# Programmeerimine

6.loeng

# Täna loengus

- Algoritmidest
- Plokkskeemidest

# Algoritm I

- 9. sajandi araabia matemaatiku Muhammad ibn Mûsâ al-Khwârizmî hüüdnimi tema sünnilinna tolleaegse nime järgi *Wikipedia*
- Eeskiri, mis määrab teatavat kindlat tüüpi ülesannete lahendamiseks vajalikud operatsioonid ning nende sooritamise järjekorra ehk nn. arvutusprotsessi. *Ü. Kaasik, Matemaatika leksikon, 2003*
- Korrastatud reeglikogum ülesande lahendamiseks lõpliku arvu sammudega. *V.Hanson, A. Tavast, Arvutikasutaja sõnastik* `http://www.keeleveeb.ee/`

# Algoritm II

- Täpne formaliseeritud eeskirjade kogum kindlat tüüpi ülesannete lahendamiseks, lahenduseeskiri. *Eesti keele seletav sõnaraamat, 2006* `http://www.keeleveeb.ee/`
- Lahenduseeskiri. *Eesti õigekeelsussõnaraamat, 2006* `http://www.keeleveeb.ee/`
- Eeskiri, mille järgi saab lõpliku arvu sammudega lahendada iga ülesande mingist kindlast ülesannete klassist. Lahendusalgoritmide leidmine ülesannete klasside jaoks on põhiprobleem klassikalises algebras, arvutusmatemaatikas jm. Arvutisse sisestatakse algoritm programmina. *ENE 1985*

# Jaota

- Jaota ideed ja instruktsioonid väiksemateks selliselt, et see oleks arusaadav ka sellele, kellele probleem on uus.
- Ole selgesõnaline ja ühemõtteline.
- Ära mine liiga detailseks.
- Ära kirjuta koodi.

# Ülesanne - täita kliendi soov

- Vaja jalanõusid

# Ülesanne - täita kliendi soov

- Vaja jalanõusid
- Pole varem kunagi jalanõusid ostnud

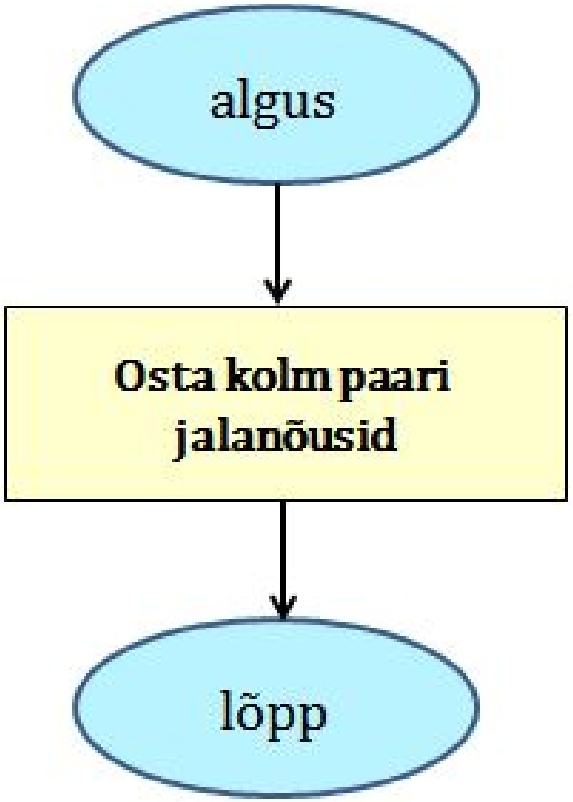
# Ülesanne - täita kliendi soov

- Vaja jalanõusid
- Pole varem kunagi jalanõusid ostnud
- Vaja erinevaid

# Ülesanne - täita kliendi soov

- Vaja jalanõusid
- Pole varem kunagi jalanõusid ostnud
- Vaja erinevaid

# Osta kolm paari jalanõusid



# Millised on ülesande tingimused, kontekst

- Kas ta teab midagi jalanõude stiilidest?
- Kas ta teab midagi jalanõude suurusest?
- Kas osta veebist või poest?
- Kas ta teab, kus on jalanõude pood?
- Kas oskab arvutada?
- ...

- Mine poodi
- Osta üks paar jalanõusid
- Osta üks paar jalanõusid
- Osta üks paar jalanõusid
- Mine koju

Osta üks paar jalanõusid

# Jaotame osadeks

Osta üks paar jalanõusid

Jaotame osadeks

Vaata, mida müüakse

Lihtne

Osta üks paar jalanõusid

Jaotame osadeks

Vaata, mida müüakse

↓

Vali stiil

Lihtne

Isiklik eelistus

Osta üks paar jalanõusid

Jaotame osadeks

Vaata, mida müüakse

Lihtne

Vali stiil

Isiklik eelistus

Leia sobiv paar

Vajab detailsemat

Osta üks paar jalanõusid

Jaotame osadeks

Vaata, mida müüakse

Lihtne

Vali stiil

Isiklik eelistus

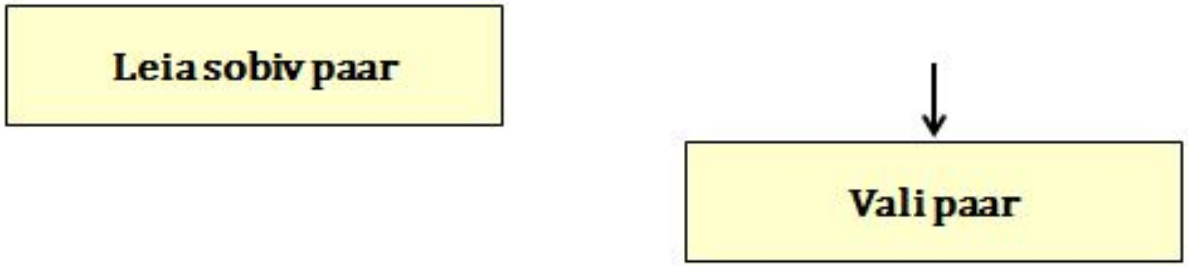
Leia sobiv paar

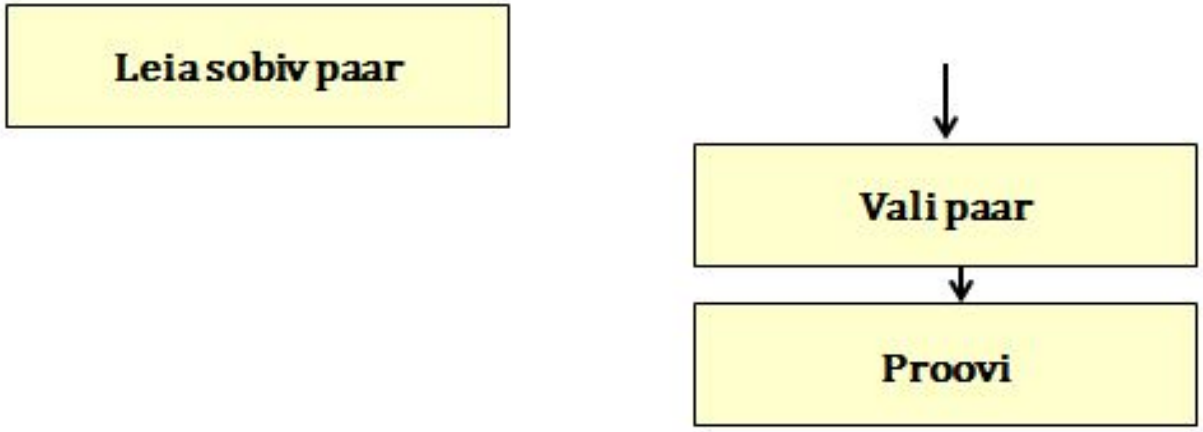
Vajab detailsemat

Maksa kauba eest

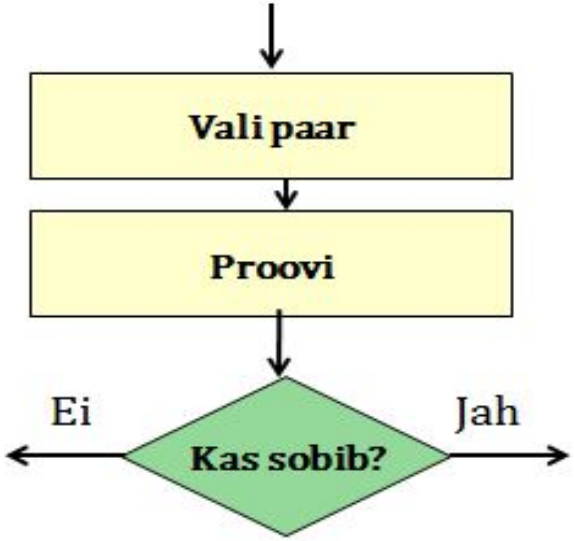
Liiga keeruline

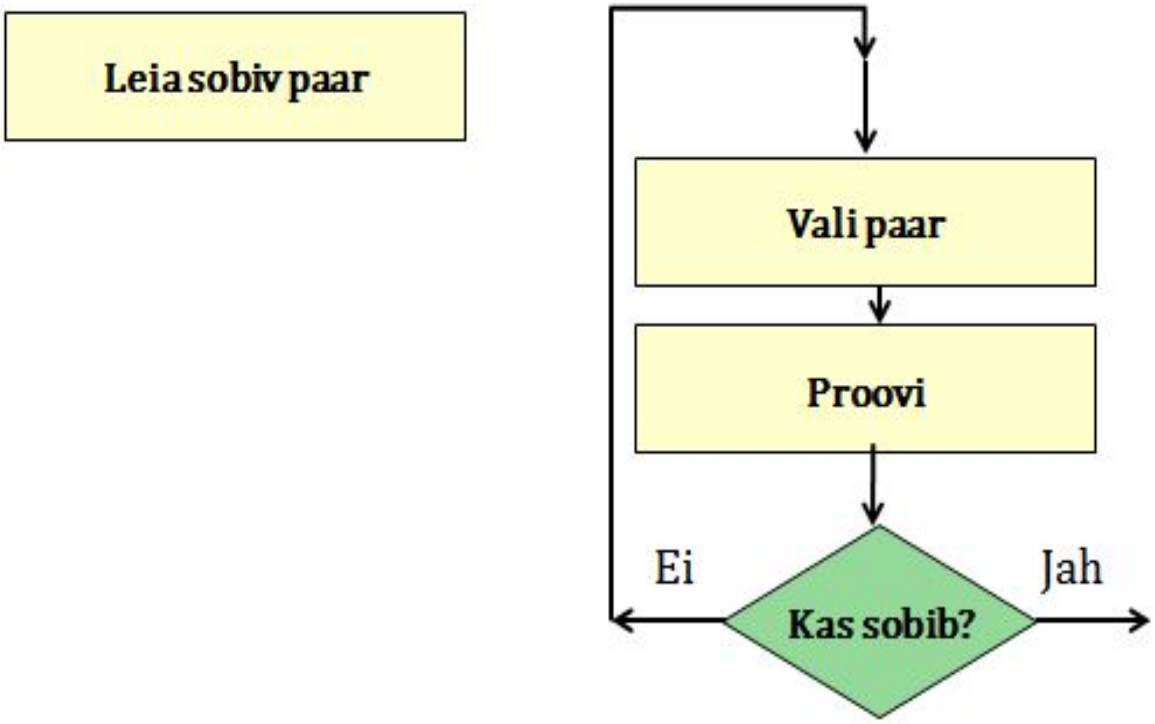
**Leia sobiv paar**


Leia sobiv paar
Vali paar


Leia sobiv paar
Vali paar
Proovi

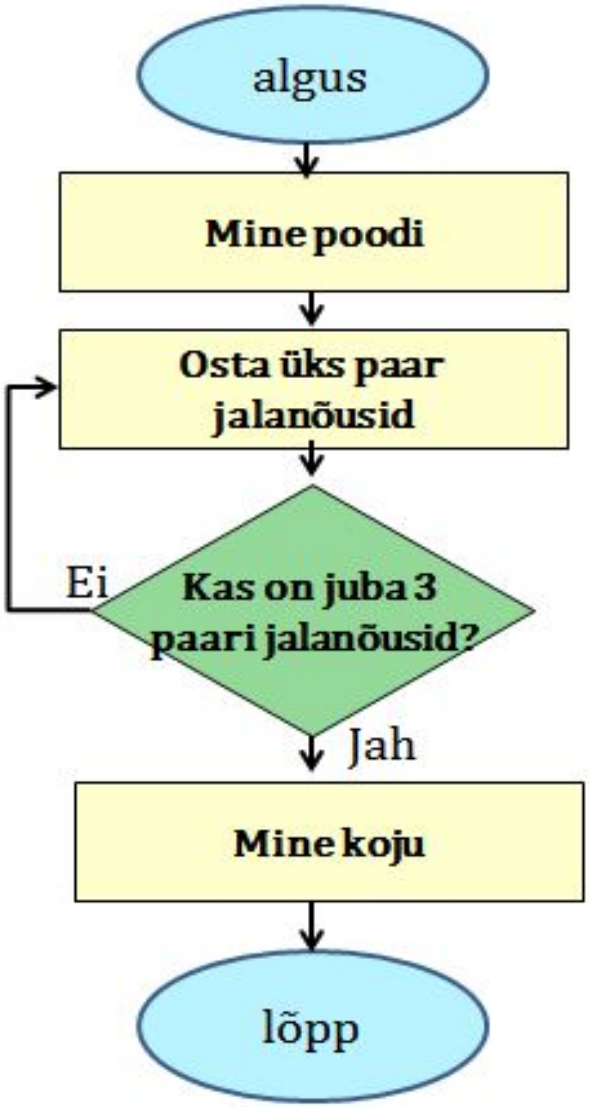
**Leia sobiv paar**

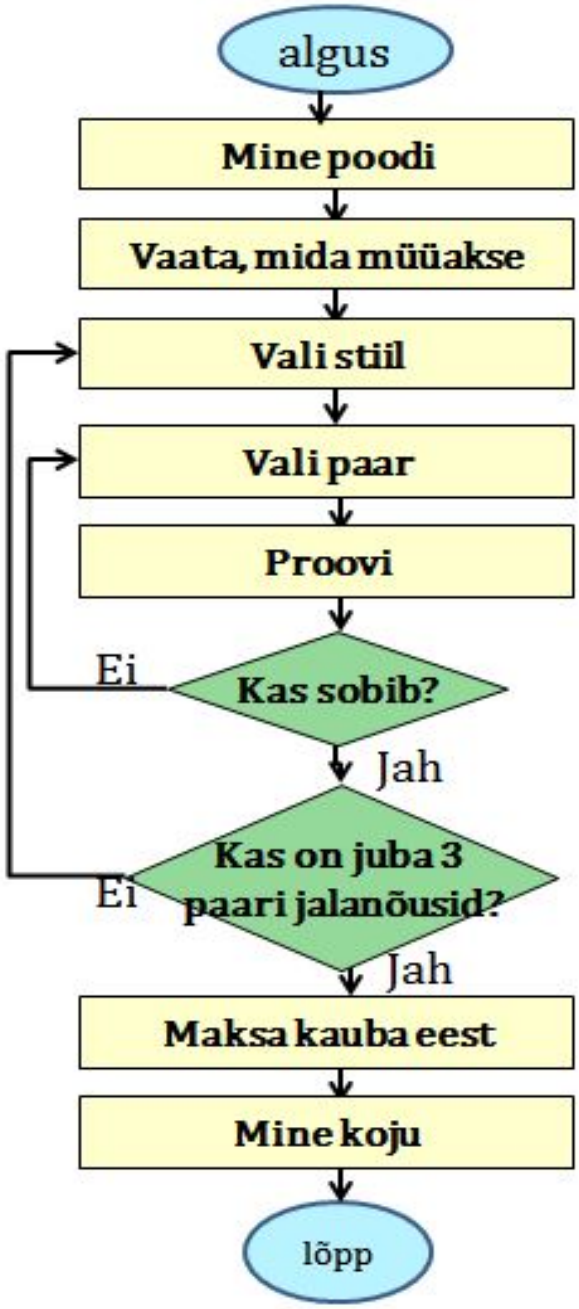



Leia sobiv paar
Vali paar
Proovi
Ei
Jah
Kas sobib?

