# 1. uzdevums

Uzzīmēt funkcijas y=ax^3 + bx^2 + cx + d grafiku.

## Kods:

```
# Programmas nosaukums: 1. uzd MPR9
# 1. uzdevums MPR9
# Uzdevuma formulējums: uzzīmēt funkcijas y=ax^3 + bx^2 + cx + d grafiku.
# Versija 1.0

import tkinter
from tkinter import ttk

def paradit(): # funkcija zīmēšanai kanvā

    kanva.configure(bg='AliceBlue')

    kanva.create_line(150,350,850,350, arrow="last", fill="gray")
    kanva.create_text(847, 330, text="X", anchor = "nw")

    kanva.create_line(500, 0, 500, 700, arrow="first", fill="gray")
    kanva.create_text(507, 0, text= "Y", anchor="nw")

    # Y ass

    for i in range (175, 826, 25): #Y ASS
        kanva.create_line(i, 347, i, 353, fill="gray")

    for i in range(-13,0): # minusiem uz Y
        kanva.create_text(505, 341 - i*25, text = str(i), anchor = "nw", fill= "gray")

    for i in range(1,14): # minusiem uz X
```

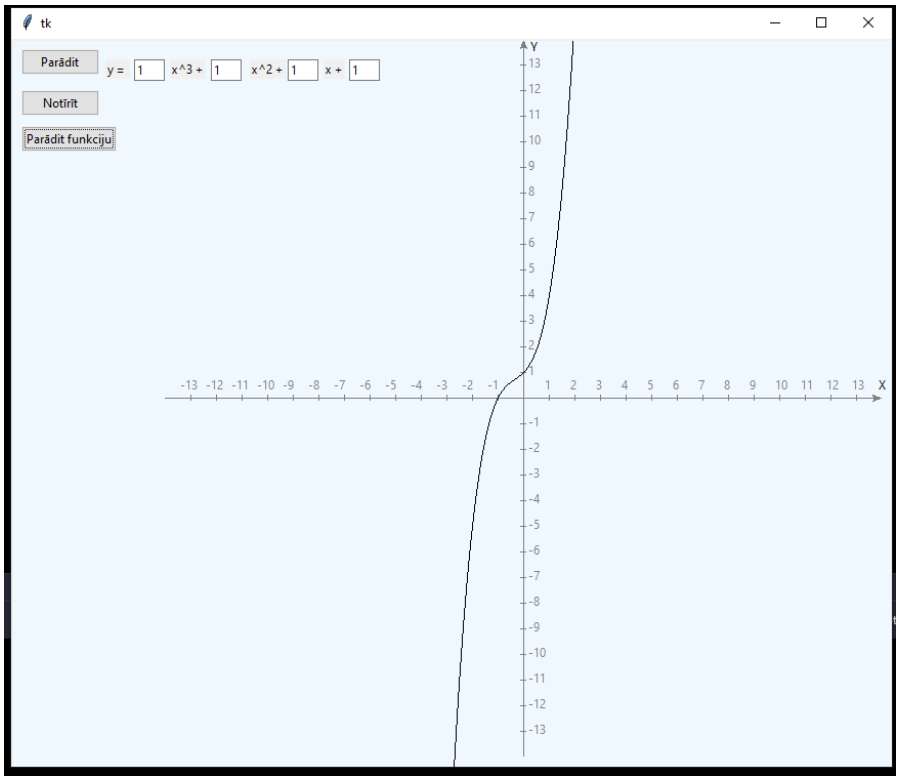
```
        kanva.create_text(505, 341 - i*25, text = str(i), anchor = "nw", fill= "gray")

    # X ass

    for i in range (25, 676,25) : # X ASS
        kanva.create_line(497, i, 503, i, fill="gray")

    for i in range(-13,0): # minusiem uz X
        kanva.create_text(490 + i*25, 330, text = str(i), anchor = "nw", fill= "gray")

    for i in range(1, 14): # plusiem uz X
        kanva.create_text(496 + i*25, 330, text = str(i), anchor = "nw", fill= "gray")



def funkcija(): # funkcijas grafika zīmēšanai kanvā

        a = float(e1.get())
        b = float(e2.get())
        c = float(e3.get())
        d = float(e4.get())

        x = (175 - 500)/25
        y = a*x*x*x + b*x*x + c*x + d # Funkcija
        x1 = 175
        y1 = 350-y*25

        for i in range (175, 825):
            x = (i - 500)/25
            y = a*x*x*x + b*x*x + c*x + d # Funkcija
            x2 = i
            y2 = 350-y*25
```

```
        kanva.create_line(x1, y1, x2, y2)
        x1=x2
        y1=y2

# funkcija satura dzēšanai

def notirit():
    kanva.configure(bg='white')
    kanva.delete("all")

# Loga atribūti

logs = tkinter.Tk()
logs.geometry("860x710")

# Kanvas novietošana

kanva = tkinter.Canvas(logs, bg="white", height=710, width=860)
kanva.place(x=0, y=0)

# Pogas

poga1= ttk.Button(logs, text="Parādit", command=paradit)
poga1.place(x=10, y=10)

poga2 = ttk.Button(logs, text="Notīrīt", command=notirit)
poga2.place(x=10, y=50)

poga3= ttk.Button(logs, text="Parādit funkciju", command=funkcija)
poga3.place(x=10, y=85)
```

