1. uzdevums

Sastādīt programmu, kas nodrošina datora "iedomata" naturāla skaitļa no 1 līdz N minēšanu. Skaitli N ievada lietotajs.

Kods:

```
# Programmas nosaukums: 1. uzd MPR11

# 1. uzdevums MPR11

# Uzdevuma formulējums: Sastādīt programmu, kas nodrošina datora "iedomata" naturāla skaitļa no 1 līdz N minēšanu. Skaitli N ievada lietotajs.

# Versija 1.0


import random


print("Spēlē:\nDators 'iedomāsies' veselo skaitli no 1 līdz N un Jums būs nepieciešāms to atminēt.")

N = int(input("Ievādiet veselo skaitli N ==> "))



x = random.randint(1, N)

sk=1

y=0


y = int(input("Mini skaitļi! ==> "))


while y != x:

   if y < x:

      y = int(input("Neatminēji! Ievadi lielāko ==> "))

   elif y > x:

      y = int(input("Neatminēji! Ievadi mazako ==> "))

   sk = sk+1


print("Malacis! Atminēji skaitļi " + str(x) + " intervāla no 1 līdz " + str(N) + " no " + str(sk) + " reizes")
```

Testa piemēri:

1)

```
Spēlē:
Dators 'iedomāsies' veselo skaitli no 1 līdz N un Jums būs nepieciešāms to atminēt.
Ievādiet veselo skaitli N ==> 10
Mini skaitļi! ==> 5
Neatminēji! Ievadi mazako ==> 3
Neatminēji! Ievadi lielāko ==> 4
Malacis! Atminēji skaitļi 4 intervāla no 1 līdz 10 no 3 reizes
```

2)

```
Spēlē:
Dators 'iedomāsies' veselo skaitli no 1 līdz N un Jums būs nepieciešāms to atminēt.
Ievādiet veselo skaitli N ==> 100
Mini skaitļi! ==> 50
Neatminēji! Ievadi lielāko ==> 75
Neatminēji! Ievadi mazako ==> 62
Neatminēji! Ievadi lielāko ==> 69
Neatminēji! Ievadi lielāko ==> 72
Neatminēji! Ievadi lielāko ==> 74
Malacis! Atminēji skaitļi 74 intervāla no 1 līdz 100 no 6 reizes
```

3)

```
Spēlē:
Dators 'iedomāsies' veselo skaitli no 1 līdz N un Jums būs nepieciešāms to atminēt.
Ievādiet veselo skaitli N ==> 20
Mini skaitļi! ==> 1
Neatminēji! Ievadi lielāko ==> 2
Neatminēji! Ievadi lielāko ==> 3
Neatminēji! Ievadi lielāko ==> 4
Neatminēji! Ievadi lielāko ==> 5
Neatminēji! Ievadi lielāko ==> 3
Neatminēji! Ievadi lielāko ==> 3
Neatminēji! Ievadi lielāko ==> 3
Neatminēji! Ievadi lielāko ==> 2
Neatminēji! Ievadi lielāko ==> 1
Neatminēji! Ievadi lielāko ==> 0
Neatminēji! Ievadi lielāko ==> 6
Neatminēji! Ievadi lielāko ==> 7
Neatminēji! Ievadi lielāko ==> 8
Neatminēji! Ievadi lielāko ==> 9
Neatminēji! Ievadi lielāko ==> 20
Neatminēji! Ievadi mazako ==> 19
Neatminēji! Ievadi mazako ==> 18
Neatminēji! Ievadi mazako ==> 17
Neatminēji! Ievadi mazako ==> 16
Neatminēji! Ievadi mazako ==> 15
Neatminēji! Ievadi mazako ==> 14
Neatminēji! Ievadi mazako ==> 6
Neatminēji! Ievadi lielāko ==> 8
Neatminēji! Ievadi lielāko ==> 0
Neatminēji! Ievadi lielāko ==> 1
Neatminēji! Ievadi lielāko ==> 5
Neatminēji! Ievadi lielāko ==> 6
Neatminēji! Ievadi lielāko ==> 18
Neatminēji! Ievadi mazako ==> 17
Neatminēji! Ievadi mazako ==> 16
Neatminēji! Ievadi mazako ==> 15
Neatminēji! Ievadi mazako ==> 14
Neatminēji! Ievadi mazako ==> 13
Malacis! Atminēji skaitļi 13 intervāla no 1 līdz 20 no 34 reizes
```

4)

```
Spēlē:
Dators 'iedomāsies' veselo skaitli no 1 līdz N un Jums būs nepieciešāms to atminēt.
Ievādiet veselo skaitli N ==> 10
Mini skaitļi! ==> 1
Malacis! Atminēji skaitļi 1 intervāla no 1 līdz 10 no 1 reizes
```

5)

```
Spēlē:
Dators 'iedomāsies' veselo skaitli no 1 līdz N un Jums būs nepieciešāms to atminēt.
Ievādiet veselo skaitli N ==> 1000
Mini skaitļi! ==> 500
Neatminēji! Ievadi mazako ==> 250
Neatminēji! Ievadi mazako ==> 125
Neatminēji! Ievadi mazako ==> 75
Neatminēji! Ievadi lielāko ==> 100
Neatminēji! Ievadi lielāko ==> 110
Neatminēji! Ievadi lielāko ==> 114
Neatminēji! Ievadi lielāko ==> 120
Neatminēji! Ievadi lielāko ==> 124
Neatminēji! Ievadi mazako ==> 123
Malacis! Atminēji skaitļi 123 intervāla no 1 līdz 1000 no 10 reizes
```

1. papilduzdevums (PU1)

Sastādīt programmu, kas nodrošina datora "iedomata" naturāla skaitļa no 1 līdz N minēšanu. Skaitli N ievada lietotajs.