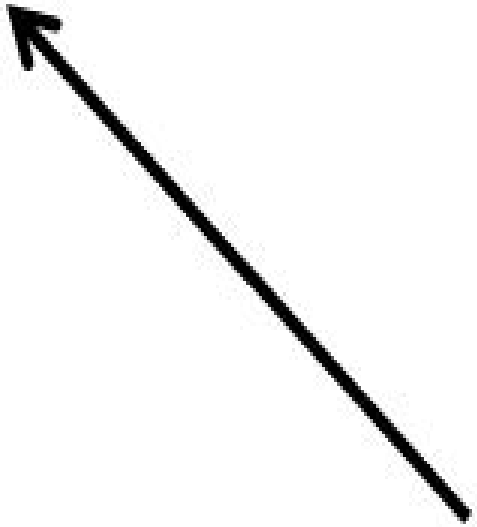
algus
Mine poodi
Osta üks paar jalanõusid
Osta üks paar jalanõusid
Osta üks paar jalanõusid
Mine koju
lõpp

algus
Mine poodi
Osta üks paar jalanõusid
Osta üks paar jalanõusid
Osta üks paar jalanõusid
Mine koju
lõpp

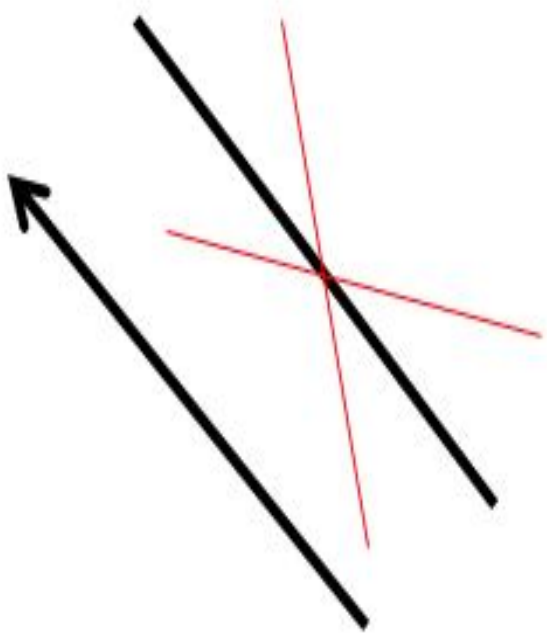

algus
Mine poodi
Osta üks paar jalanõusid
Ei
Kas on juba 3 paari jalanõusid?
Jah
Mine koju
lõpp


algus
Mine poodi
Vaata, mida müüakse
Vali stiil
Vali paar
Proovi
Kas sobib?
Ei
Jah
Kas on juba 3 paari jalanõusid?
Ei
Jah
Maksa kauba eest
Mine koju
lõpp

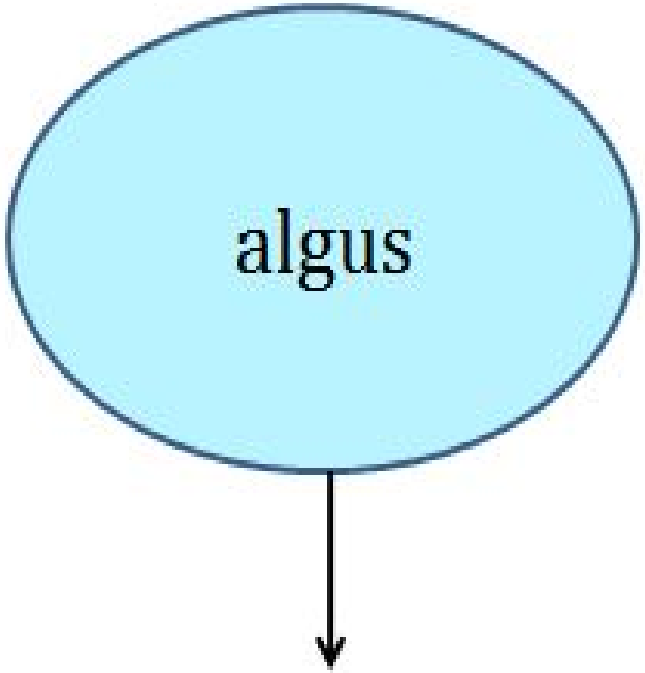
# Tegevuste järjekord



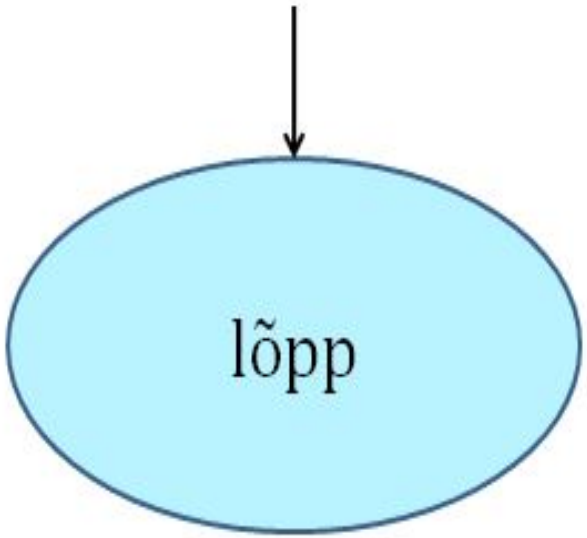
# Tegevuste järjekord



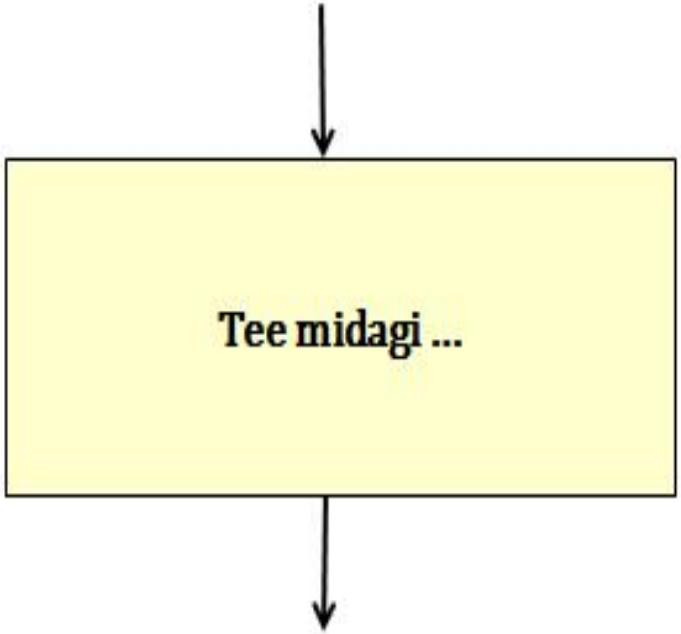
# Algus, lõpp



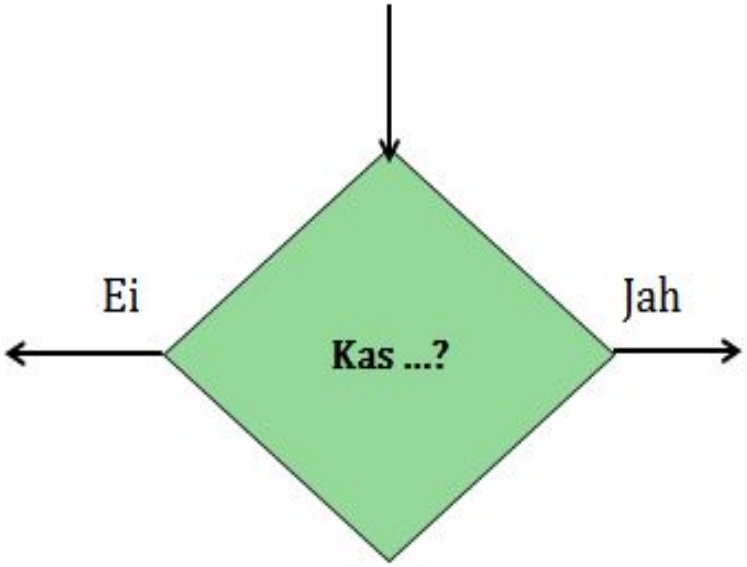
# Algus, lõpp



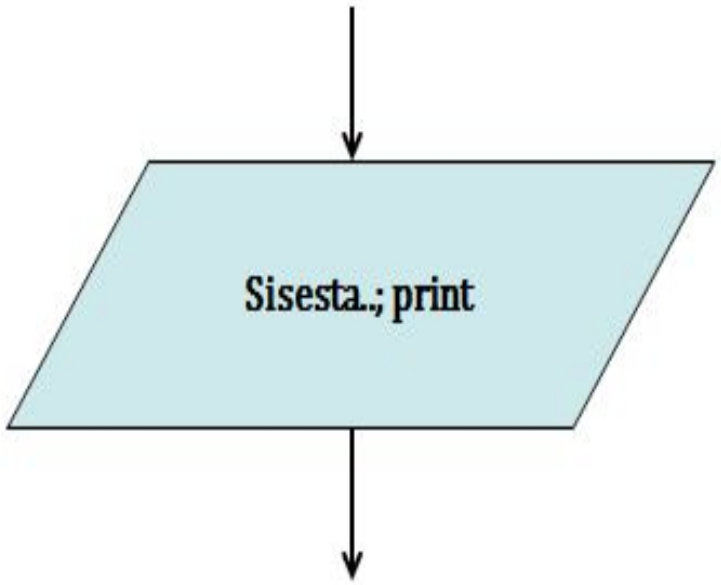
# Protsess



# Otsustus



# Sisend, väljund



# Ülesanne - leida arvude summa

# Ülesanne - leida n arvu summa

# Ülesanne - leida n arvu summa

- 1. sum = 0 /protsess

# Ülesanne - leida n arvu summa

- 1. sum = 0 /protsess
- 2. loend = 0 /protsess

# Ülesanne - leida n arvu summa

- 1. sum = 0 /protsess
- 2. loend = 0 /protsess
- 3. Küsi kasutajalt n /sisend

# Ülesanne - leida n arvu summa

- 1. sum = 0 /protsess
- 2. loend = 0 /protsess
- 3. Küsi kasutajalt n /sisend
- 4. Kas kõik liidetud? /otsustus
  kui JAH, siis jätka punktist ..