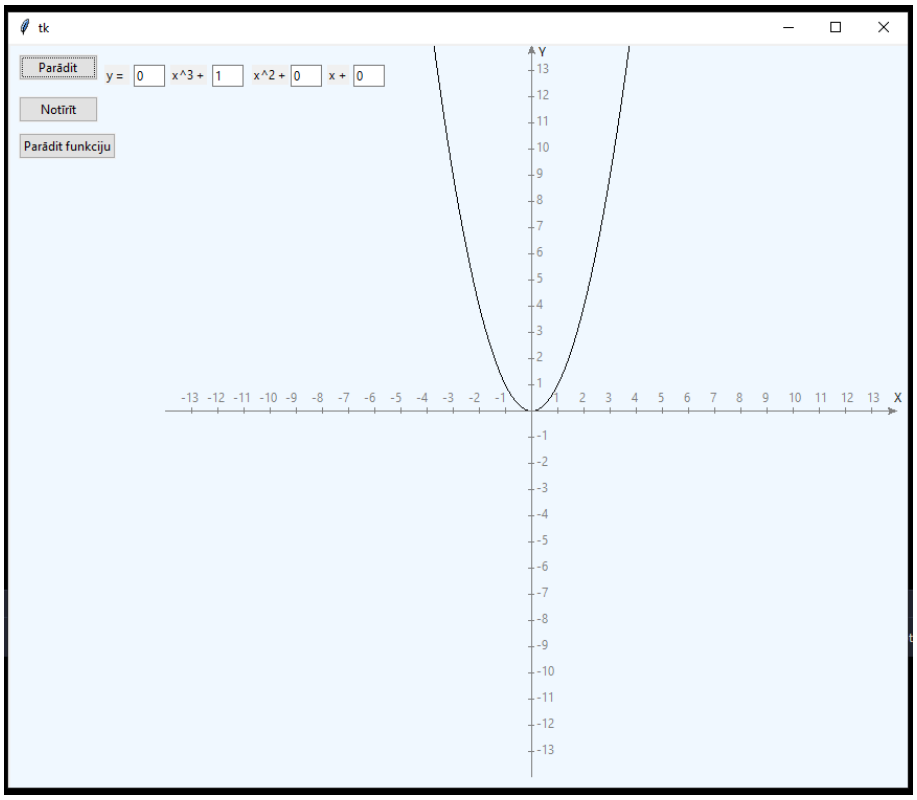
```
# Ievadisanas laukumi

e1 = ttk.Entry(logs)
e1.place(x=120, y=20, width=30)

e2 = ttk.Entry(logs)
e2.place(x=195, y=20, width=30)

e3 = ttk.Entry(logs)
e3.place(x=270, y=20, width=30)

e4 = ttk.Entry(logs)
e4.place(x=330, y=20, width=30)



# Etiketes
l0 = ttk.Label(logs, text="y = ")
l0.place(x=92, y=20)

l1 = ttk.Label(logs, text="x^3 +")
l1.place(x=155, y=20)

l2 = ttk.Label(logs, text="x^2 +")
l2.place(x=233, y=20)

l3 = ttk.Label(logs, text="x +")
l3.place(x=305, y=20)

# Obligāta rindiņa, lai logs būtu redzāms visu laiku
logs.mainloop()
```

