

**Priskyrimo sakiny**

# Priskyrimo sakiny

Tai kintamasis ir reiškinys, atskirti priskyrimo simboliu = . Vykdam programą apskaičiuojama reiškinio reikšmė ir ji priskiriama kairėje pusėje esančiam kintamajam.

Priskyrimo sakinių pavyzdžiai:

```
sk = 5;      suma = x + y;
```

```
a = b = c = 2 * 2;
```

```
y = (5 + 2 * x) / (3 * x - 1);
```

```
t = true;   s = 'N';   l = "labas";
```

```
a + b = d;
```

# Kiti priskyrimo sakinio pavyzdžiai

Užrašas	Sutrumpintai
<code>x = x + 1;</code>	<code>x++;</code>
<code>y = y - 1;</code>	<code>y--;</code>
<code>a = a + 5;</code>	<code>a += 5;</code>
<code>b = b * 5;</code>	<code>b *= 5;</code>

## Dažniausiai naudojamos operacijos:

- + sudėties veiksmas ir teigiamas skaičiaus ženklas;
- atimties veiksmas ir neigiamas skaičiaus ženklas;
- \* daugyba;
- / dalyba;
- % tik sveikajam tipui taikomas liekanos skaičiavimo veiksmas.

*Jeigu dalijame sveikuosius skaičius ir norime gauti dalmenį tik sveikąjį skaičių, tai renkames sveikųjų skaičių dalybos veiksmą / , o jeigu norime gauti tik liekaną - %*

Sveikųjų skaičių dalybos pavyzdžiai:

$$13 / 5 = 2; \quad 3 / 8 = 0; \quad 0 d / 7 = 0;$$

$$13 \% 5 = 3; \quad 3 \% 8 = 3; \quad 0 \% 7 = 0.$$

**Primename, kad iš nulio dalyti negalima.**

Nuo paros pradžios  
praėjo  $s$  sekundžių.  
Parašykite programą,  
kuri apskaičiuotų kiek  
sveikų valandų ir ir  
minučių jos sudaro.

```
int sek, min, val, liekana;
cout << "įveskite sekundes s";
cin >> sek;
//3796
val = sek /3600;
//3796/3600 = 1
liekana = sek % 3600;
// 3796-1*3600 = 196
min = liekana / 60;
//196 / 60 = 3
cout << val << " val. " << min
<< " min.";
```

Parašykite programą,  
kuri apskaičiuotų bet  
kurio triženklio skaičiaus  
skaitmenų sumą.

```
int main ()
{
int tr_sk, s1, s2, s3, suma;
cout << "İveskite 3ženklį sk.";
cin >> tr_sk;
s1 = tr_sk / 100;
s2 = tr_sk % 100 / 10;
s3 = tr_sk % 10;
suma = s1 + s2 + s3;
cout << "Suma" << setw(5) << suma;
return 0;
}
```



# Sąlyginis sakiny

Būna situacijų, kai reikia pasirinkti vieną veiksmą iš dviejų. Toks veiksmų pasirinkimas vadinamas šakojimu, o jam aprašyti programavime naudojamas sąlyginis sakiny.

Jei yra daugiau nei du pasirinkimo atvejai, tuomet į vieną sąlyginį sakinį įterpiamas kitas sąlyginis sakiny.

# Sąlyginis sakinyys

```
if (sąlyga)
    Sakinys1;
else
    Sakinys2;
```

Jei sąlyga tenkinama  
atliekamas **Sakinys1** ,  
jei ne – po žodžiu **else**  
esantis **Sakinys2**

```
if (sąlyga) {
    Sakinys;
    Sakinys;
    ...
}
else {
    Sakinys;
    Sakinys;
    ...
}
```

# Sutrumpintas sąlyginis sakinyys

```
if (sąlyga)  
    Sakinys1;
```

Atliekami veiksmai  
tik tuomet kai  
sąlyga tenkinama

```
if (sąlyga) {  
    Sakinys;  
    Sakinys;  
    ...  
}
```

```
int main()
{
int a, b, c;
cout << "Iveskite 2 skaicius skaicius:" <<
endl;
cin >> a >> b;
cout << "a ir b sandauga yra? ";
cin >> c;

if (a*b == c) //salyginis sakinyys
cout << "Teisingai!" << endl;
else
cout << "Klaida..." << endl;
return 0;
}
```

# Sąlygose naudojamos santykio operacijos

- ==** tikrinama ar kintamieji lygūs
- !=** tikrinama ar kintamieji nelygūs
- <** tikrinama ar pirmas kintamasis mažesnis už antrą
- >** tikrinama ar pirmas kintamasis didesnis už antrą
- <=** tikrinama ar pirmas kintamasis mažesnis už antrą arba lygus jam
- >=** tikrinama ar pirmas kintamasis didesnis už antrą arba lygus jam
- ||** tikrinama ar bent vienas iš kintamųjų tinka pagal sąlygą (loginis ARBA)
- &&** tikrinama ar visi kintamieji tinka pagal sąlygą (loginis IR)
- !** – (loginis ne)