Kods:

Programmas nosaukums: 1. uzd PU1 MPR11

1. uzd PU1 MPR11

Uzdevuma formulējums: Sastādīt programmu, kas nodrošina, ka dators pēc iepriekš aprakstītā algoritma min lietotāja iedomāto skaitli no 1 līdz N minēšanu. Skaitli N ievada lietotājs. Minēšana uz labu laimi - 1 punkts, minēšana ar mazāko nepareizo atbilžu skaitu (vispārīgajā gadījumā) - 3 punkti.

Versija 1.0

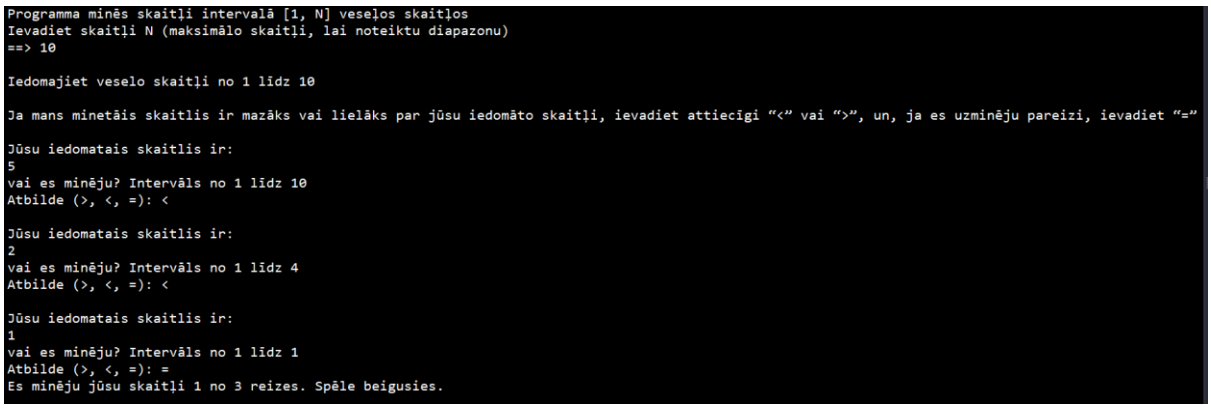
```
import random

import math



min_diapazons = 1
```

```
try:
    max_diapazons = int(input("Programma minēs skaitļi intervalā [1, N] veseļos skaitļos \nIevadiet skaitļi N (maksimālo skaitļi, lai noteiktu diapazonu) \n==> "))
    print("\nIedomajiet veselo skaitļi no 1 līdz " + str(max_diapazons))
    print("\nJa mans minetāis skaitlis ir mazāks vai lielāks par jūsu iedomāto skaitļi, ievadiet attiecīgi “<” vai “>”, un, ja es uzminēju pareizi, ievadiet “=”")

except ValueError:
    print("Nepareiza ievade! Ievadiet veselo skaitļi!")
    quit(0)

rnd = math.floor(max_diapazons/2)
sk = 1
atbilde = ""

while 1 > 0:
    print("\nJūsu iedomatais skaitlis ir:\n" + str(rnd) + "\nvai es minēju? Intervāls no", min_diapazons, "līdz", max_diapazons)

    atbilde = input("Atbilde (>, <, =): ")

    if atbilde == "=":
        print("Es minēju jūsu skaitļi " + str(rnd) + " no " + str(sk) + " reizes. Spēle beigusies.")
        break

    elif atbilde == "<":
        max_diapazons = rnd - 1

        try:
            rnd = random.randint(min_diapazons, max_diapazons)
```

```
            sk = sk + 1
        except:
            print("Tā nevar būt! Tu melo! Es ar tevi vairs nespēlēšos!")
            break

    elif atbilde == ">":
        min_diapazons = rnd + 1

        try:
            rnd = random.randint(min_diapazons, max_diapazons)
            sk = sk + 1
        except:
            print("Tā nevar būt! Tu melo! Es ar tevi vairs nespēlēšos!")
            break

    else:
        print("Nepareiza ievade! Ievadiet: '<', '>' vai '='")
```

Testa piemēri:

1)