# Testa piemēri:

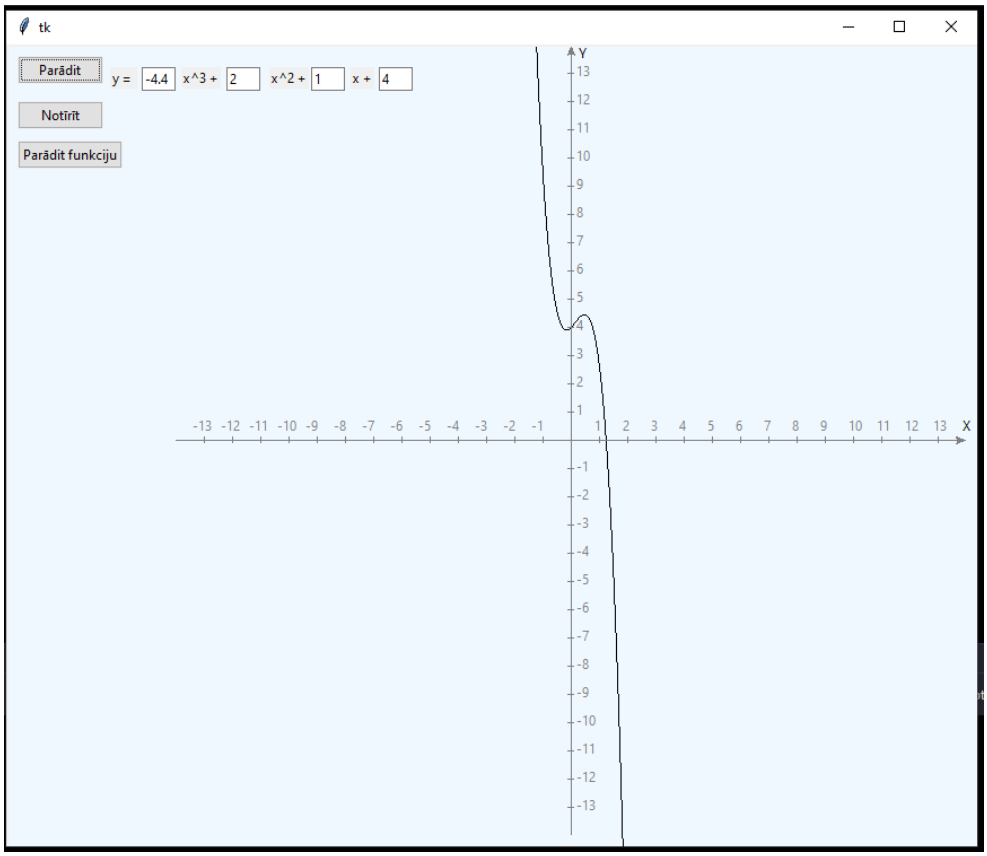
1)