# Ülesanne - leida n arvu summa

- 1. sum = 0 /protsess
- 2. loend = 0 /protsess
- 3. Küsi kasutajalt n /sisend
- 4. Kas kõik liidetud? /otsustus
  kui JAH, siis jätka punktist ..
- 5. Küsi kasutajalt arv x /sisend

# Ülesanne - leida n arvu summa

- 1. sum = 0 /protsess
- 2. loend = 0 /protsess
- 3. Küsi kasutajalt n /sisend
- 4. Kas kõik liidetud? /otsustus
  kui JAH, siis jätka punktist ..
- 5. Küsi kasutajalt arv x /sisend
- 6. sum = sum + x /protsess

# Ülesanne - leida n arvu summa

- 1. sum = 0 /protsess
- 2. loend = 0 /protsess
- 3. Küsi kasutajalt n /sisend
- 4. Kas kõik liidetud? /otsustus
  kui JAH, siis jätka punktist ..
- 5. Küsi kasutajalt arv x /sisend
- 6. sum = sum + x /protsess
- 7. loend = loend + 1 /protsess

# Ülesanne - leida n arvu summa

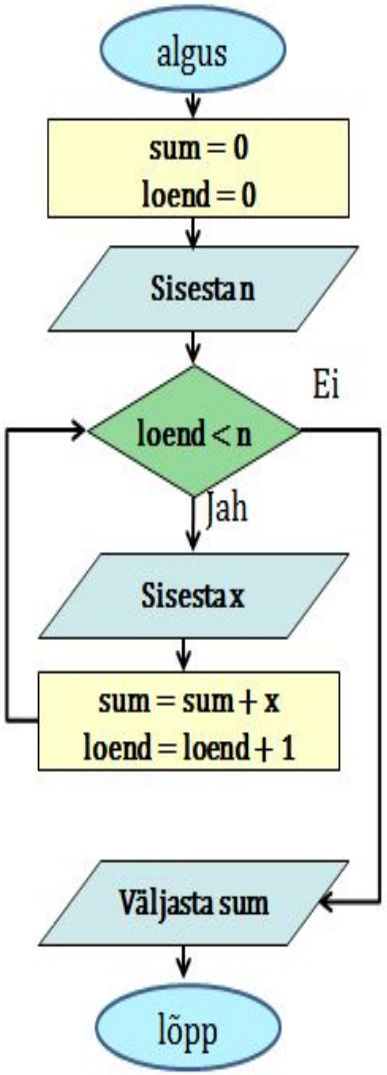
- 1. sum = 0 /protsess
- 2. loend = 0 /protsess
- 3. Küsi kasutajalt n /sisend
- 4. Kas kõik liidetud? /otsustus
  kui JAH, siis jätka punktist ..
- 5. Küsi kasutajalt arv x /sisend
- 6. sum = sum + x /protsess
- 7. loend = loend + 1 /protsess
- 8. Jätka punktist 4

# Ülesanne - leida n arvu summa

- 1. sum = 0 /protsess
- 2. loend = 0 /protsess
- 3. Küsi kasutajalt n /sisend
- 4. Kas kõik liidetud? /otsustus
  kui JAH, siis jätka punktist 9
- 5. Küsi kasutajalt arv x /sisend
- 6. sum = sum + x /protsess
- 7. loend = loend + 1 /protsess
- 8. Jätka punktist 4
- 9. Väljasta sum /väljund

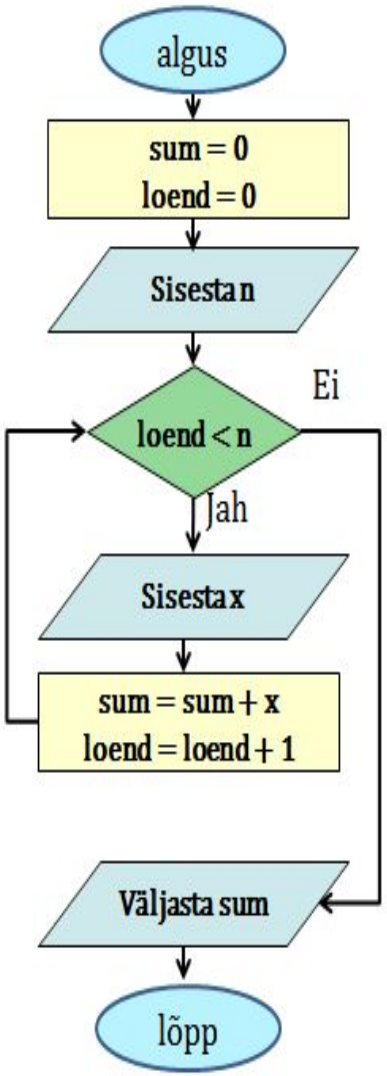
# Ülesanne - leida n arvu summa

- 1. sum = 0 /protsess
- 2. loend = 0 /protsess
- 3. Küsi kasutajalt n /sisend
- 4. Kas kõik liidetud? /otsustus
  kui JAH, siis jätka punktist 9
- 5. Küsi kasutajalt arv x /sisend
- 6. sum = sum + x /protsess
- 7. loend = loend + 1 /protsess
- 8. Jätka punktist 4
- 9. Väljasta sum /väljund



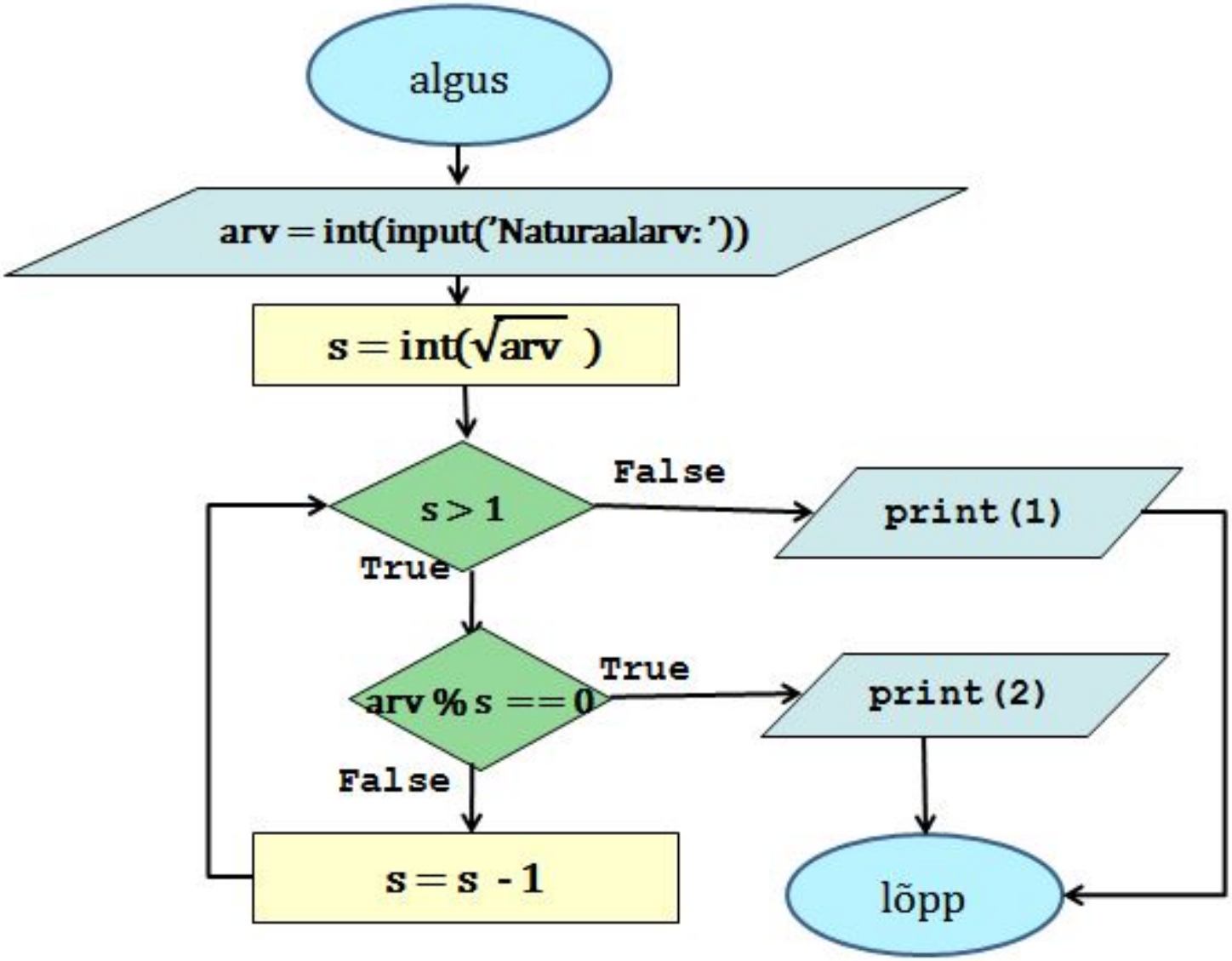
# Ülesanne - leida n arvu summa

```
sum = 0
loend = 0
n = int(input("Sisesta n: "))
while loend < n:
  x = float(input("Sisesta arv: "))
  sum += x
  loend += 1
print("Arvude summa: ", sum)
```



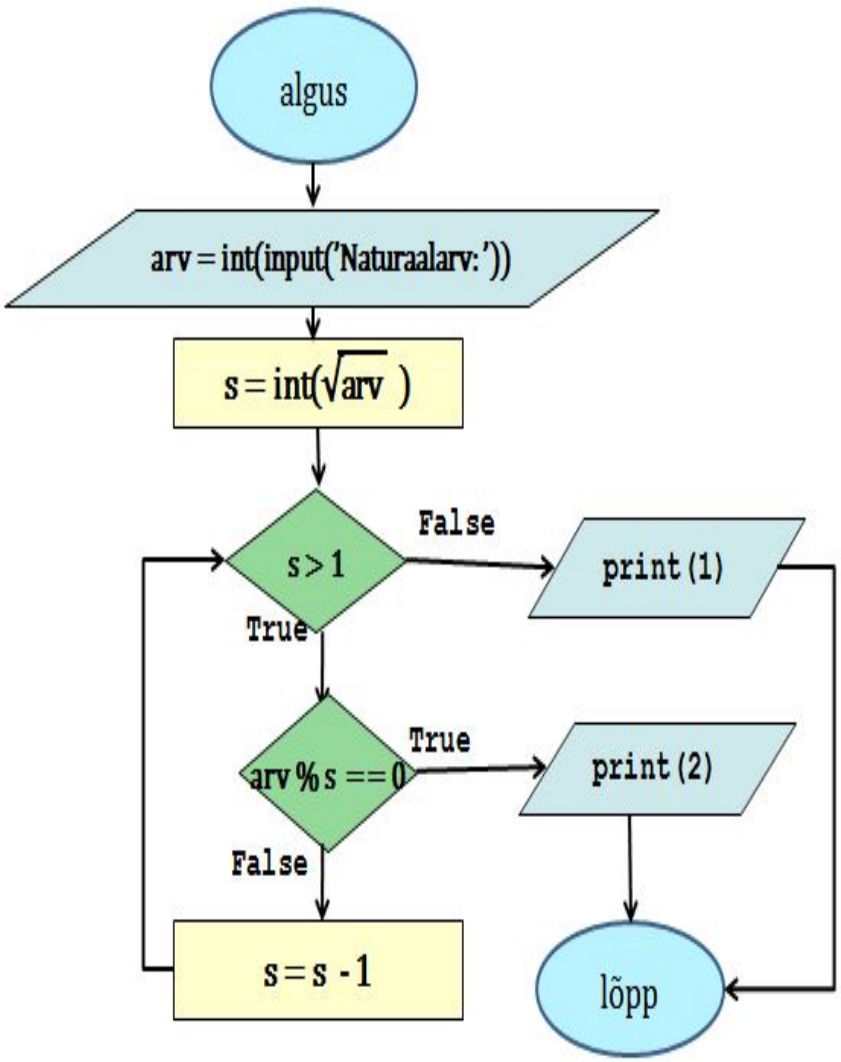
# Plokkskeem - I

Kasutaja sisestab 71. Mida trükitakse?



# Plokkskeem - I

```
from math import sqrt
arv = int(input("Sisesta arv: "))
s = int(sqrt(arv))
while True:
    if not s > 1:
        print(1)
        break
    elif arv % s == 0:
        print(2)
        break
    else:
        s -= 1
```

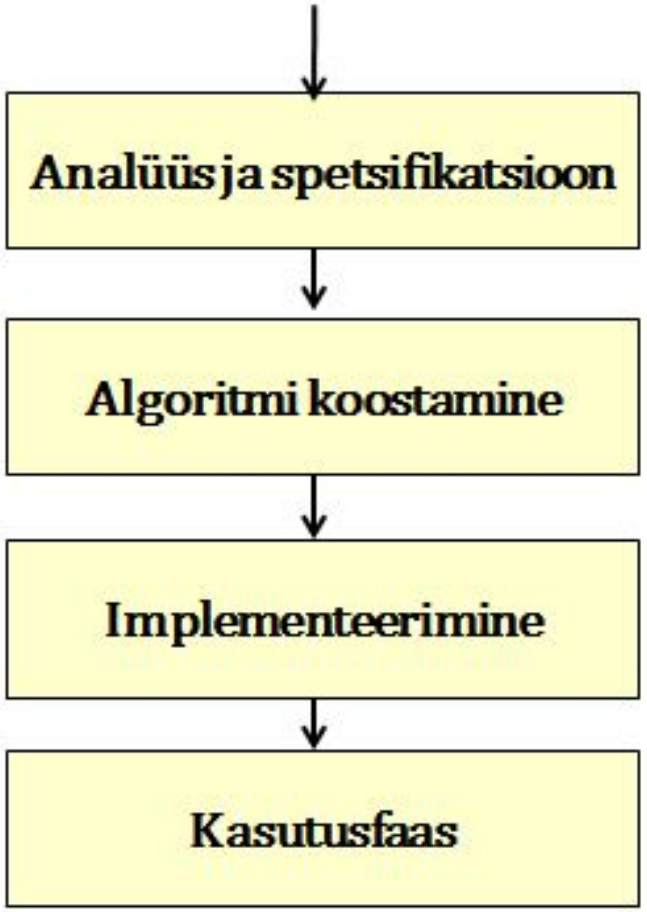


# Ülesannete lahendamine (*G. Pólya*)

- Ülesandest arusaamine.
- Lahendamise idee ja sellele vastava plaani koostamine.
- Lahendusplaani täitmine.
- Tagasivaade.