2)

```
Programma minēs skaitļi intervalā [1, N] veseļos skaitļos
Ievadiet skaitļi N (maksimālo skaitļi, lai noteiktu diapazonu)
==> 15

Iedomajiet veselo skaitļi no 1 līdz 15

Ja mans minetāis skaitlis ir mazāks vai lielāks par jūsu iedomāto skaitļi, ievadiet attiecīgi "<" vai ">", un, ja es uzminēju pareizi, ievadiet "="

Jūsu iedomatais skaitlis ir:
7
vai es minēju? Intervāls no 1 līdz 15
Atbilde (>, <, =): <

Jūsu iedomatais skaitlis ir:
4
vai es minēju? Intervāls no 1 līdz 6
Atbilde (>, <, =): <

Jūsu iedomatais skaitlis ir:
1
vai es minēju? Intervāls no 1 līdz 3
Atbilde (>, <, =): >

Jūsu iedomatais skaitlis ir:
2
vai es minēju? Intervāls no 2 līdz 3
Atbilde (>, <, =): >

Jūsu iedomatais skaitlis ir:
3
vai es minēju? Intervāls no 3 līdz 3
Atbilde (>, <, =): <
Tā nevar būt! Tu melo! Es ar tevi vairs nespēlēšos!
```

3)

```
Programma minēs skaitļi intervalā [1, N] veseļos skaitļos
Ievadiet skaitļi N (maksimālo skaitļi, lai noteiktu diapazonu)
==> 100

Iedomajiet veselo skaitļi no 1 līdz 100

Ja mans minetāis skaitlis ir mazāks vai lielāks par jūsu iedomāto skaitļi, ievadiet attiecīgi "<" vai ">", un, ja es uzminēju pareizi, ievadiet "="

Jūsu iedomatais skaitlis ir:
50
vai es minēju? Intervāls no 1 līdz 100
Atbilde (>, <, =): =
Es minēju jūsu skaitļi 50 no 1 reizes. Spēle beigusies.
```

4)

```
Programma minēs skaitļi intervalā [1, N] veseļos skaitļos
Ievadiet skaitļi N (maksimālo skaitļi, lai noteiktu diapazonu)
==> 1000

Iedomajiet veselo skaitļi no 1 līdz 1000

Ja mans minetāis skaitlis ir mazāks vai lielāks par jūsu iedomāto skaitļi, ievadiet attiecīgi "<" vai ">", un, ja es uzminēju pareizi, ievadiet "="

Jūsu iedomatais skaitlis ir:
500
vai es minēju? Intervāls no 1 līdz 1000
Atbilde (>, <, =): >

Jūsu iedomatais skaitlis ir:
772
vai es minēju? Intervāls no 501 līdz 1000
Atbilde (>, <, =): <

Jūsu iedomatais skaitlis ir:
589
vai es minēju? Intervāls no 501 līdz 771
Atbilde (>, <, =): >

Jūsu iedomatais skaitlis ir:
691
vai es minēju? Intervāls no 590 līdz 771
Atbilde (>, <, =): <

Jūsu iedomatais skaitlis ir:
630
vai es minēju? Intervāls no 590 līdz 690
Atbilde (>, <, =): >

Jūsu iedomatais skaitlis ir:
647
vai es minēju? Intervāls no 631 līdz 690
Atbilde (>, <, =): >

Jūsu iedomatais skaitlis ir:
665
vai es minēju? Intervāls no 648 līdz 690
Atbilde (>, <, =): >

Jūsu iedomatais skaitlis ir:
676
vai es minēju? Intervāls no 666 līdz 690
Atbilde (>, <, =): <

Jūsu iedomatais skaitlis ir:
675
vai es minēju? Intervāls no 666 līdz 675
Atbilde (>, <, =): <

Jūsu iedomatais skaitlis ir:
670
vai es minēju? Intervāls no 666 līdz 674
Atbilde (>, <, =): <

Jūsu iedomatais skaitlis ir:
667
vai es minēju? Intervāls no 666 līdz 669
Atbilde (>, <, =): <

Jūsu iedomatais skaitlis ir:
666
vai es minēju? Intervāls no 666 līdz 666
Atbilde (>, <, =): =
Es minēju jūsu skaitļi 666 no 12 reizes. Spēle beigusies.
```

5)

```
Programma minēs skaitļi intervalā [1, N] veseļos skaitļos
Ievadiet skaitļi N (maksimālo skaitļi, lai noteiktu diapazonu)
==> 7

Iedomajiet veselo skaitļi no 1 līdz 7

Ja mans minētāis skaitlis ir mazāks vai lielāks par jūsu iedomāto skaitļi, ievadiet attiecīgi “<” vai “>”, un, ja es uzminēju pareizi, ievadiet “=”

Jūsu iedomatais skaitlis ir:
3
vai es minēju? Intervāls no 1 līdz 7
Atbilde (>, <, =): 4
Nepareiza ievade! Ievadiet: '<', '>' vai '='

Jūsu iedomatais skaitlis ir:
3
vai es minēju? Intervāls no 1 līdz 7
Atbilde (>, <, =): 7
Nepareiza ievade! Ievadiet: '<', '>' vai '='

Jūsu iedomatais skaitlis ir:
3
vai es minēju? Intervāls no 1 līdz 7
Atbilde (>, <, =): >

Jūsu iedomatais skaitlis ir:
7
vai es minēju? Intervāls no 4 līdz 7
Atbilde (>, <, =): <

Jūsu iedomatais skaitlis ir:
6
vai es minēju? Intervāls no 4 līdz 6
Atbilde (>, <, =): <

Jūsu iedomatais skaitlis ir:
5
vai es minēju? Intervāls no 4 līdz 5
Atbilde (>, <, =): >
Tā nevar būt! Tu melo! Es ar tevi vairs nespēlēšos!
```