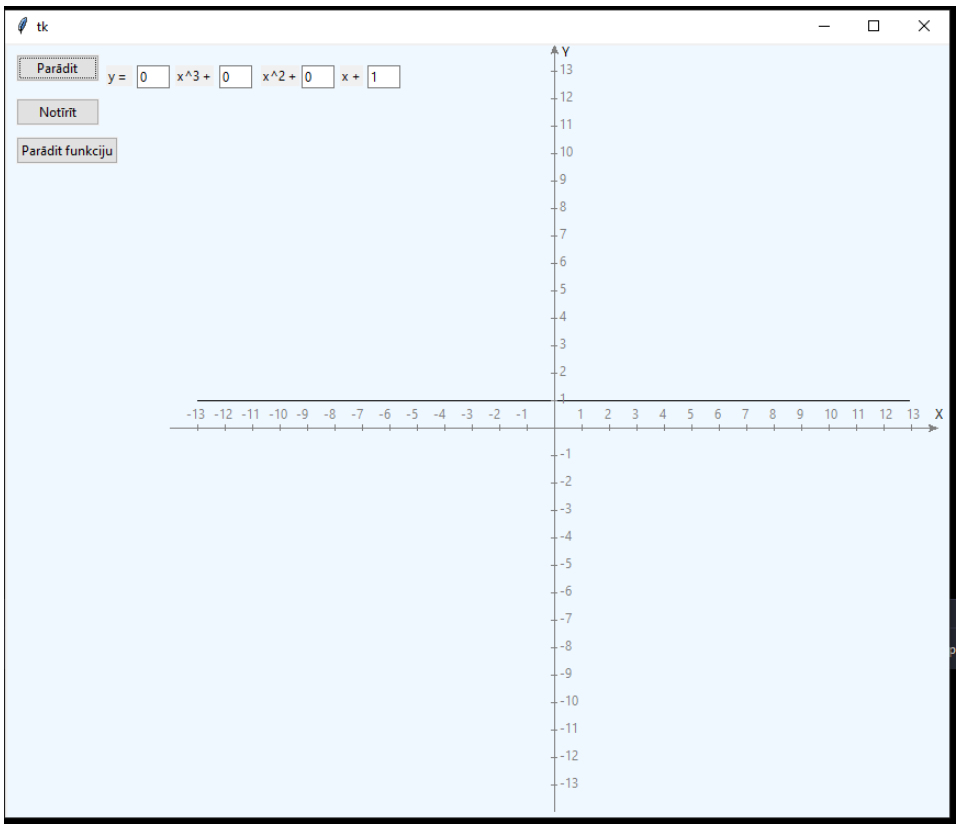

2)



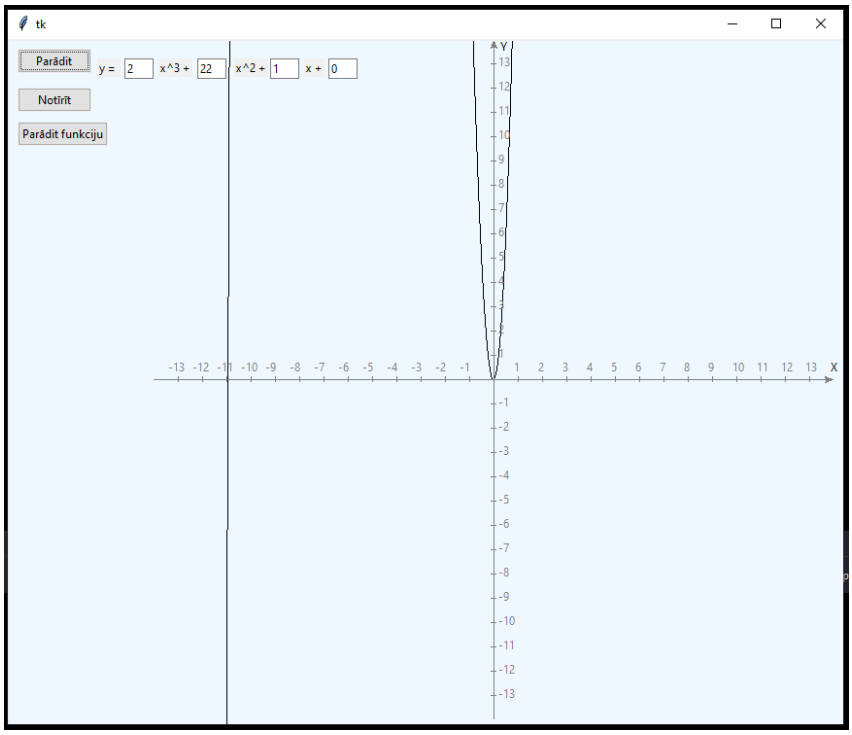
3)