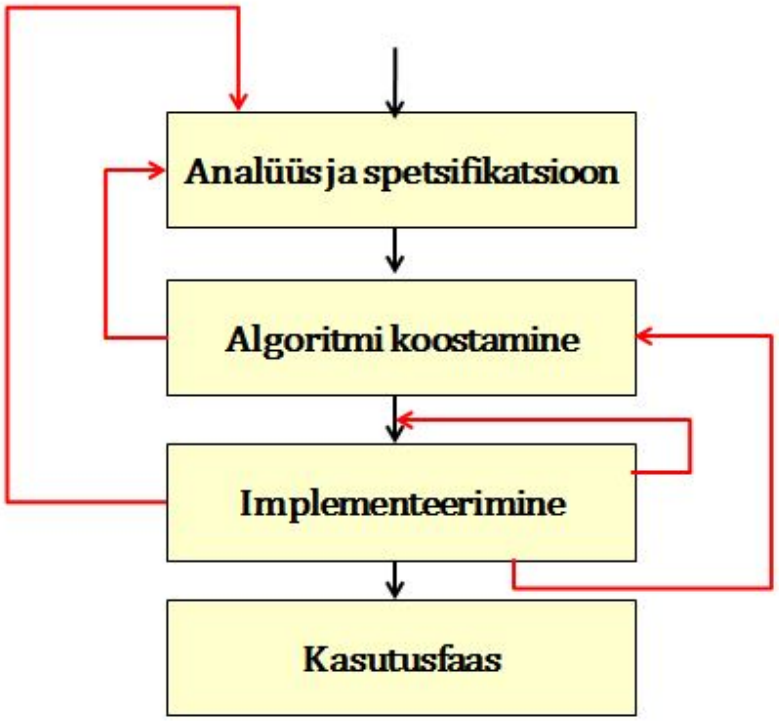
# Ülesande lahendamine arvutil

- Analüüs ja spetsifikatsioon
- Algoritmi koostamine
- Implementeerimine
- Kasutusfaas



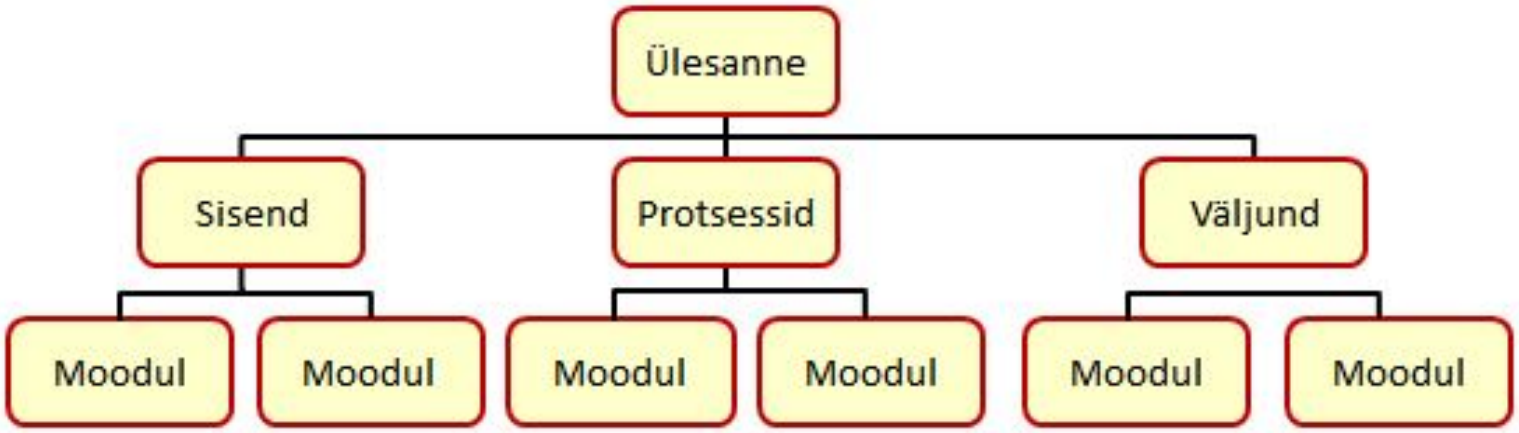
# Ülesande lahendamine arvutil

- Analüüs ja spetsifikatsioon
- Algoritmi koostamine
- Implementeerimine
- Kasutusfaas



# Algoritmi koostamine

- Sisendid
- Protsessid
- Väljundid
- Protsesside hierarhia
- Moodulid

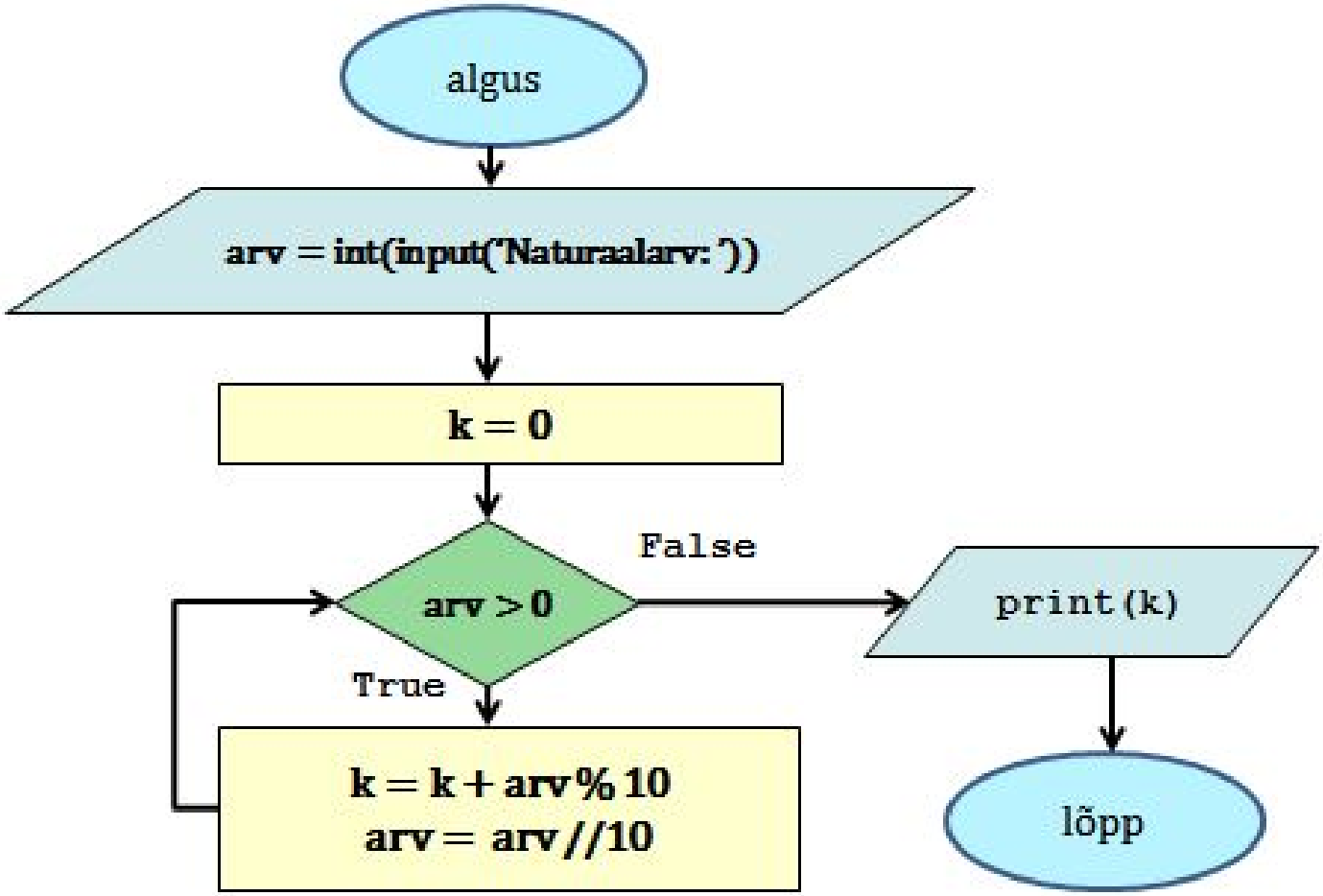


# Küsimused algoritmi kohta

- Kas algoritm lahendab püstitatud probleemi?
- Kas algoritm on korrektne?
- Kas algoritm annab soovitud väljundi?
- Kas algoritm lõpetab mõistliku ajaga?

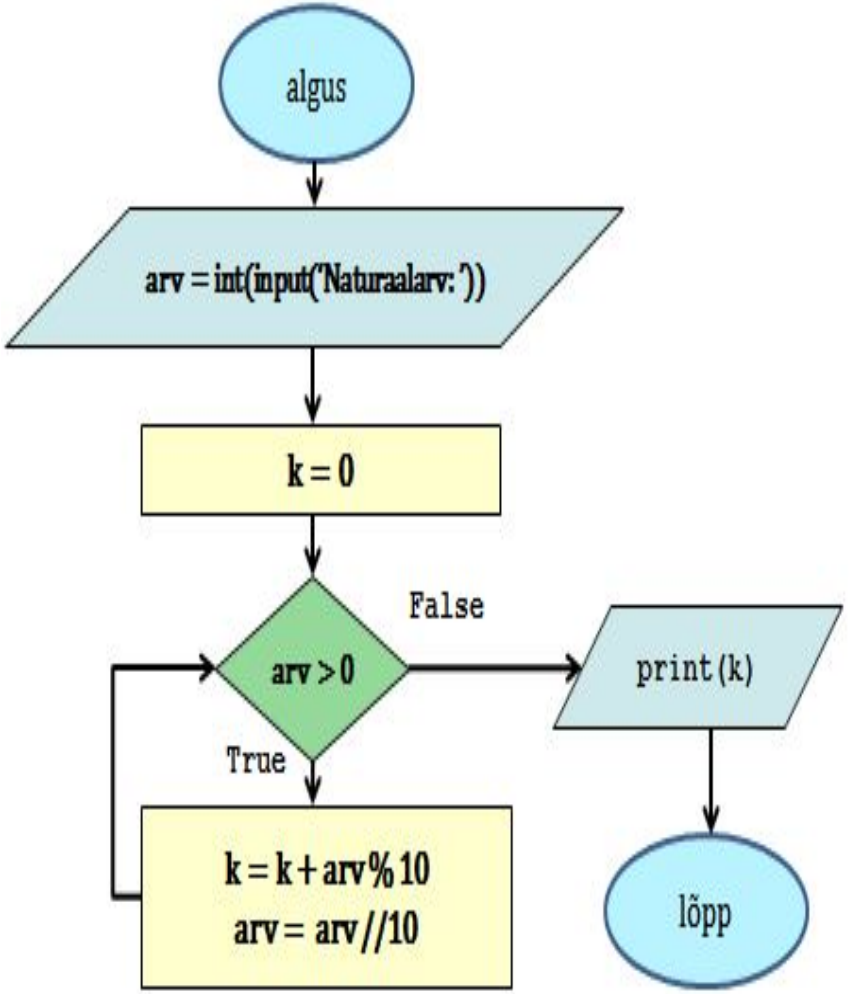
# Plokkskeem - II

Kasutaja sisestab 1234. Mida trükitakse?



# Plokkskeem - II

```
arv = int(input('Naturaalarv: '))
k = 0
while arv > 0:
    k += arv % 10
    arv //= 10
print(k)
```