6)

```
Programma minēs skaitļi intervalā [1, N] veseļos skaitļos
Ievadiet skaitļi N (maksimālo skaitļi, lai noteiktu diapazonu)
==> 7

Iedomajiet veselo skaitļi no 1 līdz 7

Ja mans minētāis skaitlis ir mazāks vai lielāks par jūsu iedomāto skaitļi, ievadiet attiecīgi “<” vai “>”, un, ja es uzminēju pareizi, ievadiet “=”

Jūsu iedomatais skaitlis ir:
3
vai es minēju? Intervāls no 1 līdz 7
Atbilde (>, <, =): <

Jūsu iedomatais skaitlis ir:
2
vai es minēju? Intervāls no 1 līdz 2
Atbilde (>, <, =): >
Tā nevar būt! Tu melo! Es ar tevi vairs nespēlēšos!
```

2. uzdevums

Sastādīt programmu, kas atrod 4 naturālo skaitļu lielāko kopīgo dalītāju un mazāko kopīgo dalāmo. Skaitļus ievada lietotājs.

Kods:

```
# Programmas nosaukums: 2. uzd MPR11

# 2. uzdevums MPR11

# Uzdevuma formulējums: Sastādīt programmu, kas atrod 4 naturālo skaitļu lielāko kopīgo dalītāju un mazāko kopīgo dalāmo. Skaitļus ievada lietotājs.

# Versija 1.0


#----------------


# Uzdevumu skicē:


# 4 skaitli:

# a b c d

# lcm (a, b) = f

# f c d

# lcm (f, c) = g

# g d

# lcm (g, d) = lcm (a, b, c, d)


# Lai atrāstu lcm diviem skaitliem:

# a b

# gcd (a, b) = c

# lcm (a b) = a * b / c

# gcd var atrāst pēc Eiklīda algoritma


#--------------------------
```

```
# Uzdevumu plāns


# Četri skaitli
# a b c d
# 1) gcd(a, b) = C
# 2) lcm (a, b) = a*b/C = f
# 3) gcd(f, c) = D
# 4) lcm(f, c) = f*c/D = g
# 5) gcd(g, d) = K
# 6) lcm(g, d) = g*d/K
# 7) lcm(g, d) = lcm(a, b, c, d) (tas vajag printēt)
# 8) lcm(a,b) = f
# 9) lcm(c,d) = n
# 10) lcm(a,b,c,d) = lcm(f,n) (tas vajag printēt)

import math

x = int(input("Ievadi 1.skaitli ===> ")) # skaitlis a
y = int(input("Ievadi 2.skaitli ===> ")) # skaitlis b

if x<=0 or y<=0:
    print("Ievadiet pozitivo veselo skaitli!")
    quit()

A1 = x
B1 = y

#----------
# aprēķini
# gcd (a, b) = C
# lcm (a, b) = f
```

```
a = x
b = y

while b != 0 :
   c = a % b
   a = b
   b = c

LKD12 = a       # gcd (a,b) = C
MKD12 = (x*y)/a # lcm (a, b) = a*b/C
f = MKD12       # lcm (a, b) = f

#----------

# aprēķini
# gcd(f, c) = D
# lcm(f, c) = g

x = int(input("Ievadi 3.skaitli ===> ")) # skaitlis c

if x<=0:
   print("Ievadiet pozitivo veselo skaitli!")
   quit()

C1 = x
y = f
a = x
b = y

while b != 0 :
```

```
    c = a % b
    a = b
    b = c

LKDfc = a        # gcd(f, c) = D
MKDfc = (x*y)/a  # lcm(f, c) = f*c/D = g
g = MKDfc        # lcm(f, c) = g
x = g



#----------

# aprēķini
# gcd(g, d) = K
# lcm(g, d) = lcm(a, b, c, d)

y = int(input("Ievadi 4.skaitli ===> ")) # skaitlis d

if y<=0:
    print("Ievadiet pozitivo veselo skaitli!")
    quit()

D1 = y

a = x
b = y

while b != 0 :
    c = a % b
    a = b
    b = c
```

```
LKDgd = a             # gcd(g, d) = K
MKDgd = int((x*y)/a)

#----------

print("\nMazākais kopīgais dalāmais MKD = " + str(MKDgd))
print("lcm(" + str(A1) + ", " + str(B1) + ", " + str(C1) + ", " + str(D1) + ") = " + str(MKDgd) + "\n")

#----------
# aprēķini
# lcm(c,d) = n
# lcm(c,d) = n

a = C1
b = D1

while b != 0 :
   c = a % b
   a = b
   b = c



LKD34 = a # lcm(c,d) = n

a = LKD12 # lcm(a,b) = f
b = LKD34 # lcm(c,d) = n

#----------
# aprēķini
```

```
# lcm(a,b,c,d) = lcm(f,n)



while b != 0 :
   c = a % b
   a = b
   b = c


LKDkop = a    # lcm(a,b,c,d) = lcm(f,n)


#----------


print("Lielākais kopīgais dalītājs LKD = " + str(LKDkop))
print("gcd(" + str(A1) + ", " + str(B1) + ", " + str(C1) + ", " + str(D1) + ") = " + str(LKDkop) + "\n")
```