tk
Parādit
y = -4.4 x^3 + 2 x^2 + 1 x + 4
Notīrīt
Parādit funkciju
Y
X

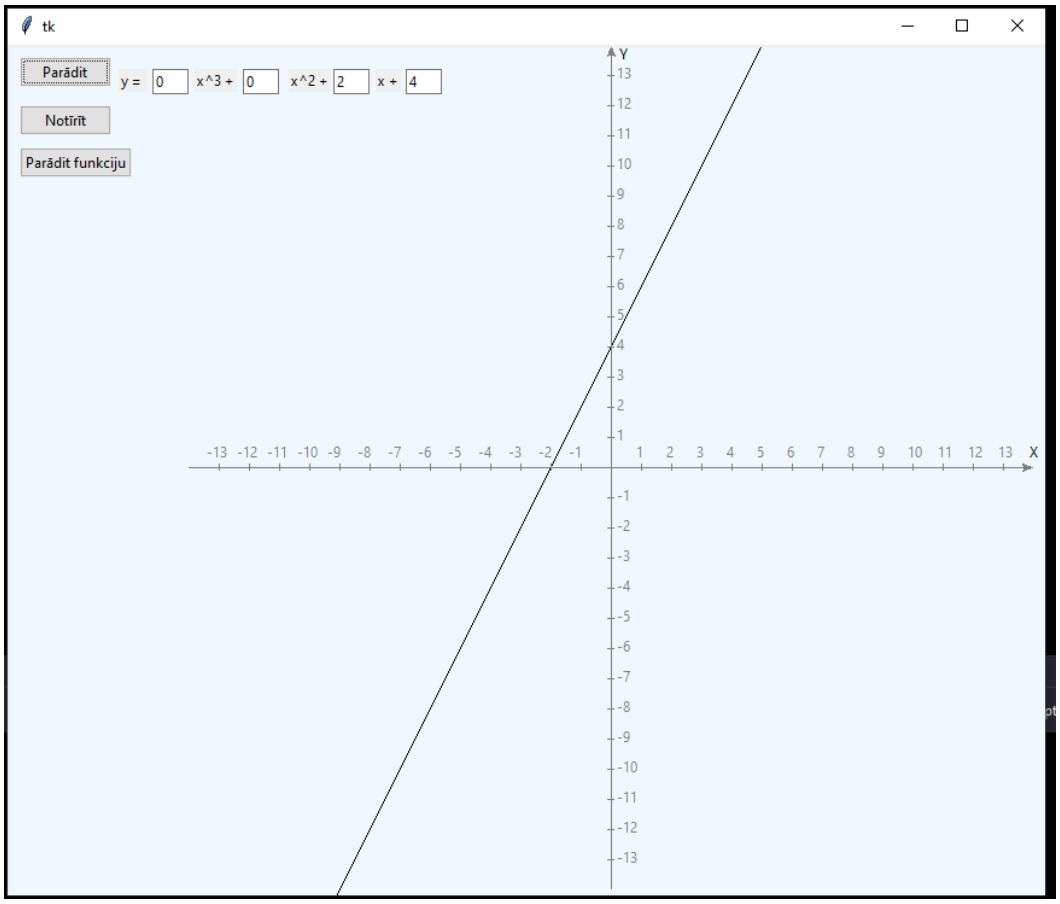
4)


tk
Parādit
y = 0 x^3 + 0 x^2 + 0 x + 1
Notīrīt
Parādit funkciju
Y
X

5)


tk
Parādit
y = 2 x^3 + 22 x^2 + 1 x + 0
Notīrīt
Parādit funkciju
Y
X

6)


tk
Parādit
y = 0 x^3 + 0 x^2 + 2 x + 4
Notīrīt
Parādit funkciju
Y
X

# 2. uzdevums

Uzzīmēt attēlā redzamo līniju, ja zināms, ka tā uzdota polārajās koordinātās r =a*cos(6*alpha). Parametra a vērtību ievada lietotājs, bet leņķis a pieder [0;2Pi]

## Kods:

```
# Programmas nosaukums: 2. uzd MPR9

# 2. uzdevums MPR9

# Uzdevuma formulējums: Uzzimēt attēlā redzamo līniju, ja zināms, ka tā uzdota polārajās koordinātās r = a*cos(6*alpha). Parametra a vērtibu ievada lietotājs, bet leņķis a pieder [0;2Pi]

# Versija 1.0


# Import

import math

import tkinter

from tkinter import ttk


def paradit(): # funkcija zīmēšanai kanvā


    kanva.create_line(150,350,850,350, arrow="last", fill="gray")

    kanva.create_text(847, 330, text="X", anchor = "nw")


    kanva.create_line(500, 0, 500, 700, arrow="first", fill="gray")

    kanva.create_text(507, 0, text= "Y", anchor="nw")



    # Y ass

    for i in range (175, 826, 25): #Y ASS

        kanva.create_line(i, 347, i, 353, fill="gray")


    for i in range(-13,0): # minusiem uz Y

        kanva.create_text(505, 341 - i*25, text = str(i), anchor = "nw", fill= "gray")
```