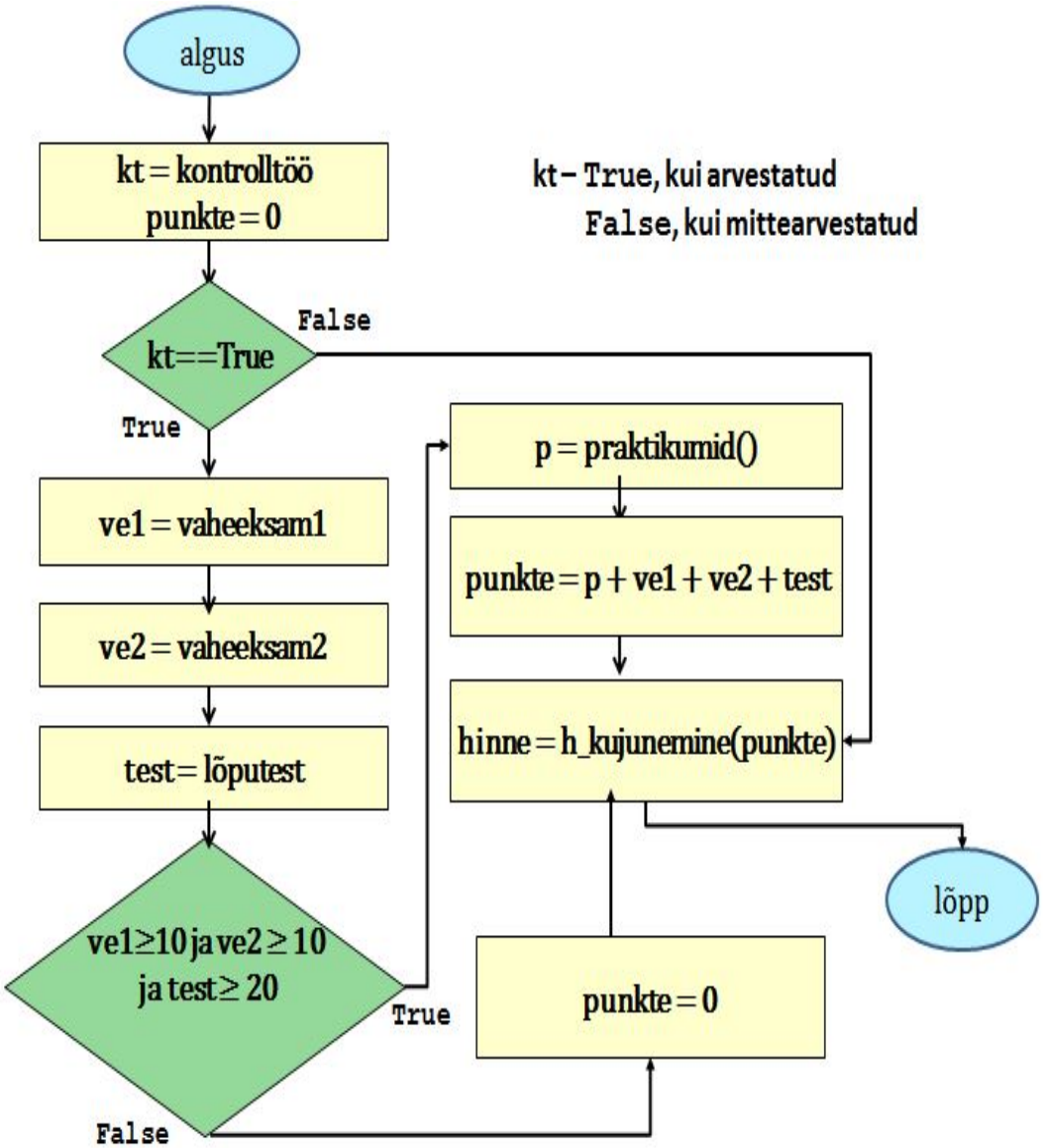
# Programmeerimise kursuse läbimine

# Programmeerimise kursuse läbimine

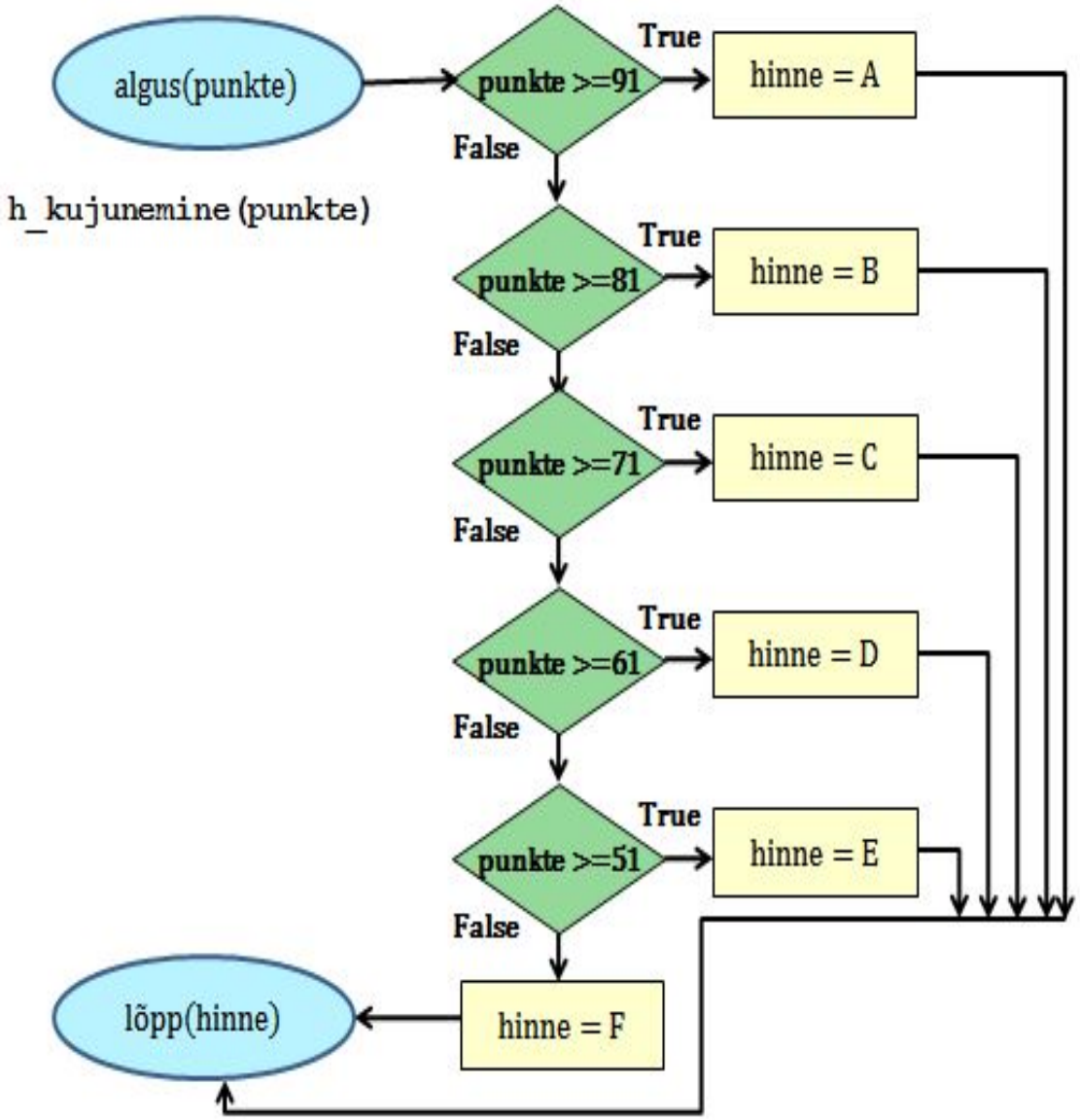
- Praktikumid
- 1. vaheeksam
- 2. vaheeksam
- Lõputest
- Hinne

# Programmeerimise kursuse läbimine

- Praktikumid
- 1. vaheeksam
- 2. vaheeksam
- Lõputest
- Hinne

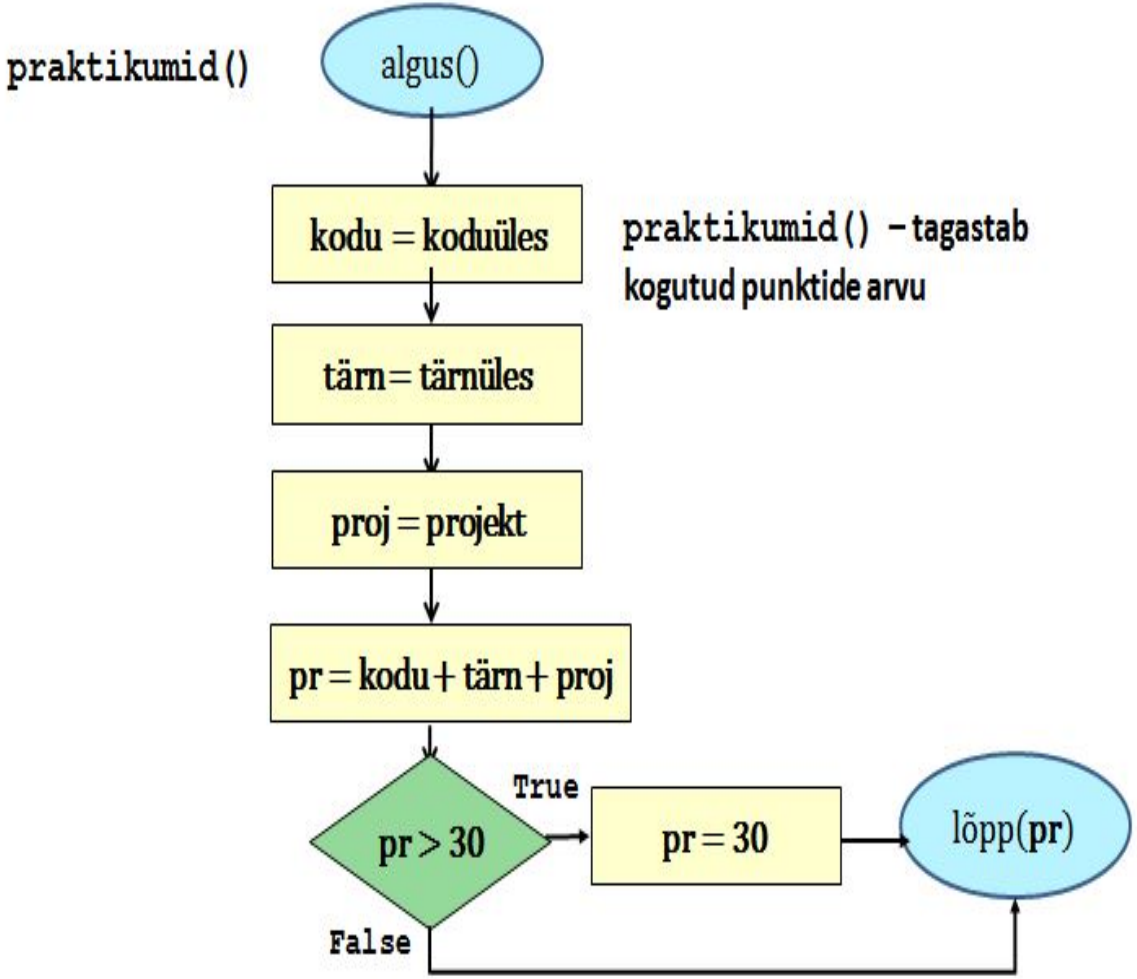


# Hinde kujunemine

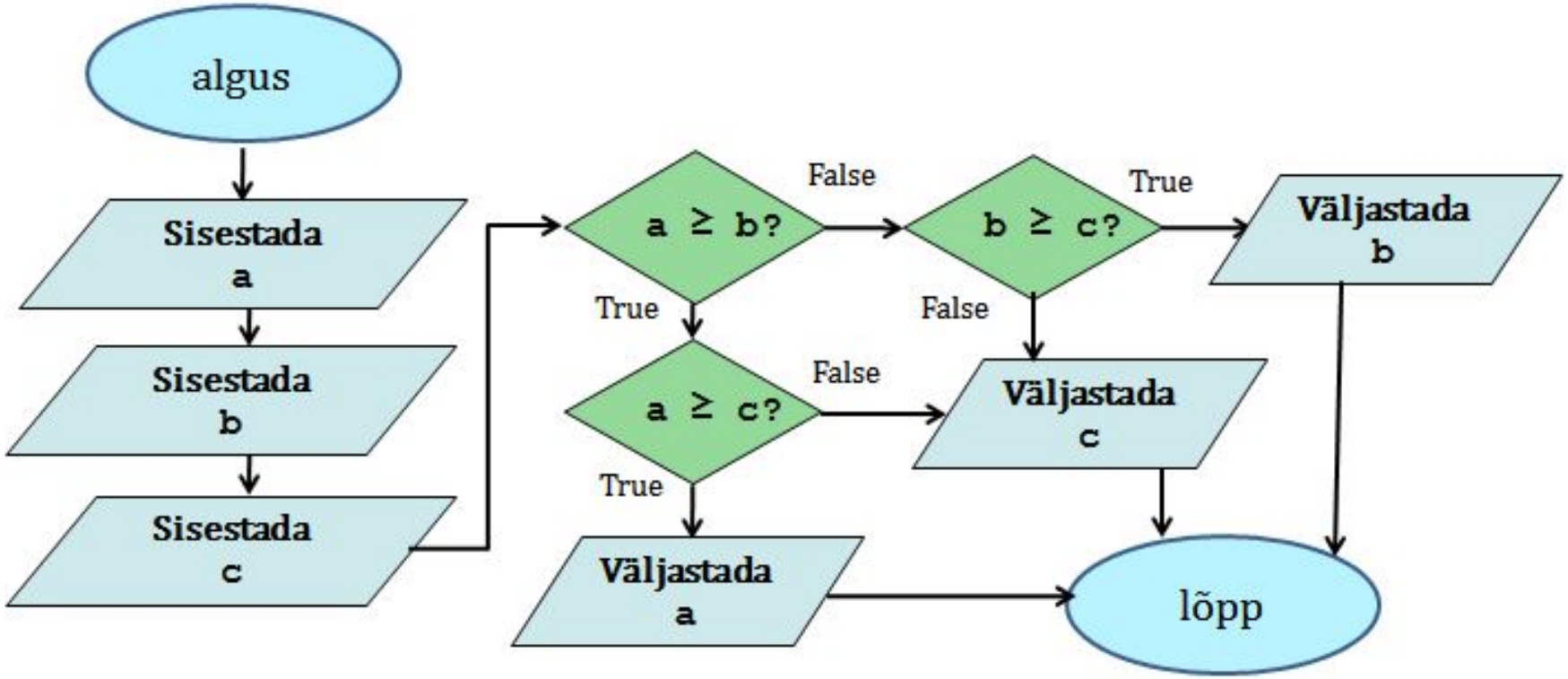



h_kujunemine(punkte)