Testa piemēri:

1)

```
Ievadi 1.skaitli ===> 5
Ievadi 2.skaitli ===> 10
Ievadi 3.skaitli ===> 15
Ievadi 4.skaitli ===> 20

Mazākais kopīgais dalāmais MKD = 60
lcm(5, 10, 15, 20) = 60

Lielākais kopīgais dalītājs LKD = 5
gcd(5, 10, 15, 20) = 5
```

2)

```
Ievadi 1.skaitli ===> 1
Ievadi 2.skaitli ===> 1
Ievadi 3.skaitli ===> 3
Ievadi 4.skaitli ===> 6

Mazākais kopīgais dalāmais MKD = 6
lcm(1, 1, 3, 6) = 6

Lielākais kopīgais dalītājs LKD = 1
gcd(1, 1, 3, 6) = 1
```

3)

```
Ievadi 1.skaitli ===> 5
Ievadi 2.skaitli ===> 10
Ievadi 3.skaitli ===> 0
Ievadiet pozitivo veselo skaitli!
```

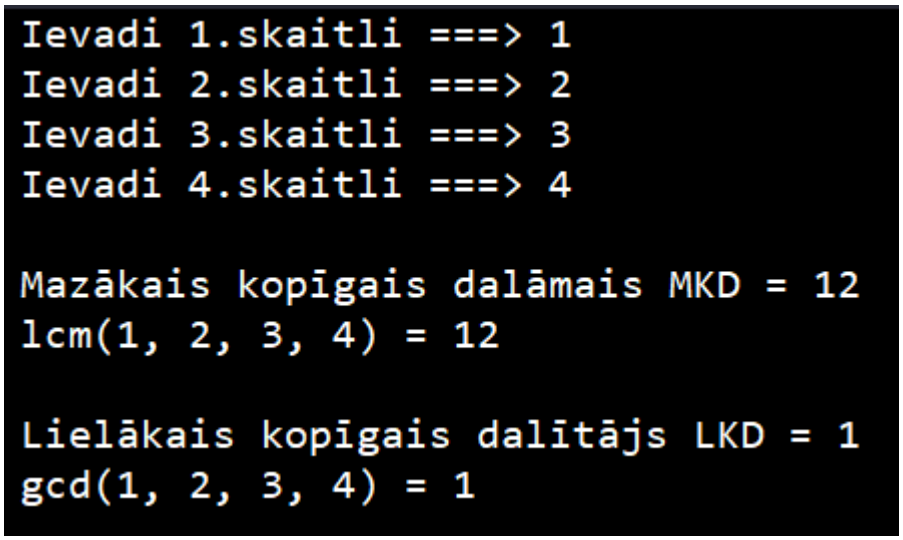
4)

```
Ievadi 1.skaitli ===> 4
Ievadi 2.skaitli ===> 10
Ievadi 3.skaitli ===> 23
Ievadi 4.skaitli ===> -12
Ievadiet pozitivo veselo skaitli!
```

5)

```
Ievadi 1.skaitli ===> 2
Ievadi 2.skaitli ===> 4
Ievadi 3.skaitli ===> 6
Ievadi 4.skaitli ===> 8

Mazākais kopīgais dalāmais MKD = 24
lcm(2, 4, 6, 8) = 24

Lielākais kopīgais dalītājs LKD = 2
gcd(2, 4, 6, 8) = 2
```

6)



3. uzdevums

Sastādīt programmu, kas nosaka mazāko Fibonači skaitli, kas pārsniedz lietotāja ievadīto skaitli N.

Kods:

```
# Programmas nosaukums: 3. uzd MPR11

# 3. uzdevums MPR11

# Uzdevuma formulējums: Sastādīt programmu, kas nosaka mazāko Fibonači skaitli, kas pārsniedz lietotāja ievadīto skaitli N.

# Versija 1.0


print("Programma nosaka mazāko Fibonači skaitli (skaitli no Fibonači virknes), kas pārsniedz lietotāja ievadīto skaitli N.")

n = float(input("Ievadiet skaitli N ==> "))


b=1

a=1

c=0       # a,b=b,a+b (ja to izmantojam, tad nevajadzēs izmantot palīgmainīgo c)


while n>=a: # kāmer n>=a, tad


        # a,b=b,a+b
```

```
        # var arī to konstrukciju izmantot
    c = a + b
    a = b
    b = c

print("Mazakais Fibonači skaitlis, kas pārsniedz skaitli " + str(n) + " ir " + str(a))
```

