

*Sąlyginis sakinyis IF*  
**Sąlyginis sakinyis IF**

Labai dažnai sprendžiame uždavinius, kai vieną ar kelis veiksmus reikia atlikti ne nuosekliai, o tuomet, kai tenkinamos apibrėžtos sąlygos.

Pavyzdžiui, reikia nustatyti, ar įvestas natūralusis skaičius yra lyginis, ar nelyginis; parašyti sąlygą, kuri patikrintų, ar iš skaičiaus galima ištraukti kvadratinę šaknį (skaičius negali būti neigiamas) ir daugybę kitų uždavinių.

Užrašant sąlyginius sakinius naudojami palyginimo operatoriai:

==	<b>lygu</b>
!=	<b>nelygu</b>
>	<b>daugiau</b>
>=	<b>daugiau arba lygu</b>
<	<b>mažiau</b>
<=	<b>mažiau arba lygu</b>

### 1. Paprastas sąlyginis sakinyis

Paprastas sąlyginis sakinyis rašomas tuomet, kai vienu atveju atliekamas sakinyis, arba sakinių grupė, kai sąlyga tenkinama, kitu atveju – sakinyis, arba sakinių grupė, kai sąlyga netenkinama. Paprastas sąlyginis sakinyis užrašomas:

<b>if</b> ( <i>sąlyga</i> ) <i>Sakinys, kai sąlyga tenkinama;</i> <b>else</b> <i>Sakinys, kai sąlyga netenkinama;</i>	<b>if</b> ( <i>sąlyga</i> ) { <i>Sakiniai, kai sąlyga tenkinama;</i> } <b>else</b> { <i>Sakiniai, kai sąlyga netenkinama;</i> }
--------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------

#### 1.1. Pavyzdinė programa: Programa, tikrinanti, ar įvestas natūralusis skaičius $x$ yra lyginis, ar nelyginis.

```
//Lyginumas
#include <iostream>
#include <iomanip>
#include <cmath>
#include <fstream>

using namespace std;

int main () {
    int x;
    cout << "Įveskite x: ";
    cin >> x;
    if (x % 2 == 0) cout << "Lyginis" << endl;
    else cout << "Nelyginis" << endl;
    return 0;
}
```

Rezultatas:

Kai sąlyga tenkinama	Kai sąlyga netenkinama
Įveskite x: 4 Lyginis	Įveskite x: 5 Nelyginis

Sąlyginį sakinį IF galima užrašyti ir kitaip:

```
if (x % 2 != 0) cout << "Nelyginis" << endl;
    else cout << "Lyginis" << endl;
```

Rezultatas:

Kai sąlyga tenkinama	Kai sąlyga netenkinama
Iveskite x: 5 Nelyginis	Iveskite x: 4 Lyginis

**1.2. Pavyzdinė programa:** Programa, ištraukianti kvadratinę šaknį iš realiojo skaičiaus  $x$ .

```
//Kvadratinės šaknies traukimas
#include <iostream>
#include <iomanip>
#include <cmath>
#include <fstream>

using namespace std;

int main () {
    double x;
    cout << "Iveskite x: ";
    cin >> x;
    if (x >= 0) {
        double y = sqrt(x);
        cout << fixed << setprecision(2) << y << endl;
    }
    else cout << "Skaičius neigiamas" << endl;
    return 0;
}
```

Kai sąlyga tenkinama	Kai sąlyga netenkinama
Iveskite x: 5 2.24	Iveskite x: -6 Skaičius neigiamas

Sąlyginį sakinį galima užrašyti ir taip:

```
if (x < 0) cout << "Skaičius neigiamas" << endl;
else
{
    double y = sqrt(x);
    cout << fixed << setprecision(2) << y << endl;
}
```

Kai sąlyga tenkinama	Kai sąlyga netenkinama
Iveskite x: -6 Skaičius neigiamas	Iveskite x: 5 2.24

Jei koks nors rezultatas išvedamas tik tuomet, kai sąlyga tenkinama, tuomet rašomas sutrumpintas sąlyginis sakinyis. Pvz., jei 1.1. pavyzdinėje programoje sąlyginį sąkinį užrašytume:

```
if (x % 2 == 0) cout << "Lyginis" << endl;
```

Tuomet rezultatai būtų tokie:

Kai įvedamas lyginis skaičius	Kai įvedamas nelyginis skaičius
Iveskite x: 4 Lyginis	Iveskite x: 5

Galima sąlyginį sąkinį užrašyti ir taip, kad rezultatas būtų išvedamas tuomet, kai sąlyga netenkinama. Pvz., jei 1.1. pavyzdinėje programoje sąlyginį sąkinį užrašytume:

```
if (x % 2 == 0);
else cout << "Nelyginis" << endl;
```

Tuomet rezultatai būtų tokie:

Kai įvedamas lyginis skaičius	Kai įvedamas nelyginis skaičius
Iveskite x: 4	Iveskite x: 5 Nelyginis

1.3. [Uždaviniai iš konspekto](#): psl. 21 – 2; psl. 22 – 4, 5, 6, 7, 8, 9; psl. 23 – 10, 11, 12, 13, 14, 15; psl. 24 – 1, 2, 3, 4, 5.

