
    return 0;                    // Kończymy program poprawnie.
}                                // Kończymy funkcję main.
```

Zadanie 5 - policz liczby parzyste

```
#include <iostream>                // Dołączamy bibliotekę iostream.
using namespace std;              // Używamy standardowej przestrzeni nazw.
int main() {                      // Rozpoczynamy program.
    int tab[10], licznik = 0;     // Tworzymy tablicę oraz licznik liczb parzystych ustawiony na 0.
    for (int i = 0; i < 10; i++) { // Przechodzimy po wszystkich elementach tablicy.
        cin >> tab[i];           // Wczytujemy aktualną liczbę.
        if (tab[i] % 2 == 0) licznik++; // Jeśli liczba jest parzysta, zwiększamy licznik o 1.
    }                             // Kończymy pętlę.
    cout << licznik << endl;      // Wypisujemy liczbę elementów parzystych.
    return 0;                    // Kończymy program poprawnie.
}                                // Kończymy funkcję main.
```

Zadanie 6 - wypisz tylko liczby dodatnie

```
#include <iostream> // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int tab[10]; // Tworzymy tablicę 10 liczb całkowitych.
    for (int i = 0; i < 10; i++) cin >> tab[i]; // Wczytujemy wszystkie wartości do tablicy.
    for (int i = 0; i < 10; i++) { // Przechodzimy ponownie przez wszystkie elementy.
        if (tab[i] > 0) cout << tab[i] << endl; // Jeśli element jest dodatni, wypisujemy go.
    } // Kończymy pętlę wypisującą.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Zadanie 7 - oblicz średnią 10 liczb

```
#include <iostream> // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int tab[10]; // Tworzymy tablicę 10 liczb całkowitych.
    double suma = 0; // Tworzymy sumę typu double, aby średnia mogła mieć część ułamkową.
    for (int i = 0; i < 10; i++) { // Przechodzimy przez 10 elementów.
        cin >> tab[i]; // Wczytujemy kolejną wartość.
        suma += tab[i]; // Dodajemy ją do sumy.
    } // Kończymy pętlę.
    cout << suma / 10 << endl; // Dzielimy sumę przez 10 i wypisujemy średnią.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Zadanie 8 - wypisz tablicę w odwrotnej kolejności

```
#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int tab[10]; // Tworzymy tablicę 10 liczb.
    for (int i = 0; i < 10; i++) cin >> tab[i]; // Wczytujemy 10 wartości do tablicy.
    for (int i = 9; i >= 0; i--) { // Zaczynamy od ostatniego indeksu 9 i cofamy się do 0.
        cout << tab[i] << endl; // Wypisujemy aktualny element tablicy.
    } // Kończymy pętlę.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Zadanie 9 - znajdź indeks największej liczby

```
#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int tab[10]; // Tworzymy tablicę 10 liczb.
    for (int i = 0; i < 10; i++) cin >> tab[i]; // Wczytujemy wartości do tablicy.
    int maxL = tab[0]; // Ustawiamy pierwszy element jako początkowe maksimum.
    int indeks = 0; // Zapamiętujemy jego indeks równy 0.
    for (int i = 1; i < 10; i++) { // Sprawdzamy pozostałe elementy.
        if (tab[i] > maxL) { // Jeśli aktualny element jest większy od maksimum...
            maxL = tab[i]; // ...zapisujemy jego wartość jako nowe maksimum.
            indeks = i; // ...oraz zapamiętujemy jego indeks.
        } // Kończymy instrukcję if.
    } // Kończymy pętlę.
    cout << indeks << endl; // Wypisujemy indeks największego elementu.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Zadanie 10 - sprawdź, czy występuje liczba 0