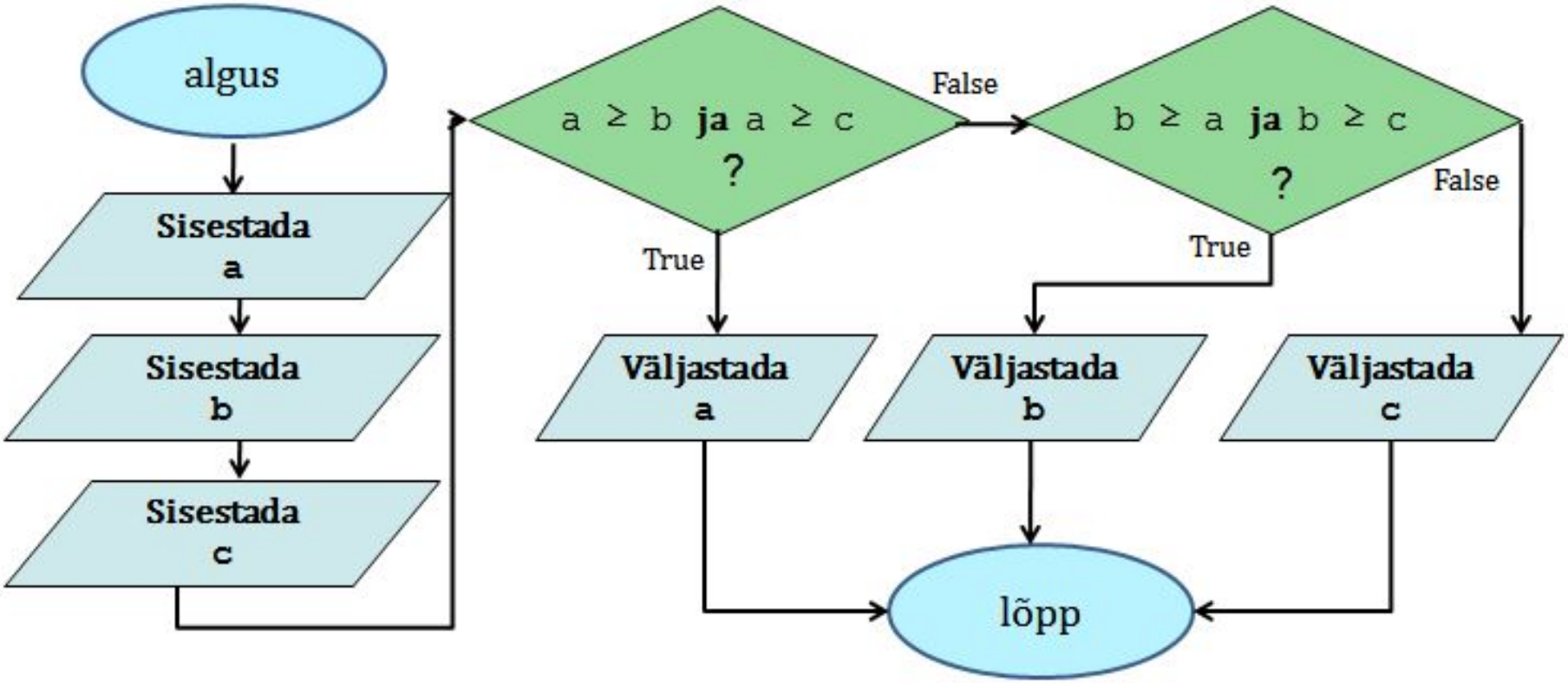
# Praktikumid



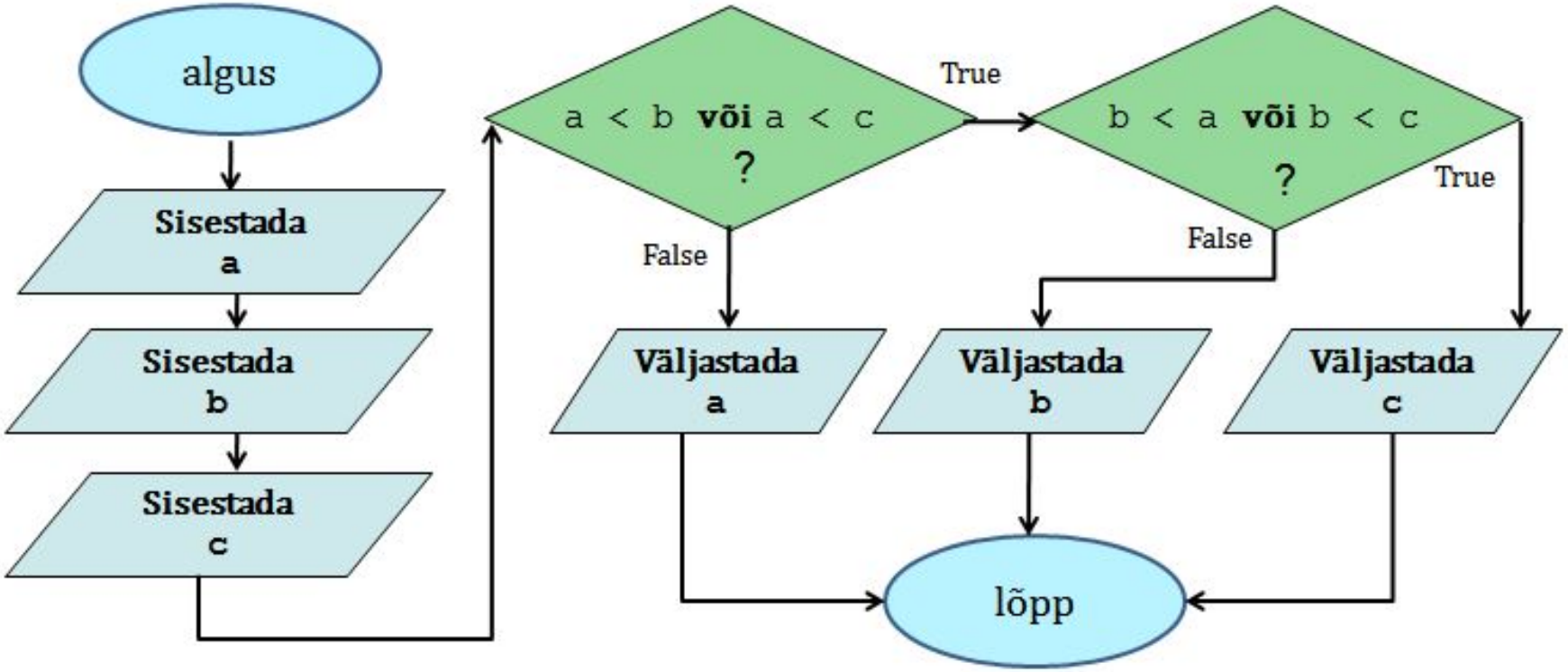
# Kolmest arvust suurim - I



# Kolmest arvust suurim - II

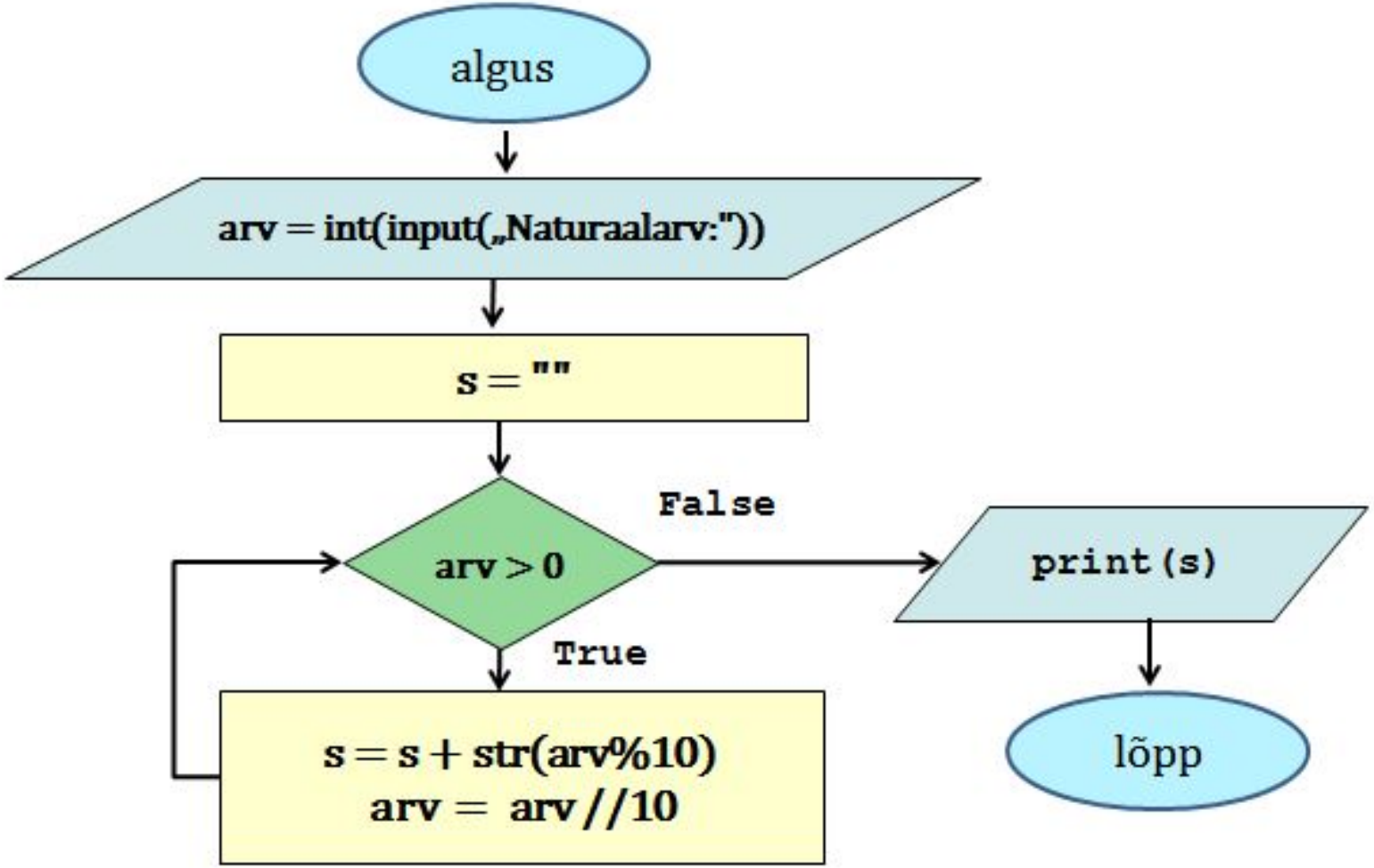


# Kolmest arvust suurim - III



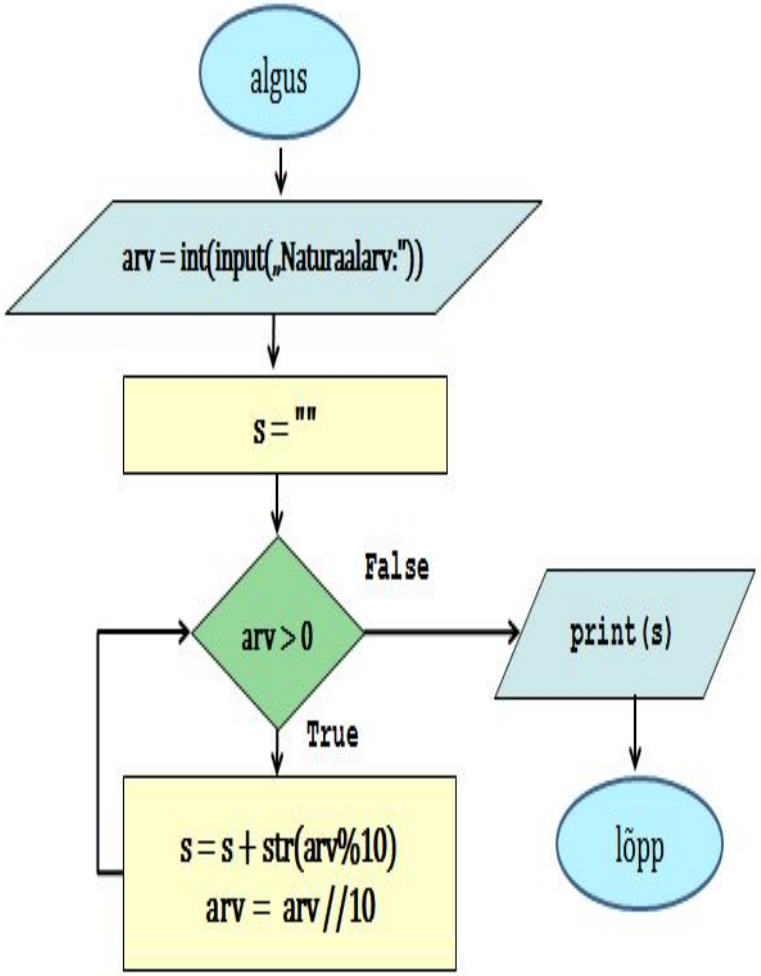
# Plokkskeem - III

Kasutaja sisestab 1324. Mida trükitakse?



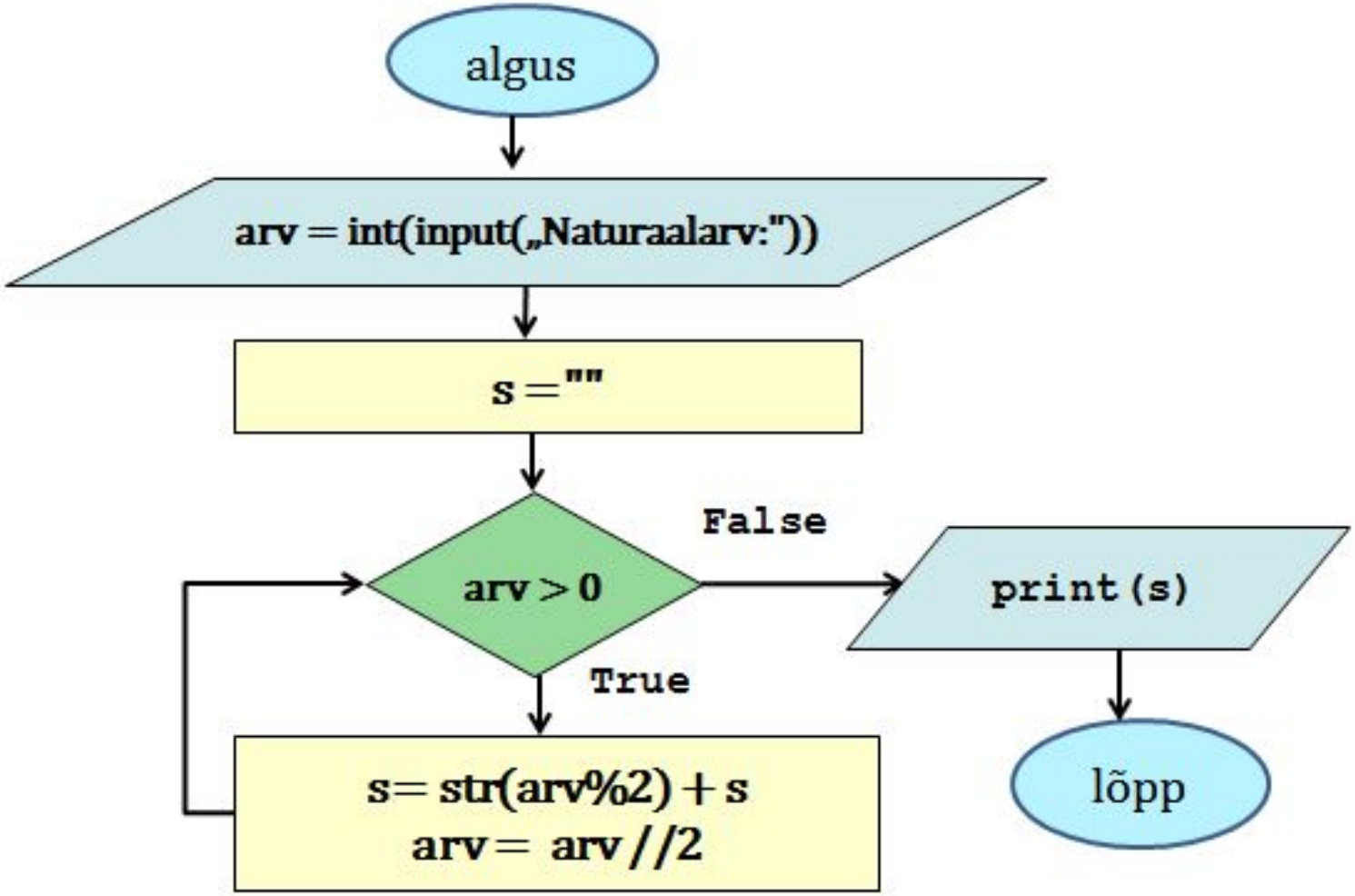
# Plokkskeem - III

```
arv = int(input('Naturaalarv: '))
s = ""
while arv > 0:
    s = s + str(arv%10)
    arv //= 10
print(s)
```



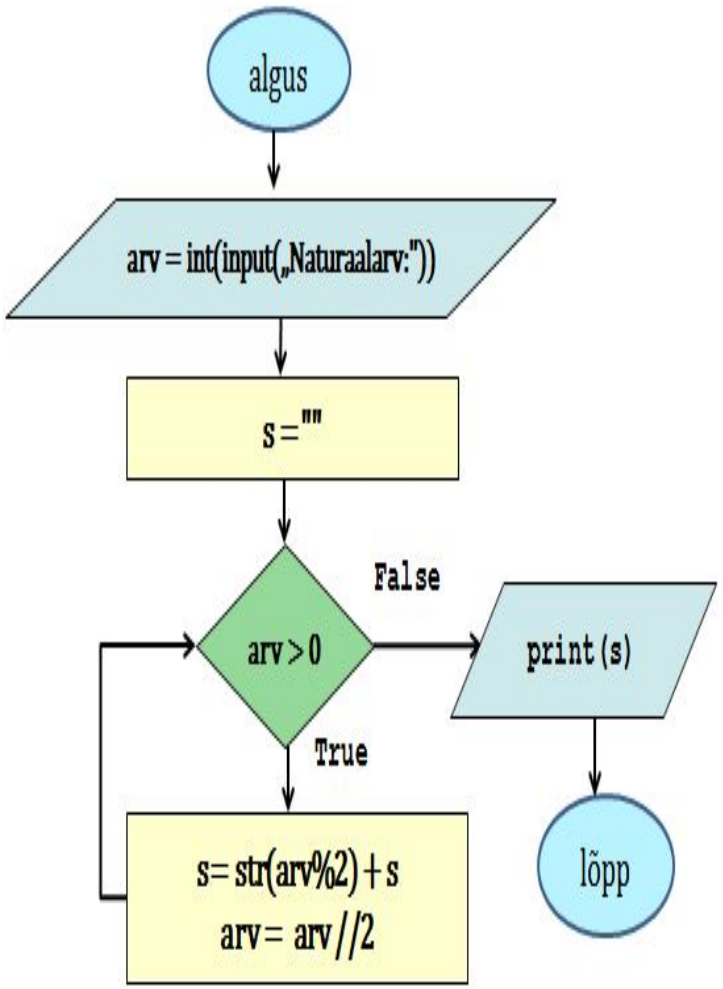
# Plokkskeem - IV

Kasutaja sisestab 25. Mida trükitakse?