Testa piemēri:

1)

```
Programma nosaka mazāko Fibonači skaitli (skaitli no Fibonači virknes), kas pārsniedz lietotāja ievadīto skaitli N.
Ievadiet skaitli N ==> 13
Mazakais Fibonači skaitlis, kas pārsniedz skaitli 13.0 ir 21
```

2)

```
Programma nosaka mazāko Fibonači skaitli (skaitli no Fibonači virknes), kas pārsniedz lietotāja ievadīto skaitli N.
Ievadiet skaitli N ==> 12.999
Mazakais Fibonači skaitlis, kas pārsniedz skaitli 12.999 ir 13
```

3)

```
Programma nosaka mazāko Fibonači skaitli (skaitli no Fibonači virknes), kas pārsniedz lietotāja ievadīto skaitli N.
Ievadiet skaitli N ==> -1000
Mazakais Fibonači skaitlis, kas pārsniedz skaitli -1000.0 ir 1
```

4)

```
Programma nosaka mazāko Fibonači skaitli (skaitli no Fibonači virknes), kas pārsniedz lietotāja ievadīto skaitli N.
Ievadiet skaitli N ==> 1
Mazakais Fibonači skaitlis, kas pārsniedz skaitli 1.0 ir 2
```

5)

```
Programma nosaka mazāko Fibonači skaitli (skaitli no Fibonači virknes), kas pārsniedz lietotāja ievadīto skaitli N.
Ievadiet skaitli N ==> 100
Mazakais Fibonači skaitlis, kas pārsniedz skaitli 100.0 ir 144
```

6)

```
Programma nosaka mazāko Fibonači skaitli (skaitli no Fibonači virknes), kas pārsniedz lietotāja ievadīto skaitli N.
Ievadiet skaitli N ==> 1000000
Mazakais Fibonači skaitlis, kas pārsniedz skaitli 1000000.0 ir 1346269
```

4. uzdevums

Sastādit programmu, kas atrod N ciparu pēc kārtas bez atstarpēm uzrakstītajā Fibonači skaitļu virknē. Skaitli N ievada lietotājs.

Kods:

```
# Programmas nosaukums: 4. uzd MPR11

# 4. uzdevums MPR11

# Uzdevuma formulējums: Sastādit programmu, kas atrod N ciparu pēc kārtas bez atstarpēm uzrakstītajā Fibonači skaitļu virknē. Skaitli N ievada lietotājs.

# Versija 1.0


n = int(input("Ievadiet skaitli N ==> "))


if n<=0:
    print("Neēksistē cipars ar tādu numuru")
    quit()


a=1
b=1
c=0


sv = str(a) + str(b)


while len(sv) < n:
    c = a+b
    a=b
    b=c
    sv = sv+str(c)


print("Fibonači virknē zem numura " + str(n) + " atrodas cipars:")
print(sv[n-1])
```

Testa piemēri:

1)

```
Ievadiet skaitli N ==> -5
Neēksistē cipars ar tādu numuru
```

2)

```
Ievadiet skaitli N ==> 0
Neēksistē cipars ar tādu numuru
```

3)

```
Ievadiet skaitli N ==> 1
Fibonači virknē zem numura 1 atrodas cipars:
1
```

4)

```
Ievadiet skaitli N ==> 2
Fibonači virknē zem numura 2 atrodas cipars:
1
```

5)

```
Ievadiet skaitli N ==> 10
Fibonači virknē zem numura 10 atrodas cipars:
1
```

6)

```
Ievadiet skaitli N ==> 5
Fibonači virknē zem numura 5 atrodas cipars:
5
```

7)

```
Ievadiet skaitli N ==> 6
Fibonači virknē zem numura 6 atrodas cipars:
8
```

8)

```
Ievadiet skaitli N ==> 12
Fibonači virknē zem numura 12 atrodas cipars:
4
```

9)

```
Ievadiet skaitli N ==> 666
Fibonači virknē zem numura 666 atrodas cipars:
2
```

5. uzdevums

Sastādīt programmu, kas noskaidro, vai lietotāja secīgi ievadītā skaitļu virkne veido aritmētisko progresiju. Skaitļu virknes ievadi pārtrauc ievadot skaitli 0.