```
#include <iostream> // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int tab[10]; // Tworzymy tablicę 10 liczb całkowitych.
    bool znaleziono = false; // Tworzymy zmienną logiczną informującą, czy znaleziono 0.
    for (int i = 0; i < 10; i++) { // Przechodzimy przez wszystkie elementy.
        cin >> tab[i]; // Wczytujemy aktualną liczbę.
        if (tab[i] == 0) znaleziono = true; // Jeśli liczba jest równa 0, ustawiamy flagę na true.
    } // Kończymy pętlę.
    if (znaleziono) cout << "TAK\n"; // Jeśli flaga ma wartość true, wypisujemy TAK.
    else cout << "NIE\n"; // W przeciwnym razie wypisujemy NIE.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Zadanie 11 - zamień liczby ujemne na 0

```
#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int tab[10]; // Tworzymy tablicę 10 liczb.
    for (int i = 0; i < 10; i++) cin >> tab[i]; // Wczytujemy wszystkie liczby.
    for (int i = 0; i < 10; i++) { // Przechodzimy po wszystkich elementach.
        if (tab[i] < 0) tab[i] = 0; // Jeśli element jest ujemny, zastępujemy go zerem.
    } // Kończymy pętlę modyfikującą.
    for (int i = 0; i < 10; i++) cout << tab[i] << endl; // Wypisujemy zmodyfikowaną tablicę.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Zadanie 12 - policz liczby większe od 10

```
#include <iostream> // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int tab[10], licznik = 0; // Tworzymy tablicę oraz licznik elementów większych od 10.
    for (int i = 0; i < 10; i++) { // Przechodzimy przez wszystkie elementy.
        cin >> tab[i]; // Wczytujemy kolejny element.
        if (tab[i] > 10) licznik++; // Jeśli element jest większy od 10, zwiększamy licznik.
    } // Kończymy pętlę.
    cout << licznik << endl; // Wypisujemy liczbę elementów większych od 10.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Zadanie 13 - wypisz liczby podzielne przez 3

```
#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int tab[10]; // Tworzymy tablicę 10 liczb.
    for (int i = 0; i < 10; i++) cin >> tab[i]; // Wczytujemy 10 elementów.
    for (int i = 0; i < 10; i++) { // Przechodzimy ponownie po wszystkich elementach.
        if (tab[i] % 3 == 0) cout << tab[i] << endl; // Jeśli liczba dzieli się przez 3 bez reszty, wypisujemy ją.
    } // Kończymy pętlę.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Zadanie 14 - znajdź drugą największą liczbę

```
#include <iostream> // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int tab[10]; // Tworzymy tablicę 10 liczb.
    for (int i = 0; i < 10; i++) cin >> tab[i]; // Wczytujemy wartości do tablicy.
    int max1 = tab[0], max2 = tab[0]; // Zgodnie z materiałem oba maksima początkowo ustawiamy na pierwszy element.
    for (int i = 1; i < 10; i++) { // Przechodzimy przez pozostałe elementy.
        if (tab[i] > max1) { // Jeśli aktualny element jest większy od największego...
            max2 = max1; // ...dotychczasowe maksimum przesuwamy na pozycję drugiego maksimum.
            max1 = tab[i]; // ...a aktualny element ustawiamy jako nowe maksimum.
        } else if (tab[i] > max2 && tab[i] != max1) { // Jeśli nie jest największy, ale jest większy od max2 i różny od max1...
            max2 = tab[i]; // ...ustawiamy go jako drugą największą wartość.
        } // Kończymy instrukcję if/else if.
    } // Kończymy pętlę.
    cout << max2 << endl; // Wypisujemy wartość drugiego maksimum zgodnie z algorytmem z materiału.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Zadanie 15 - policz wystąpienia liczby podanej przez użytkownika

```
#include <iostream> // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int tab[10], x, licznik = 0; // Tworzymy tablicę, zmienną szukaną x oraz licznik wystąpień.
    for (int i = 0; i < 10; i++) cin >> tab[i]; // Wczytujemy 10 elementów tablicy.
    cin >> x; // Wczytujemy liczbę, której wystąpienia chcemy policzyć.
    for (int i = 0; i < 10; i++) { // Przechodzimy przez wszystkie elementy tablicy.
        if (tab[i] == x) licznik++; // Jeśli element jest równy x, zwiększamy licznik.
    } // Kończymy pętlę.
    cout << licznik << endl; // Wypisujemy liczbę wystąpień wartości x.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Zadanie 16 - sprawdź, czy tablica jest rosnąca