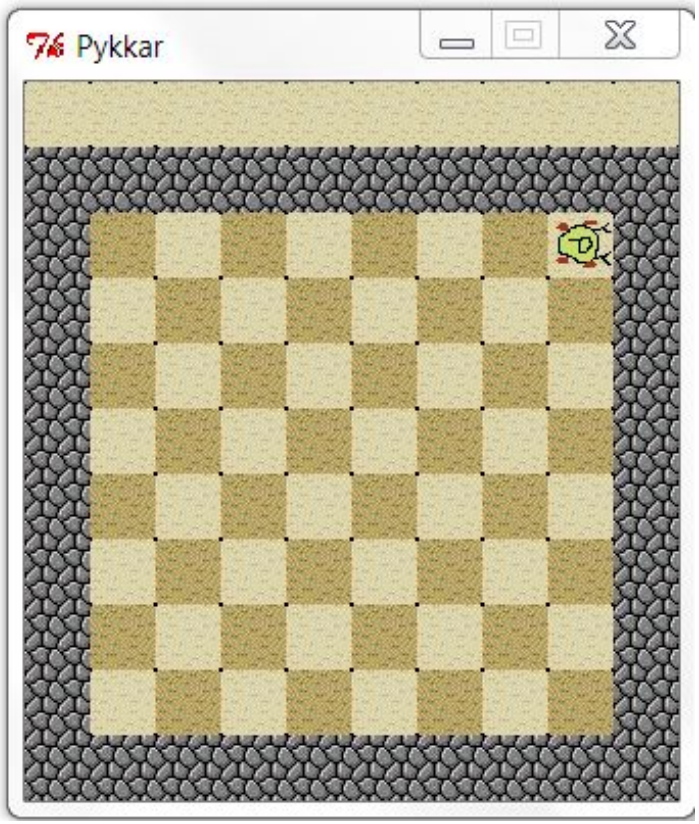
# Plokkskeem - IV

```
arv = int(input('Naturaalarv: '))
s = ""
while arv > 0:
    s = str(arv%2) + s
    arv //= 2
print(s)
```



# Pykkar

- **step()**
- **right()**
- **paint()**
- **push()**
- **take()**
- **put()**
- **is_wall()**
- **is_box()**
- **is_cone()**
- **is_painted()**
- ...



# Ülesanne

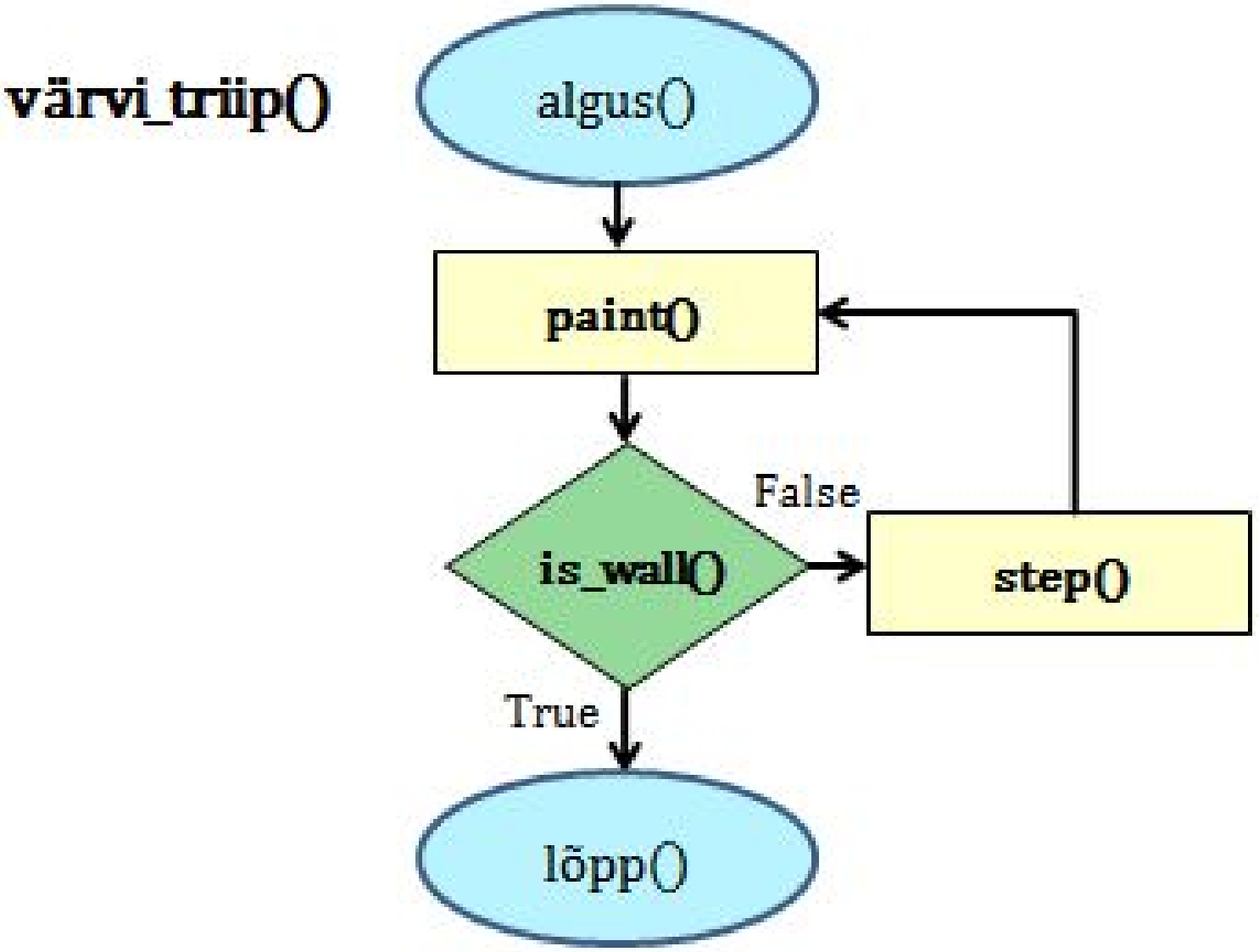
Koostada plokkskeem, mis paneb pykkari värvima põranda põhja-lõuna suunas triibuliseks.

# Lahendus

Kuidas pykkar liigub?

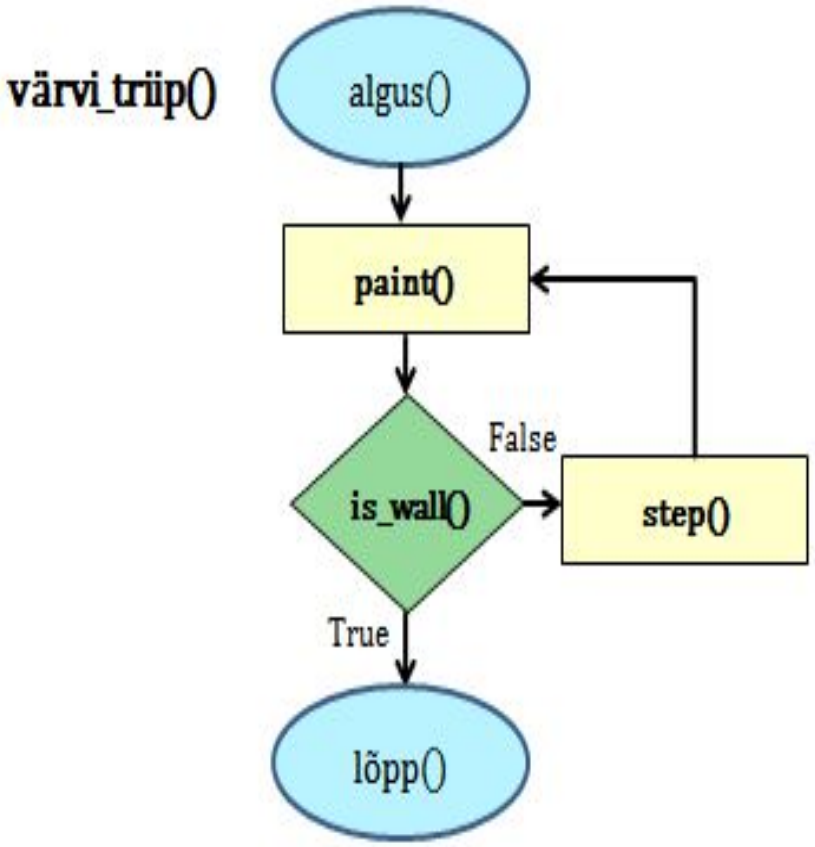
1. Pykkar värvib veeru põhjast lõunasse ja liigub tuldud teed tagasi.
2. Pükkar pöörab paremale ja üritab astuda kaks sammu.
3. Kui õnnestub (seina veel pole), siis Pükkar pöörab paremale ja jätkab punktist 1.
4. Kui ei õnnestu kahte sammu astuda, siis Pükkar lõpetab.

# Pykkar värvib triibu

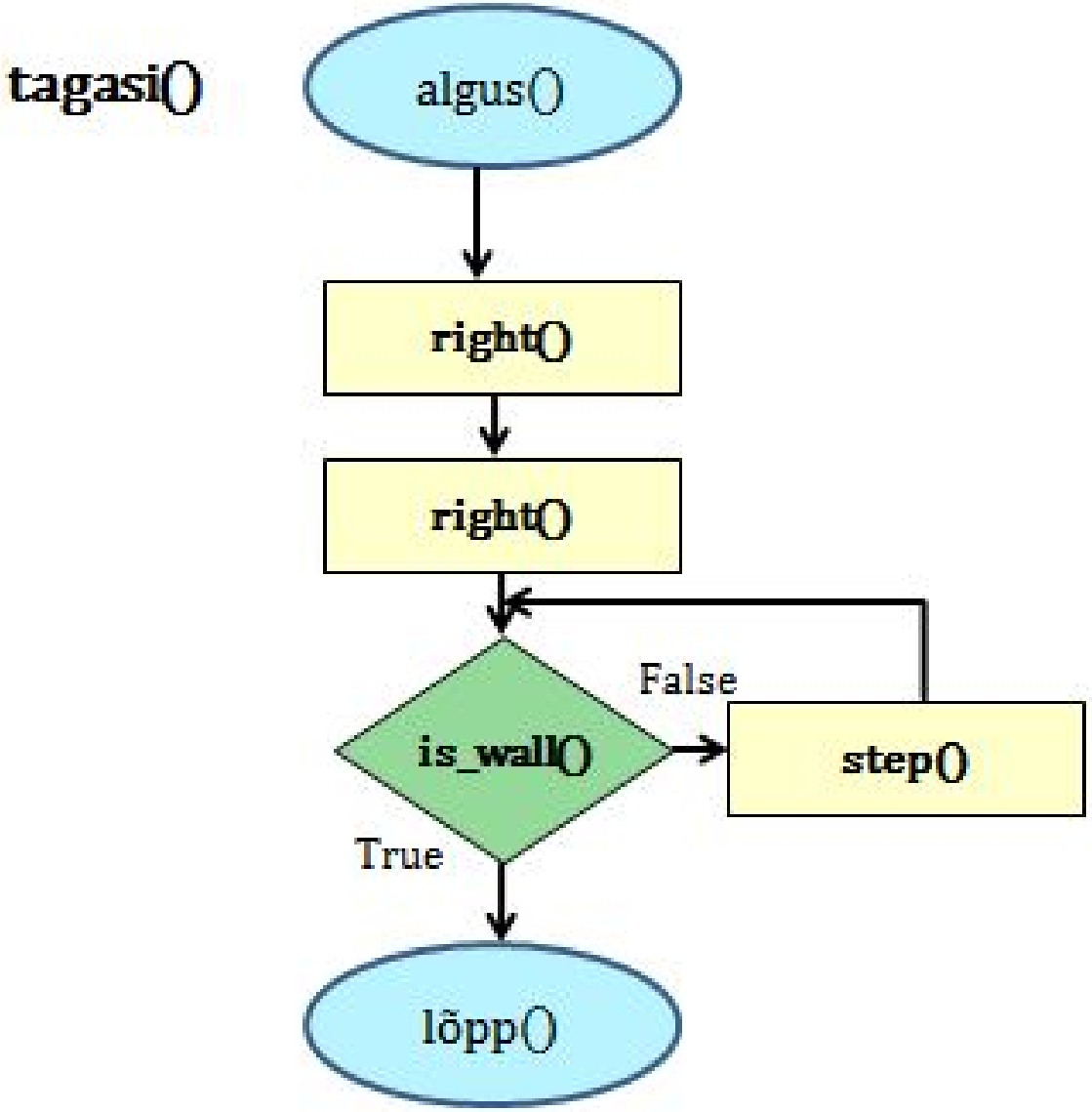


# Pykkar värvib triibu

```
def varvi_triip ():
    while True:
        paint()
        if not is_wall():
            step()
        else:
            break
```