Kods:

```
# Programmas nosaukums: 5. uzd MPR11

# 5. uzdevums MPR11

# Uzdevuma formulējums: Sastādīt programmu, kas noskaidro, vai lietotāja secīgi ievadītā skaitļu virkne veido aritmētisko progresiju. Skaitļu virknes ievadi pārtrauc ievadot skaitli 0.

# Versija 1.0


sv = "Dota virkne ir aritmētiska progresija"


x1 = float(input("Ievadiet 1 locekli ===> "))


if x1 == 0:
    print("Šajā virknē nav elementu.") # Ja uzreiz 0 ievada
    quit()


x2 = float(input("Ievadiet 2 locekli ===> "))


if x2 == 0: # Ja tikai vienu elementu ievada
```

```
    print("Šajā virknē ir tikai viens elements.\nAritmētiskas progresijas tiek definētas skaitļu virknem ar vismaz diviem elementiem.")
    quit()

d = x2 - x1

N = 3

x = float(input( "Ievadiet " + str(N) + " locekli ===> "))

while x != 0:
    d1 = x - x2

    if d != d1:
        sv = "Dota virkne nav aritmētiska progresija"

    x2 = x
    N = N+1
    x =  float(input("Ievadiet " + str(N) + " locekli ===> "))
else:
    print(sv)

if x==0: # pedējo un prikšpēdējo atņēmt
    d1 = x - x2
    if d != d1:
        sv = "Dota virkne nav aritmētiska progresija"
else:
    print(sv)
```

Testa piemēri:

1)

```
Ievadiet 1 locekli ===> 1
Ievadiet 2 locekli ===> 1
Ievadiet 3 locekli ===> 1
Ievadiet 4 locekli ===> 1
Ievadiet 5 locekli ===> 1
Ievadiet 6 locekli ===> 1
Ievadiet 7 locekli ===> 0
Dota virkne ir aritmētiska progresija
```

2)

```
Ievadiet 1 locekli ===> 1
Ievadiet 2 locekli ===> 2
Ievadiet 3 locekli ===> 3
Ievadiet 4 locekli ===> 4
Ievadiet 5 locekli ===> 5
Ievadiet 6 locekli ===> 6
Ievadiet 7 locekli ===> 0
Dota virkne ir aritmētiska progresija
```

3)

```
Ievadiet 1 locekli ===> 4
Ievadiet 2 locekli ===> 6
Ievadiet 3 locekli ===> 8
Ievadiet 4 locekli ===> 10
Ievadiet 5 locekli ===> 12
Ievadiet 6 locekli ===> 14
Ievadiet 7 locekli ===> 0
Dota virkne ir aritmētiska progresija
```

4)

```
Ievadiet 1 locekli ===> 5
Ievadiet 2 locekli ===> 7
Ievadiet 3 locekli ===> 9
Ievadiet 4 locekli ===> 11
Ievadiet 5 locekli ===> 15
Ievadiet 6 locekli ===> 12
Ievadiet 7 locekli ===> 10
Ievadiet 8 locekli ===> 0
Dota virkne nav aritmētiska progresija
```

5)

```
Ievadiet 1 locekli ===> 10
Ievadiet 2 locekli ===> 20
Ievadiet 3 locekli ===> 30
Ievadiet 4 locekli ===> 40
Ievadiet 5 locekli ===> 50
Ievadiet 6 locekli ===> 0
Dota virkne ir aritmētiska progresija
```

6)

```
Ievadiet 1 locekli ===> 1
Ievadiet 2 locekli ===> 2
Ievadiet 3 locekli ===> 3
Ievadiet 4 locekli ===> 4
Ievadiet 5 locekli ===> 5
Ievadiet 6 locekli ===> 16541641
Ievadiet 7 locekli ===> 0
Dota virkne nav aritmētiska progresija
```

7)

```
Ievadiet 1 locekli ===> 1
Ievadiet 2 locekli ===> 2
Ievadiet 3 locekli ===> 3
Ievadiet 4 locekli ===> 641469
Ievadiet 5 locekli ===> 0
Dota virkne nav aritmētiska progresija
```

5. papilduzdevums (PU2)

Sastādīt programmu, kas noskaidro, vai lietotāja secīgi ievadītā skaitļu virkne veido gan aritmētisko progresiju, gan ģeometrisko progresiju, vai tikai aritmētisko progresiju, vai tikai ģeometrisko progresiju vai arī virkne neveido ne aritmētisko progresiju, ne ģeometrisko progresiju. Skaitļu virknes ievadi pārtrauc ievadot skaitli 0.

Kods:

```
# Programmas nosaukums: 5. uzd MPR11 PU2

# 5. uzdevums MPR11 PU2

# Uzdevuma formulējums: Sastādīt programmu, kas noskaidro, vai lietotāja secīgi ievadītā skaitļu virkne veido gan aritmētisko progresiju, gan ģeometrisko progresiju, vai tikai aritmētisko progresiju, vai tikai ģeometrisko progresiju vai arī virkne neveido ne aritmētisko progresiju, ne ģeometrisko progresiju. Skaitļu virknes ievadi pārtrauc ievadot skaitli 0.

# Versija 1.0


sv = "Dota virkne ir aritmētiskā progresija"

sv1 = "Dota virkne ir geometriskā progresija"


x1 = float(input("Ievadiet 1 locekli ===> "))


if x1 == 0:

    print("Šajā virknē nav elementu.") # Ja uzreiz 0 ievada

    quit()


x2 = float(input("Ievadiet 2 locekli ===> "))


if x2 == 0: # Ja tikai vienu elementu ievada

    print("Šajā virknē ir tikai viens elements.\nAritmētiskas un ģeometriskas progresijas tiek definētas skaitļu virknem ar vismaz diviem elementiem.")

    quit()


d = x2 - x1

q = x2/x1

N = 3
```

```
x = float(input( "Ievadiet " + str(N) + " locekli ===> "))



while x != 0:
    d1 = x - x2
    q1 = x/x2

    if d != d1:
        sv = "Dota virkne nav aritmētiskā progresija"

    if q != q1:
        sv1 = "Dota virkne nav geometriskā progresija"

    x2 = x
    N = N+1
    x =  float(input("Ievadiet " + str(N) + " locekli ===> "))
else:
    print(sv)
    print(sv1)

if x==0: # pedējo un prikšpēdējo atņēmt
    d1 = x - x2
    q1 = x/x2
    if d != d1:
        sv = "Dota virkne nav aritmētiskā progresija"
    if q != q1:
        sv1 = "Dota virkne nav geometriskā progresija"
else:
    print(sv)
    print(sv1)
```

