

C++ - PĘTLE

Kod z wyjaśnieniem każdej linii

Materiały: *Cpp2U-Pętle.pdf* oraz *Cpp2U-Pętle-Cwiczenia.pdf*

W każdym programie każda linia kodu została opisana komentarzem //, tak aby uczeń mógł czytać program krok po kroku.

CZĘŚĆ I. Pętle - materiał wprowadzający

Pętla pozwala powtarzać fragment programu wiele razy. Pętla for jest używana wtedy, gdy z góry wiemy, ile razy chcemy wykonać operację.

1. Pętla for - przykład: wypisywanie liczb 1-5

```
#include <iostream>           // Dołączamy bibliotekę iostream, aby móc używać cout i endl.
using namespace std;         // Pozwala używać nazw cout i endl bez zapisu std::.
int main() {                  // Rozpoczynamy główną funkcję programu.
    for (int i = 1; i <= 5; i++) { // Ustawiamy i=1; powtarzamy, dopóki i<=5; po każdej iteracji zwiększamy i o 1.
        cout << i << endl;      // Wypisujemy aktualną wartość i i przechodzimy do nowej linii.
    }                          // Kończymy blok instrukcji wykonywany w pętli for.
    return 0;                 // Kończymy program poprawnie i zwracamy kod 0.
}                              // Kończymy funkcję main.
```

2. Pętla for - przykład: suma liczb 1-5

```
#include <iostream>           // Dołączamy bibliotekę umożliwiającą wypisywanie danych.
using namespace std;         // Używamy standardowej przestrzeni nazw.
int main() {                  // Rozpoczynamy funkcję główną programu.
    int suma = 0;             // Tworzymy zmienną suma i ustawiamy jej początkową wartość na 0.
    for (int i = 1; i <= 5; i++) { // Przechodzimy kolejno przez liczby od 1 do 5.
        suma = suma + i;      // Dodajemy aktualną wartość i do dotychczasowej sumy.
    }                          // Kończymy blok pętli for.
    cout << suma << endl;     // Wypisujemy obliczoną sumę, czyli 15.
    return 0;                 // Kończymy program poprawnie.
}                              // Kończymy funkcję main.
```

Pętla while działa tak długo, jak długo jej warunek jest prawdziwy. Nie musimy z góry wiedzieć, ile razy zostanie wykonana.

3. Pętla while - przykład: wypisywanie liczb 1-5

```
#include <iostream>           // Dołączamy bibliotekę potrzebną do cout i endl.
using namespace std;         // Używamy standardowej przestrzeni nazw.
int main() {                  // Rozpoczynamy funkcję main.
    int i = 1;                // Tworzymy licznik i i ustawiamy jego wartość początkową na 1.
    while (i <= 5) {          // Powtarzamy instrukcje, dopóki i jest mniejsze lub równe 5.
        cout << i << endl;    // Wypisujemy aktualną wartość licznika i.
        i++;                  // Zwiększamy i o 1, aby pętla zbliżała się do zakończenia.
    }                          // Kończymy blok pętli while.
    return 0;                 // Kończymy program poprawnie.
}                              // Kończymy funkcję main.
```

4. Pętla while - przykład: suma liczb 1-5

```
#include <iostream>           // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std;         // Pozwala używać cout bez prefiksu std::.
int main() {                  // Rozpoczynamy funkcję główną.
    int i = 1;                // Ustawiamy licznik i na 1.
    int suma = 0;             // Ustawiamy początkową sumę na 0.
    while (i <= 5) {          // Powtarzamy instrukcje dla i od 1 do 5.
        suma += i;            // Dodajemy aktualne i do zmiennej suma.
        i++;                  // Zwiększamy licznik i o 1.
    }                          // Kończymy blok pętli while.
    cout << suma << endl;     // Wypisujemy wynik sumowania.
    return 0;                 // Kończymy program poprawnie.
```

```
} // Kończymy funkcję main.
```

Pętla do while wykonuje swój blok co najmniej jeden raz, ponieważ warunek jest sprawdzany dopiero po wykonaniu instrukcji.