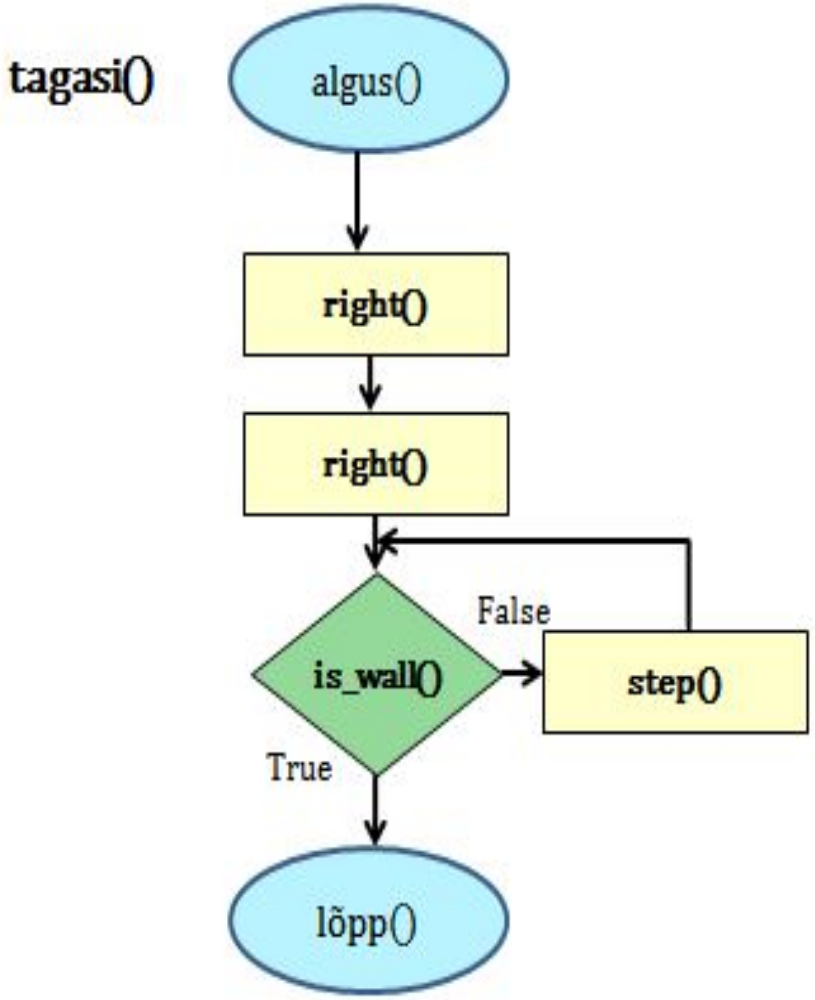


# Pykkar liigub tagasi

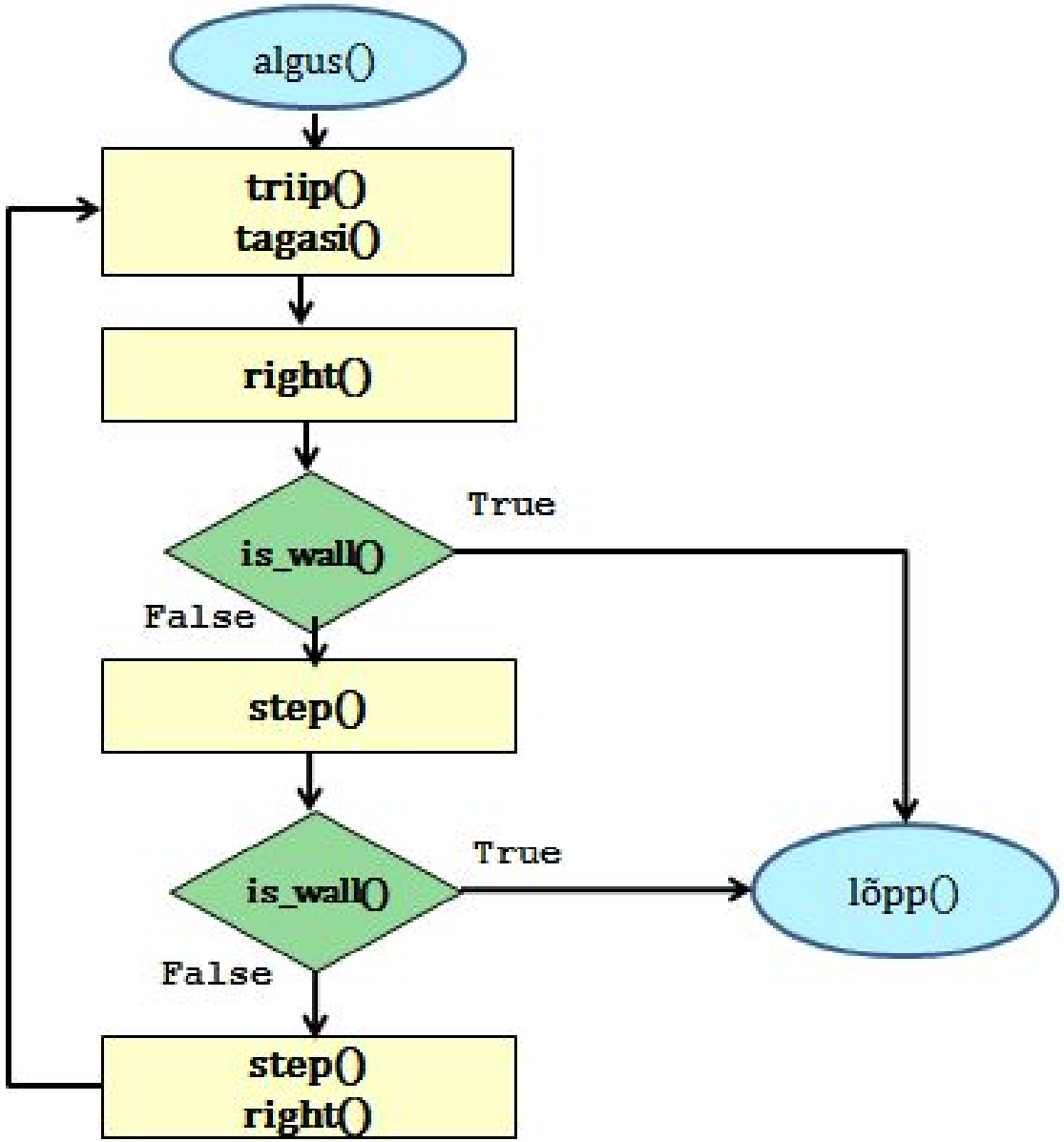


# Pykkar liigub tagasi

```
def tagasi():
    right()
    right()
    while True:
        if not is_wall():
            step()
        else:
            break
```

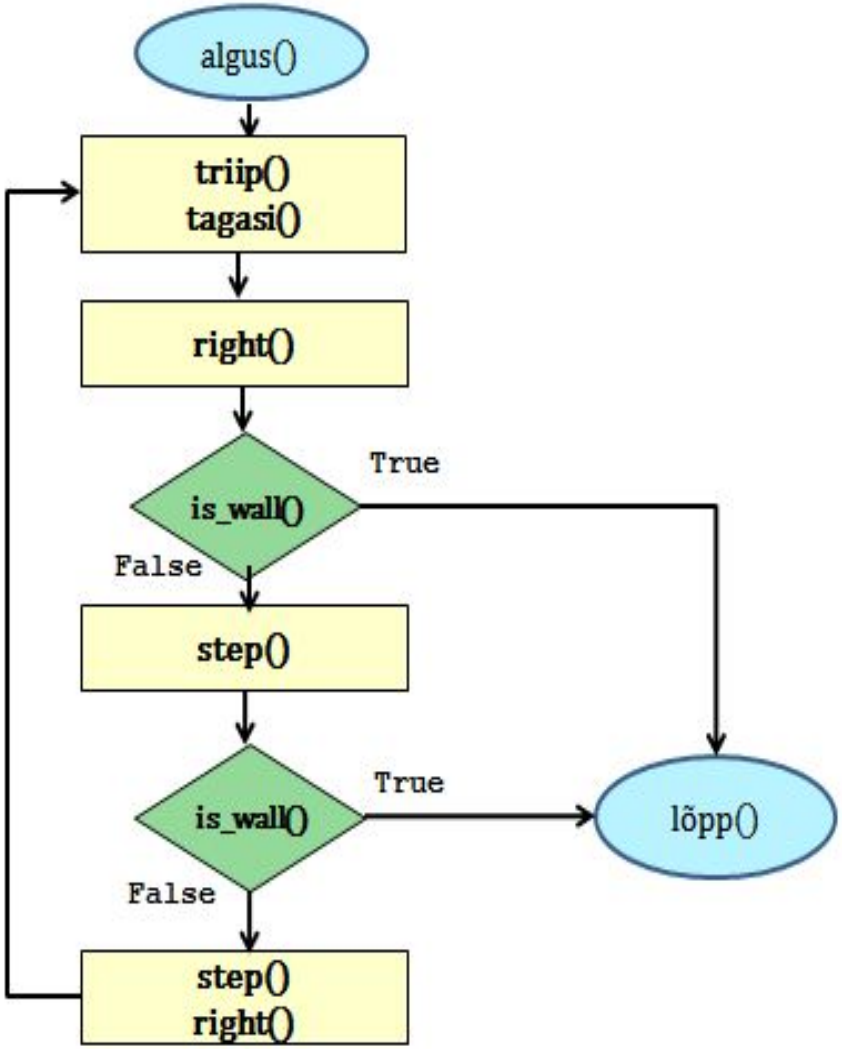


# Põhiprogramm

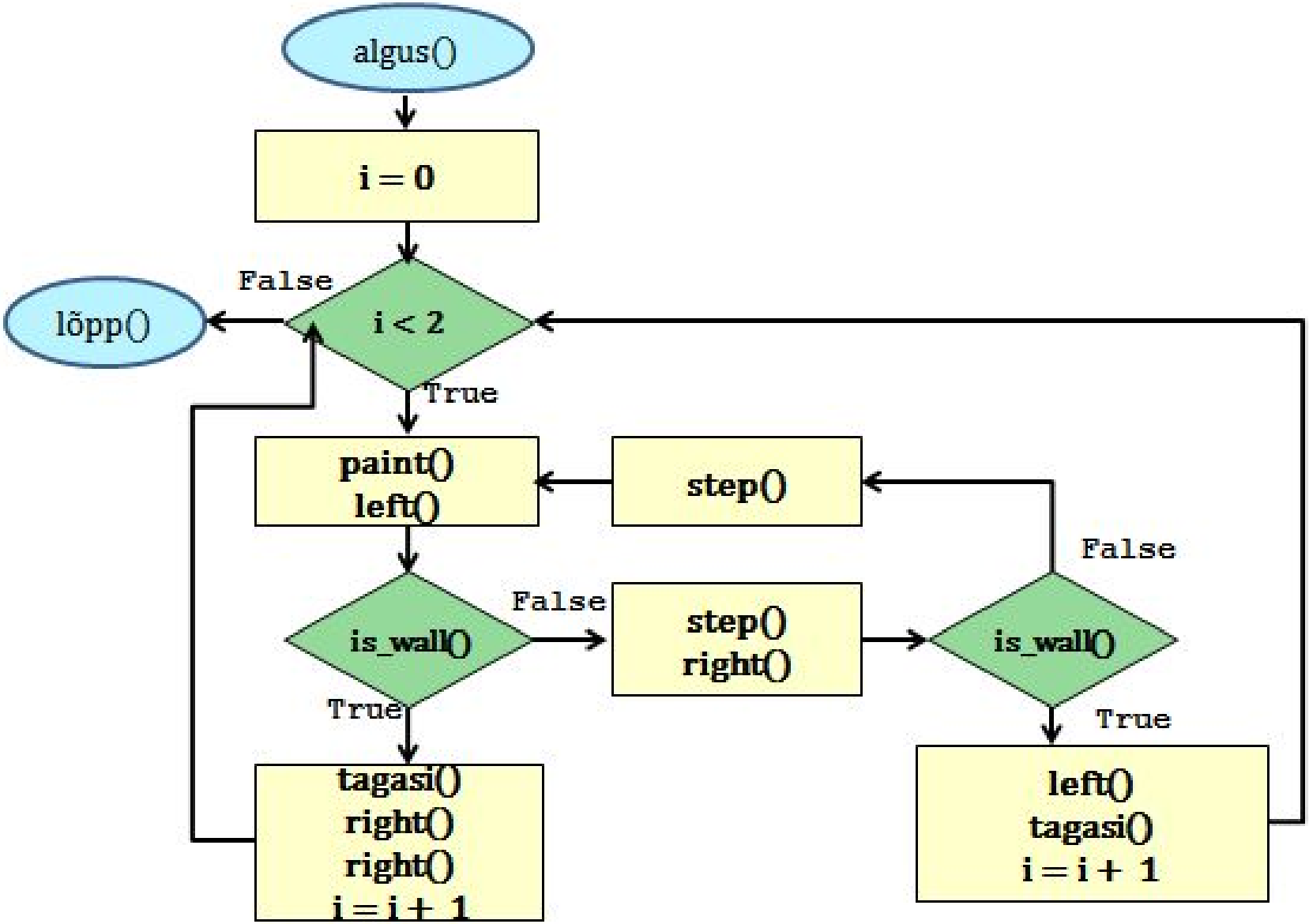


# Põhiprogramm

```
while True:
    triip ()
    tagasi ()
    right ()
    if is_wall ():
        break
    else:
        step()
        if is_wall ():
            break
        else:
            step()
            right ()
```



# Mida pykkar teeb?



Suur tänu osalemast

ja

kohtumiseni!