Testa piemēri:

1)

```
Ievadiet 1 locekli ===> 625
Ievadiet 2 locekli ===> 125
Ievadiet 3 locekli ===> 25
Ievadiet 4 locekli ===> 5
Ievadiet 5 locekli ===> 1
Ievadiet 6 locekli ===> 0.2
Ievadiet 7 locekli ===> 0
Dota virkne nav aritmētiskā progresija
Dota virkne ir geometriskā progresija
```

2)

```
Ievadiet 1 locekli ===> 1
Ievadiet 2 locekli ===> 2
Ievadiet 3 locekli ===> 3
Ievadiet 4 locekli ===> 0
Dota virkne ir aritmētiskā progresija
Dota virkne nav geometriskā progresija
```

3)

```
Ievadiet 1 locekli ===> 0
Šajā virknē nav elementu.
```

4)

```
Ievadiet 1 locekli ===> 1
Ievadiet 2 locekli ===> 0
Šajā virknē ir tikai viens elements.
Aritmētiskas un ģeometriskas progresijas tiek definētas skaitļu virknem ar vismaz diviem elementiem.
```

5)

```
Ievadiet 1 locekli ===> 1
Ievadiet 2 locekli ===> 2
Ievadiet 3 locekli ===> 3
Ievadiet 4 locekli ===> 0
Dota virkne ir aritmētiskā progresija
Dota virkne nav geometriskā progresija
```

6)

```
Ievadiet 1 locekli ===> 1
Ievadiet 2 locekli ===> 2
Ievadiet 3 locekli ===> 3
Ievadiet 4 locekli ===> 4
Ievadiet 5 locekli ===> 5
Ievadiet 6 locekli ===> 6
Ievadiet 7 locekli ===> 7
Ievadiet 8 locekli ===> 0
Dota virkne ir aritmētiskā progresija
Dota virkne nav geometriskā progresija
```

7)

```
Ievadiet 1 locekli ===> 2
Ievadiet 2 locekli ===> 4
Ievadiet 3 locekli ===> 8
Ievadiet 4 locekli ===> 16
Ievadiet 5 locekli ===> 32
Ievadiet 6 locekli ===> 0
Dota virkne nav aritmētiskā progresija
Dota virkne ir geometriskā progresija
```

8)

```
Ievadiet 1 locekli ===> 2
Ievadiet 2 locekli ===> 4
Ievadiet 3 locekli ===> 8
Ievadiet 4 locekli ===> 16
Ievadiet 5 locekli ===> 32
Ievadiet 6 locekli ===> 49648964
Ievadiet 7 locekli ===> 0
Dota virkne nav aritmētiskā progresija
Dota virkne nav geometriskā progresija
```

9)

```
Ievadiet 1 locekli ===> 2
Ievadiet 2 locekli ===> 4
Ievadiet 3 locekli ===> 8
Ievadiet 4 locekli ===> 16
Ievadiet 5 locekli ===> 65415641641641
Ievadiet 6 locekli ===> 0
Dota virkne nav aritmētiskā progresija
Dota virkne nav geometriskā progresija
```

10)

```
Ievadiet 1 locekli ===> 2
Ievadiet 2 locekli ===> 4
Ievadiet 3 locekli ===> 6
Ievadiet 4 locekli ===> 8
Ievadiet 5 locekli ===> 10
Ievadiet 6 locekli ===> 1464164
Ievadiet 7 locekli ===> 0
Dota virkne nav aritmētiskā progresija
Dota virkne nav geometriskā progresija
```

11)

```
Ievadiet 1 locekli ===> 3
Ievadiet 2 locekli ===> 9
Ievadiet 3 locekli ===> 27
Ievadiet 4 locekli ===> 0
Dota virkne nav aritmētiskā progresija
Dota virkne ir geometriskā progresija
```

12)

```
Ievadiet 1 locekli ===> 3
Ievadiet 2 locekli ===> 9
Ievadiet 3 locekli ===> 15
Ievadiet 4 locekli ===> 21
Ievadiet 5 locekli ===> 0
Dota virkne ir aritmētiskā progresija
Dota virkne nav geometriskā progresija
```