```
#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int tab[10]; // Tworzymy tablicę 10 liczb.
    bool rosnaca = true; // Zakładamy na początku, że tablica jest rosnąca.
    for (int i = 0; i < 10; i++) cin >> tab[i]; // Wczytujemy 10 elementów.
    for (int i = 1; i < 10; i++) { // Porównujemy każdy element z elementem poprzednim.
        if (tab[i] < tab[i - 1]) rosnaca = false; // Jeśli kolejny element jest mniejszy, tablica nie jest rosnąca.
    } // Kończymy pętlę sprawdzającą.
    if (rosnaca) cout << "TAK\n"; // Jeśli flaga nadal jest true, wypisujemy TAK.
    else cout << "NIE\n"; // W przeciwnym razie wypisujemy NIE.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Zadanie 17 - zamień miejscami pierwszy i ostatni element

```
#include <iostream> // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int tab[10]; // Tworzymy tablicę 10 liczb.
    for (int i = 0; i < 10; i++) cin >> tab[i]; // Wczytujemy 10 wartości.
    int temp = tab[0]; // Zapisujemy pierwszy element w zmiennej pomocniczej temp.
    tab[0] = tab[9]; // Do pierwszego elementu wpisujemy wartość ostatniego elementu.
    tab[9] = temp; // Do ostatniego elementu wpisujemy zapamiętaną pierwotną wartość pierwszego elementu.
    for (int i = 0; i < 10; i++) cout << tab[i] << endl; // Wypisujemy tablicę po zamianie.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Zadanie 18 - różnica między największą a najmniejszą

```
#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int tab[10]; // Tworzymy tablicę 10 liczb.
    for (int i = 0; i < 10; i++) cin >> tab[i]; // Wczytujemy wszystkie elementy.
    int maxL = tab[0], minL = tab[0]; // Przyjmujemy pierwszy element jako początkowe maksimum i minimum.
    for (int i = 1; i < 10; i++) { // Sprawdzamy pozostałe elementy tablicy.
        if (tab[i] > maxL) maxL = tab[i]; // Jeśli element jest większy, aktualizujemy maksimum.
        if (tab[i] < minL) minL = tab[i]; // Jeśli element jest mniejszy, aktualizujemy minimum.
    } // Kończymy pętlę.
    cout << maxL - minL << endl; // Obliczamy i wypisujemy różnicę maksimum i minimum.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Zadanie 19 - wypisz tylko liczby nieparzyste

```
#include <iostream> // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int tab[10]; // Tworzymy tablicę 10 liczb.
    for (int i = 0; i < 10; i++) cin >> tab[i]; // Wczytujemy wszystkie wartości.
    for (int i = 0; i < 10; i++) { // Przechodzimy przez każdy element tablicy.
        if (tab[i] % 2 != 0) cout << tab[i] << endl; // Jeśli liczba jest nieparzysta, wypisujemy ją.
    } // Kończymy pętlę.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Zadanie 20 - suma tylko liczb parzystych

```
#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int tab[10], suma = 0; // Tworzymy tablicę i zmienną suma ustawioną na 0.
    for (int i = 0; i < 10; i++) { // Przechodzimy przez wszystkie 10 elementów.
        cin >> tab[i]; // Wczytujemy kolejną liczbę.
        if (tab[i] % 2 == 0) suma += tab[i]; // Jeśli liczba jest parzysta, dodajemy ją do sumy.
    } // Kończymy pętlę.
    cout << suma << endl; // Wypisujemy sumę wszystkich liczb parzystych.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```