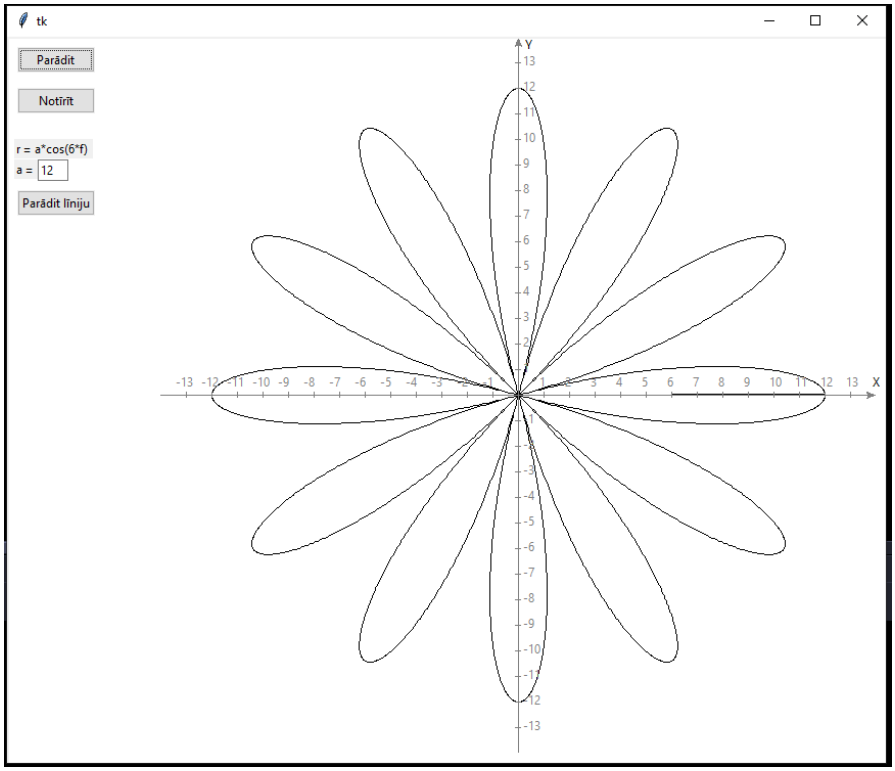
```
    for i in range(1,14): # minusiem uz X
        kanva.create_text(505, 341 - i*25, text = str(i), anchor = "nw", fill= "gray")



    # X ass
    for i in range (25, 676,25) : # X ASS
        kanva.create_line(497, i, 503, i, fill="gray")

    for i in range(-13,0): # minusiem uz X
        kanva.create_text(490 + i*25, 330, text = str(i), anchor = "nw", fill= "gray")

    for i in range(1, 14): # plusiem uz X
        kanva.create_text(496 + i*25, 330, text = str(i), anchor = "nw", fill= "gray")



def linija():

    x0 = 500
    y0 = 350
    a = float(e1.get())
    b = 2*x0+a/25
    x1 = x0 + a/25
    y1 = y0

    # Līnija
    for i in range (1, 3600, 1):

        f = i/1800*math.pi
        x = a*math.cos(6*f)*math.cos(f)

        y = a*math.cos(6*f)*math.sin(f)
```

```
        y2 = -y*25 + y0
        x2 = x*25 + x0

        kanva.create_line(x1,y1,x2,y2)
        x1=x2
        y1=y2


def notirit(): # funkcija kanvas satura dzēšanai
    kanva.delete("all")

# Logs
logs = tkinter.Tk()
logs.geometry("860x710")

# Kanva
kanva = tkinter.Canvas(logs, bg="white", height=710, width=860)
kanva.place(x=0, y=0)

# Pogu definēšana
poga1= ttk.Button(logs, text="Parādit", command=paradit)
poga1.place(x=10, y=10)

poga2 = ttk.Button(logs, text="Notīrīt", command=notirit)
poga2.place(x=10, y=50)

poga3= ttk.Button(logs, text="Parādit līniju", command=linija)
poga3.place(x=10, y=150)
```