5. Pętla do while - przykład: wypisywanie liczb 1-5

```
#include <iostream> // Dołączamy bibliotekę potrzebną do cout.
using namespace std; // Korzystamy ze standardowej przestrzeni nazw.
int main() { // Rozpoczynamy funkcję main.
    int i = 1; // Ustawiamy licznik i na 1.
    do { // Rozpoczynamy blok, który wykona się co najmniej jeden raz.
        cout << i << endl; // Wypisujemy aktualną wartość i.
        i++; // Zwiększamy i o 1.
    } while (i <= 5); // Po wykonaniu bloku sprawdzamy, czy i nadal jest <= 5.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

6. Pętla do while - wczytywanie liczby dodatniej

```
#include <iostream> // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std; // Używamy nazw cin i cout bez std::.
int main() { // Rozpoczynamy funkcję główną programu.
    int n; // Tworzymy zmienną całkowitą n.
    do { // Rozpoczynamy pętlę, która wykona się przynajmniej raz.
        cin >> n; // Pobieramy od użytkownika wartość zmiennej n.
    } while (n <= 0); // Powtarzamy wczytywanie, jeśli n jest mniejsze lub równe 0.
    cout << "Podales liczbe dodatnia"; // Informujemy, że użytkownik podał liczbę dodatnią.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Zadanie 1 - wypisz liczby od 1 do 10

```
#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    for (int i = 1; i <= 10; i++) { // Ustawiamy i=1 i zwiększamy je aż do 10 włącznie.
        cout << i << endl; // Wypisujemy aktualną wartość i.
    } // Kończymy blok pętli for.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Zadanie 2 - wypisz liczby od 10 do 1

```
#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    for (int i = 10; i >= 1; i--) { // Zaczynamy od 10 i zmniejszamy i o 1, dopóki i>=1.
        cout << i << endl; // Wypisujemy aktualną wartość i.
    } // Kończymy blok pętli.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Zadanie 3 - suma liczb od 1 do 100

```
#include <iostream> // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int suma = 0; // Tworzymy zmienną suma i ustawiamy ją na 0.
    for (int i = 1; i <= 100; i++) { // Przechodzimy kolejno przez liczby od 1 do 100.
        suma += i; // Dodajemy aktualną wartość i do sumy.
    } // Kończymy blok pętli.
    cout << suma << endl; // Wypisujemy wynik sumowania.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Zadanie 4 - wypisz liczby od 1 do n

```
#include <iostream> // Dołączamy bibliotekę obsługującą cin i cout.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int n; // Tworzymy zmienną całkowitą n.
    cin >> n; // Wczytujemy wartość n od użytkownika.
    for (int i = 1; i <= n; i++) { // Przechodzimy przez wszystkie liczby od 1 do n.
        cout << i << endl; // Wypisujemy aktualną wartość i.
    }
}
```

```

    }
    return 0;
}
// Kończymy pętlę.
// Kończymy program poprawnie.
// Kończymy funkcję main.

```

Zadanie 5 - kwadraty liczb od 1 do n

```

#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy przestrzeni nazw std.
int main() { // Rozpoczynamy program.
    int n; // Tworzymy zmienną n.
    cin >> n; // Wczytujemy wartość n.
    for (int i = 1; i <= n; i++) { // Przechodzimy przez liczby od 1 do n.
        cout << i * i << endl; // Obliczamy kwadrat i przez mnożenie i*i i wypisujemy wynik.
    } // Kończymy pętlę.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.

```

Zadanie 6 - policz liczby parzyste od 1 do n

```

#include <iostream> // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int n; // Tworzymy zmienną n.
    cin >> n; // Wczytujemy n od użytkownika.
    int licznik = 0; // Tworzymy licznik liczb parzystych i ustawiamy go na 0.
    for (int i = 1; i <= n; i++) { // Sprawdzamy kolejno wszystkie liczby od 1 do n.
        if (i % 2 == 0) { // Sprawdzamy, czy i dzieli się przez 2 bez reszty.
            licznik++; // Jeśli tak, zwiększamy licznik liczb parzystych o 1.
        } // Kończymy blok instrukcji if.
    } // Kończymy pętlę for.
    cout << licznik << endl; // Wypisujemy liczbę znalezionych liczb parzystych.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.

```

Uwaga do zadań 7-9 w materiale wprowadzającym

W źródłowym pliku nagłówki i pokazany kod nie są w pełni zgodne. Poniżej opisuję dokładnie kod widoczny w materiale, bez jego cichego poprawiania.

Zadanie 7 - kod widoczny w materiale (wczytanie 5 liczb i obliczenie sumy)