1.4. Uždaviniai iš KTU programavimo pamokėlių:

- [2013-2014 m.m. – psl. 8-9: 1. Parašiotininkas; 2. Didžioji kūdrinė varlė.](#)
- [2014-2015 m.m. – psl. 8-9: 1. Pietūs; 2. Knygos; 3. Indai.](#)

## 2. Sudėtingas sąlyginis sakinyis

Sudėtingu sąlyginiu sakiniu vadinamas toks sakinyis, kurio šakose yra papildomų sąlyginių sakinių. Sudėtingi sąlyginiai sakiniai dažnai rašomi, kai reikia patikrinti daugiau, negu 2 sąlygas.

2.1. **Pavyzdinė programa:** Lietuvių liaudies pasakose karžygys Jonas, jojantis išvaduoti karalaitės, dažnai turi rinktis vieną iš 4 kelių. Jei Jonas pasirinks pirmą kelią, jo laukia neįžengiama giria, jei antrą – kautynės su piktomis raganomis, jei trečią – tiesus kelias pas karalaite, jei ketvirtą – kautynės su piktuoju slibinu. Parašysime programą, kuri, nurodžius, kokį iš kelių pasirinko Jonas, pasakys, kur kelias veda. Kelias įvedamas kaip sveikasis skaičius.

```
//Jono kelias
#include <iostream>
#include <iomanip>
#include <cmath>
#include <fstream>

using namespace std;

int main () {
    int k;
    cout << "Kuri kelias pasirinko Jonas? Iveskite kelio numeri nuo 1 iki 4: ";
    cin >> k;
    if (k == 1) cout <<"Jono laukia neįžengiama giria" << endl;
    else if (k == 2) cout << "Jono laukia kautyines su piktomis raganomis" << endl;
        else if (k == 3) cout << "Jono laukia tiesus kelias pas karalaite" << endl;
            else if (k == 4) cout << "Jono laukia kautyines su piktuoju slibinu"
                << endl;
                else cout << "Tokio kelio nera" << endl;

    return 0;
}
```

Rezultatai, gaunami įvedus skirtingus kelius:

```
Kuri kelia pasirinko Jonas? Iveskite kelio numeri nuo 1 iki 4: 1
Jono laukia neizengiama giria
```

```
Kuri kelia pasirinko Jonas? Iveskite kelio numeri nuo 1 iki 4: 2
Jono laukia kautynes su piktomis raganomis
```

```
Kuri kelia pasirinko Jonas? Iveskite kelio numeri nuo 1 iki 4: 3
Jono laukia tiesus kelias pas karalaite
```

```
Kuri kelia pasirinko Jonas? Iveskite kelio numeri nuo 1 iki 4: 4
Jono laukia kautynes su piktuuju slibinu
```

```
Kuri kelia pasirinko Jonas? Iveskite kelio numeri nuo 1 iki 4: 10
Tokio kelio nera
```

2.2. [Uždaviniai iš konspekto](#): psl. 25 – 1, 2, 3; psl. 26 – 4, 5, 6, 7.

2.3. Uždaviniai iš KTU programavimo pamokėlių:

- [2013-2014 m.m. – psl. 9: 3. Matematika; 4. Bandelės.](#)

### 3. Kelių sąlygų jungimas loginėmis operacijomis

3.1. [Pavyzdžiai iš konspekto](#): psl. 25 – 28.

3.2. [Uždaviniai iš konspekto](#): psl. 28 – 1, 2, 3; psl. 29 – 4, 5, 6, 7; psl. 30 – 8, 9, 10, 11; psl. 31 – 12.

3.3. Iš KTU programavimo pamokėlių:

- [2013-2014 m.m. – psl. 10: 5. Geriausia klasė; 6. Žiemojantys paukščiai.](#)
- [2014-2015 m.m. – psl. 10-12: 6. Šaulys; 7. Populiariausios knygos; 8. Gimimo metai.](#)
- [2015-2016 m.m. – psl. 6-7: 1. Trikampis; 2. Dvi dėžutės; 3. Kelių remontas; 4. Bilietas į koncertą; 5. Triušiai.](#)
- [2016-2017 m.m. – psl. 8-10: 1. Kompasas; 2. Keturženklis skaičiaus formavimas; 3. Lenktynės; 4. Žalos skaičiavimas; 5. Stačiakampio ketvirtoji viršūnė.](#)