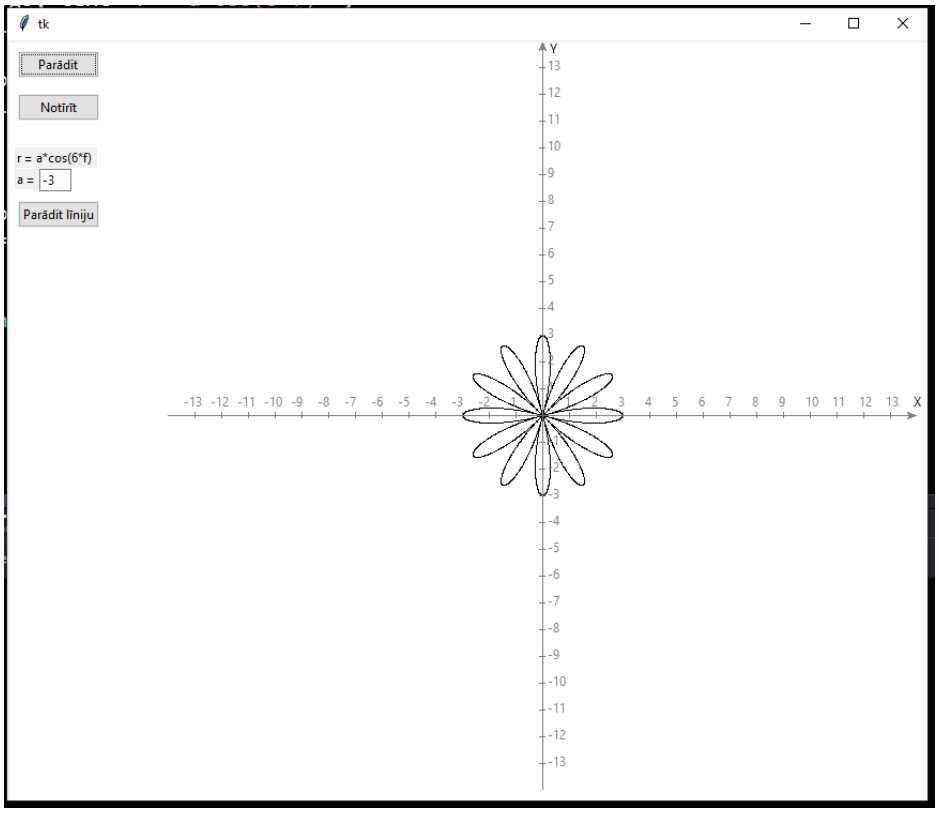
```
# Etiketes
l0 = ttk.Label(logs, text="r = a*cos(6*f) ")
l0.place(x=7, y=100)

l1 = ttk.Label(logs, text="a = ")
l1.place(x=7, y=120)

# Entry
e1 = ttk.Entry(logs)
e1.place(x=30, y=120, width=30)

# Obligāta rindiņa
logs.mainloop()
```