```

#include <iostream> // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy funkcję main.
    int suma = 0; // Tworzymy sumę i ustawiamy ją początkowo na 0.
    int x; // Tworzymy zmienną x na kolejne wczytywane liczby.
    for (int i = 1; i <= 5; i++) { // Pętlę wykonujemy dokładnie 5 razy.
        cin >> x; // Wczytujemy kolejną liczbę do x.
        suma += x; // Dodajemy wczytaną liczbę do sumy.
    } // Kończymy pętlę for.
    cout << suma << endl; // Wypisujemy sumę pięciu liczb.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.

```

Zadanie 8 - kod widoczny w materiale (tabliczka mnożenia liczby 5)

```

#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    for (int i = 1; i <= 10; i++) { // Przechodzimy przez mnożniki od 1 do 10.
        cout << "5 * " << i << " = " << 5 * i << endl; // Wypisujemy działanie 5*i oraz jego wynik.
    } // Kończymy pętlę.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.

```

Zadanie 9 - tabliczka mnożenia liczby 5

```

#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    for (int i = 1; i <= 10; i++) { // Ustawiamy kolejne mnożniki od 1 do 10.
        cout << "5 * " << i << " = " << 5 * i << endl; // Pokazujemy działanie i wynik mnożenia przez 5.
    } // Kończymy pętlę for.
    return 0; // Kończymy program poprawnie.
}

```

```
} // Kończymy funkcję main.
```

Zadanie 10 - silnia liczby n

```
#include <iostream> // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int n; // Tworzymy zmienną n.
    cin >> n; // Wczytujemy liczbę n od użytkownika.
    int silnia = 1; // Ustawiamy początkową wartość silni na 1.
    for (int i = 1; i <= n; i++) { // Przechodzimy przez liczby od 1 do n.
        silnia *= i; // Mnożymy dotychczasową wartość silni przez i.
    } // Kończymy pętlę for.
    cout << silnia << endl; // Wypisujemy obliczoną silnię.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

CZĘŚĆ II. Ćwiczenia - 20 zadań z pętli

Ćwiczenie 1 - wypisz liczby od 1 do 20

```
#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    for (int i = 1; i <= 20; i++) { // Przechodzimy przez liczby od 1 do 20.
        cout << i << endl; // Wypisujemy każdą liczbę w osobnym wierszu.
    } // Kończymy pętlę.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Ćwiczenie 2 - wypisz liczby od 20 do 1

```
#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    for (int i = 20; i >= 1; i--) { // Zaczynamy od 20 i zmniejszamy i o 1 do wartości 1.
        cout << i << endl; // Wypisujemy aktualną wartość i.
    } // Kończymy pętlę.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Ćwiczenie 3 - liczby parzyste od 1 do 50

```
#include <iostream> // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    for (int i = 1; i <= 50; i++) { // Przechodzimy przez wszystkie liczby od 1 do 50.
        if (i % 2 == 0) { // Sprawdzamy, czy reszta z dzielenia i przez 2 wynosi 0.
            cout << i << endl; // Jeśli liczba jest parzysta, wypisujemy ją.
        } // Kończymy instrukcję if.
    } // Kończymy pętlę for.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Ćwiczenie 4 - liczby nieparzyste od 1 do 50

```
#include <iostream> // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    for (int i = 1; i <= 50; i++) { // Przechodzimy przez liczby od 1 do 50.
        if (i % 2 != 0) { // Sprawdzamy, czy reszta z dzielenia i przez 2 jest różna od 0.
            cout << i << endl; // Jeśli liczba jest nieparzysta, wypisujemy ją.
        } // Kończymy instrukcję if.
    } // Kończymy pętlę.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.
```

Ćwiczenie 5 - suma liczb od 1 do 10

```
#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
```

```

int suma = 0; // Ustawiamy początkową sumę na 0.
for (int i = 1; i <= 10; i++) { // Przechodzimy przez liczby od 1 do 10.
    suma += i; // Dodajemy aktualne i do sumy.
} // Kończymy pętlę.
cout << suma << endl; // Wypisujemy końcową sumę.
return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.

```

Ćwiczenie 6 - suma liczb od 1 do n

```

#include <iostream> // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int n; // Tworzymy zmienną n.
    cin >> n; // Wczytujemy n od użytkownika.
    int suma = 0; // Ustawiamy początkową sumę na 0.
    for (int i = 1; i <= n; i++) { // Przechodzimy przez liczby od 1 do n.
        suma += i; // Dodajemy i do sumy.
    } // Kończymy pętlę.
    cout << suma << endl; // Wypisujemy sumę liczb od 1 do n.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.

```

Ćwiczenie 7 - wypisz liczby od n do 1

```

#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int n; // Tworzymy zmienną n.
    cin >> n; // Wczytujemy n od użytkownika.
    for (int i = n; i >= 1; i--) { // Zaczynamy od n i zmniejszamy i do 1.
        cout << i << endl; // Wypisujemy aktualną wartość i.
    } // Kończymy pętlę.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.

```

Ćwiczenie 8 - liczby podzielne przez 3 od 1 do n

```

#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int n; // Tworzymy zmienną n.
    cin >> n; // Wczytujemy n od użytkownika.
    for (int i = 1; i <= n; i++) { // Przechodzimy przez liczby od 1 do n.
        if (i % 3 == 0) { // Sprawdzamy, czy i dzieli się przez 3 bez reszty.
            cout << i << endl; // Jeśli tak, wypisujemy tę liczbę.
        } // Kończymy instrukcję if.
    } // Kończymy pętlę.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.

```

Ćwiczenie 9 - policz liczby parzyste od 1 do n

```

#include <iostream> // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int n; // Tworzymy zmienną n.
    cin >> n; // Wczytujemy n od użytkownika.
    int licznik = 0; // Ustawiamy licznik liczb parzystych na 0.
    for (int i = 1; i <= n; i++) { // Przechodzimy przez liczby od 1 do n.
        if (i % 2 == 0) { // Sprawdzamy, czy i jest parzyste.
            licznik++; // Zwiększamy licznik, jeśli liczba jest parzysta.
        } // Kończymy instrukcję if.
    } // Kończymy pętlę.
    cout << licznik << endl; // Wypisujemy liczbę liczb parzystych.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.

```

Ćwiczenie 10 - silnia liczby n

```

#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int n; // Tworzymy zmienną n.
    cin >> n; // Wczytujemy n.

```

```

int silnia = 1; // Ustawiamy początkową wartość silni na 1.
for (int i = 1; i <= n; i++) { // Przechodzimy przez wszystkie liczby od 1 do n.
    silnia *= i; // Mnożymy dotychczasowy wynik przez i.
} // Kończymy pętlę.
cout << silnia << endl; // Wypisujemy obliczoną silnię.
return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.

```

Ćwiczenie 11 - wczytaj 5 liczb i oblicz ich sumę

```

#include <iostream> // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int suma = 0; // Ustawiamy początkową sumę na 0.
    int x; // Tworzymy zmienną na kolejne wczytywane liczby.
    for (int i = 1; i <= 5; i++) { // Pętla wykona się dokładnie 5 razy.
        cin >> x; // Wczytujemy kolejną liczbę.
        suma += x; // Dodajemy wczytaną liczbę do sumy.
    } // Kończymy pętlę.
    cout << suma << endl; // Wypisujemy sumę pięciu liczb.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.

```

Ćwiczenie 12 - wczytaj 5 liczb i znajdź największą

```

#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int x, maxL; // Tworzymy zmienną na liczbę x i zmienną maxL na największą wartość.
    cin >> x; // Wczytujemy pierwszą z pięciu liczb.
    maxL = x; // Przyjmujemy pierwszą liczbę jako dotychczas największą.
    for (int i = 2; i <= 5; i++) { // Wczytujemy jeszcze cztery liczby, od pozycji 2 do 5.
        cin >> x; // Wczytujemy kolejną liczbę.
        if (x > maxL) { // Sprawdzamy, czy nowa liczba jest większa od dotychczasowego
            // maksimum.
            maxL = x; // Jeśli tak, zapisujemy ją jako nowe maksimum.
        } // Kończymy instrukcję if.
    } // Kończymy pętlę.
    cout << maxL << endl; // Wypisujemy największą z pięciu liczb.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.

```

Ćwiczenie 13 - wczytaj liczby do 0 i wypisz sumę

```

#include <iostream> // Dołączamy bibliotekę wejścia i wyjścia.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int x; // Tworzymy zmienną na kolejne liczby.
    int suma = 0; // Ustawiamy sumę początkowo na 0.
    while (true) { // Tworzymy pętlę nieskończoną, którą zakończymy instrukcją break.
        cin >> x; // Wczytujemy kolejną liczbę.
        if (x == 0) { // Sprawdzamy, czy użytkownik wpisał 0.
            break; // Jeśli tak, natychmiast wychodzimy z pętli.
        } // Kończymy instrukcję if.
        suma += x; // Dodajemy niezerową liczbę do sumy.
    } // Kończymy blok pętli while.
    cout << suma << endl; // Wypisujemy sumę wszystkich liczb wpisanych przed zerem.
    return 0; // Kończymy program poprawnie.
} // Kończymy funkcję main.

```

Ćwiczenie 14 - wczytaj liczby do 0 i policz dodatnie

```

#include <iostream> // Dołączamy bibliotekę iostream.
using namespace std; // Używamy standardowej przestrzeni nazw.
int main() { // Rozpoczynamy program.
    int x; // Tworzymy zmienną na wczytywaną liczbę.
    int licznik = 0; // Ustawiamy licznik liczb dodatnich na 0.
    while (true) { // Rozpoczynamy pętlę działającą do użycia break.
        cin >> x; // Wczytujemy kolejną liczbę.
        if (x == 0) { // Sprawdzamy, czy wpisano wartość kończącą 0.
            break; // Jeśli tak, wychodzimy z pętli.
        } // Kończymy pierwszy warunek if.
        if (x > 0) { // Sprawdzamy, czy wczytana liczba jest dodatnia.
            licznik++; // Zwiększamy licznik dodatnich liczb o 1.
        } // Kończymy drugi warunek if.
    }

```



```

    }
    cout << licznik << endl;
    return 0;
}
// Kończymy pętlę while.
// Wypisujemy liczbę dodatnich wartości.
// Kończymy program poprawnie.
// Kończymy funkcję main.

```

Ćwiczenie 15 - tabliczka mnożenia liczby n

```

#include <iostream>
using namespace std;
int main() {
    int n;
    cin >> n;
    for (int i = 1; i <= 10; i++) {
        cout << n << " * " << i << " = " << n * i << endl;
    }
    return 0;
}
// Dołączamy bibliotekę wejścia i wyjścia.
// Używamy standardowej przestrzeni nazw.
// Rozpoczynamy program.
// Tworzymy zmienną n.
// Wczytujemy liczbę, dla której powstanie tabliczka mnożenia.
// Ustawiamy mnożniki od 1 do 10.
// Wypisujemy działanie n*i oraz jego wynik.
// Kończymy pętlę.
// Kończymy program poprawnie.
// Kończymy funkcję main.

```

Ćwiczenie 16 - wypisz 1, 12, 123, ... aż do n

```

#include <iostream>
using namespace std;
int main() {
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++) {
            cout << j;
        }
        cout << endl;
    }
    return 0;
}
// Dołączamy bibliotekę iostream.
// Używamy standardowej przestrzeni nazw.
// Rozpoczynamy program.
// Tworzymy zmienną n określającą liczbę wierszy.
// Wczytujemy n od użytkownika.
// Pętla zewnętrzna tworzy kolejne wiersze od 1 do n.
// Pętla wewnętrzna wypisuje liczby od 1 do aktualnego i.
// Wypisujemy kolejną liczbę j bez przejścia do nowej linii.
// Kończymy pętlę wewnętrzną.
// Po zakończeniu wiersza przechodzimy do nowej linii.
// Kończymy pętlę zewnętrzną.
// Kończymy program poprawnie.
// Kończymy funkcję main.

```

Ćwiczenie 17 - n wierszy po 5 gwiazdek

```

#include <iostream>
using namespace std;
int main() {
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= 5; j++) {
            cout << "*";
        }
        cout << endl;
    }
    return 0;
}
// Dołączamy bibliotekę iostream.
// Używamy standardowej przestrzeni nazw.
// Rozpoczynamy program.
// Tworzymy zmienną n określającą liczbę wierszy.
// Wczytujemy n od użytkownika.
// Pętla zewnętrzna wykonuje się n razy - po jednym razie na wiersz.
// Pętla wewnętrzna wykonuje się 5 razy w każdym wierszu.
// Wypisujemy jedną gwiazdkę bez przejścia do nowej linii.
// Kończymy pętlę wewnętrzną.
// Po pięciu gwiazdkach przechodzimy do nowego wiersza.
// Kończymy pętlę zewnętrzną.
// Kończymy program poprawnie.
// Kończymy funkcję main.

```

Ćwiczenie 18 - kwadrat z gwiazdek n x n

```

#include <iostream>
using namespace std;
int main() {
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= n; j++) {
            cout << "*";
        }
        cout << endl;
    }
    return 0;
}
// Dołączamy bibliotekę iostream.
// Używamy standardowej przestrzeni nazw.
// Rozpoczynamy program.
// Tworzymy zmienną n określającą bok kwadratu.
// Wczytujemy rozmiar kwadratu.
// Pętla zewnętrzna tworzy n wierszy.
// Pętla wewnętrzna wypisuje n gwiazdek w każdym wierszu.
// Wypisujemy jedną gwiazdkę.
// Kończymy pętlę wewnętrzną.
// Przechodzimy do nowego wiersza po wypisaniu n gwiazdek.
// Kończymy pętlę zewnętrzną.
// Kończymy program poprawnie.
// Kończymy funkcję main.

```

Ćwiczenie 19 - od 1 do n, zamiast parzystych wypisz X

```

#include <iostream>
using namespace std;
int main() {
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++) {
        if (i % 2 == 0) {

```

```

        cout << "X" << endl;
    } else {
        cout << i << endl;
    }
}
return 0;
}
// Dla liczby parzystej wypisujemy znak X.
// Jeśli liczba nie jest parzysta...
// ...wypisujemy jej rzeczywistą wartość.
// Kończymy instrukcję if/else.
// Kończymy pętlę.
// Kończymy program poprawnie.
// Kończymy funkcję main.

```

Ćwiczenie 20 - ile liczb od 1 do n jest podzielnych jednocześnie przez 3 i 5

```

#include <iostream>
using namespace std;
int main() {
    int n;
    cin >> n;
    int licznik = 0;
    for (int i = 1; i <= n; i++) {
        if (i % 3 == 0 && i % 5 == 0) {
5.         licznik++;
        }
    }
    cout << licznik << endl;
    return 0;
}
// Dołączamy bibliotekę wejścia i wyjścia.
// Używamy standardowej przestrzeni nazw.
// Rozpoczynamy program.
// Tworzymy zmienną n.
// Wczytujemy górną granicę zakresu.
// Ustawiamy licznik znalezionych liczb na 0.
// Sprawdzamy kolejno liczby od 1 do n.
// Sprawdzamy, czy i jest podzielne zarówno przez 3, jak i przez
// Jeśli oba warunki są prawdziwe, zwiększamy licznik o 1.
// Kończymy instrukcję if.
// Kończymy pętlę for.
// Wypisujemy liczbę wartości spełniających oba warunki.
// Kończymy program poprawnie.
// Kończymy funkcję main.

```