# Testa piemēri:

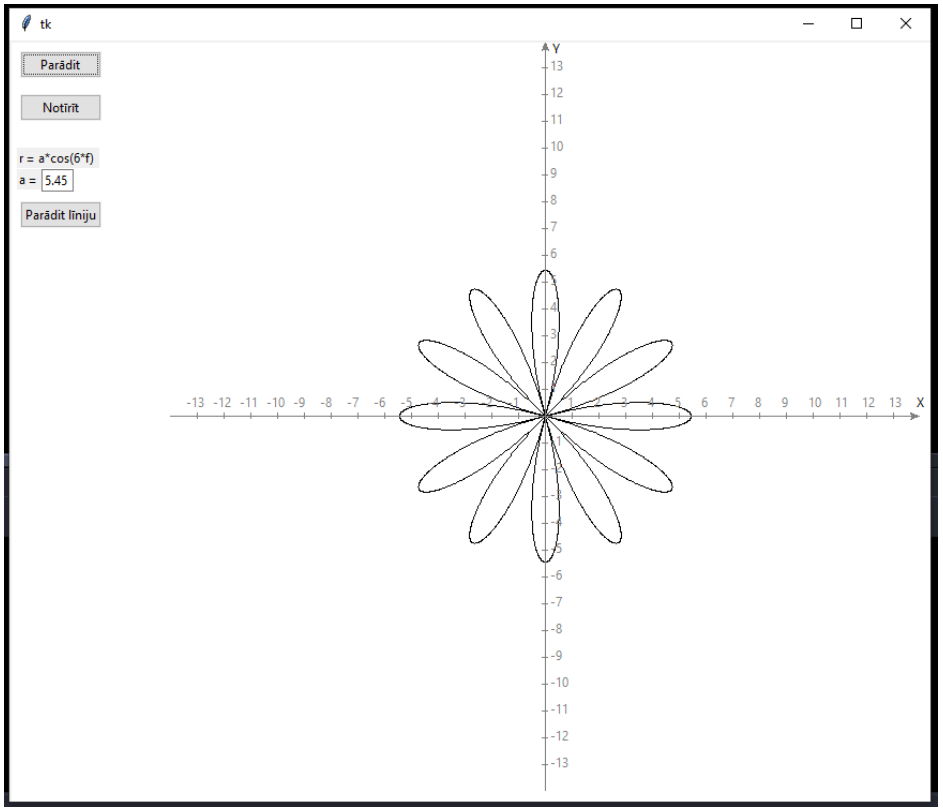
1)



2)


tk
Parādit
Notīrīt
r = a*cos(6*f)
a = -3
Parādit līniju
Y
X

3)


tk
Parādit
Notīrīt
r = a*cos(6*f)
a = 5.45
Parādit līniju
Y
X

# 3. uzdevums

Sastādīt programmu, kas noskaidro, vai lietotāja ievadītais skaitlis ir pirmskaitlis.

## Kods:

```
# Programmas nosaukums: 3. uzd MPR9

# 3. uzdevums MPR9

# Uzdevuma formulējums: Sastādīt programmu, kas noskaidro, vai lietotāja ievadītais skaitlis ir pirmskaitlis.

# Versija 1.0


import sys


a=int(input("Ievadiet naturālo skaitļi, kurš ir lielāks par 1.\n==> "))


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

if a <= 0:

   print("Skaitlis "+ str(a) + " nav naturāls skaitlis.")

   sys.exit(0)


if a==1:

   print("Skaitlis 1 nav lielāks par 1.")

   sys.exit(0)

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


b=1

c=a


for a in range (1, a-1, 1):

   b = b+1

   if c%b == 0:
```

```
        print("Skaitlis " + str(c) + " nav pirmskaitlis")
        sys.exit(0)


print("Skaitlis " + str(c) + " ir pirmskaitlis")
```

## Testa piemēri:

1)

```
Ievadiet naturālo skaitļi, kurš ir lielāks par 1.
==> 20
Skaitlis 20 nav pirmskaitlis
```

2)

```
Ievadiet naturālo skaitļi, kurš ir lielāks par 1.
==> 1
Skaitlis 1 nav lielāks par 1.
```

3)

```
Ievadiet naturālo skaitļi, kurš ir lielāks par 1.
==> -5
Skaitlis -5 nav naturāls skaitlis.
```

4)

```
Ievadiet naturālo skaitļi, kurš ir lielāks par 1.
==> 9929
Skaitlis 9929 ir pirmskaitlis
```

5)

```
Ievadiet naturālo skaitļi, kurš ir lielāks par 1.
==> 2
Skaitlis 2 ir pirmskaitlis
```

6)

```
Ievadiet naturālo skaitļi, kurš ir lielāks par 1.
==> 661
Skaitlis 661 ir pirmskaitlis
```

7)

```
Ievadiet naturālo skaitļi, kurš ir lielāks par 1.
==> 14714292174017924102491289449284488484444444442
Skaitlis 14714292174017924102491289449284488484444444442 nav pirmskaitlis
```

# 4. uzdevums

Aprēķināt C n m divos veidos, t.i., izmantojot faktoriālu aprēķināšanu un veicot pakāpenisku reizināšanu un dalīšanu.

## Kods:

```
# Programmas nosaukums: 4. uzd MPR9

# 4. uzdevums MPR9

# Uzdevuma formulējums: Aprēķināt C n m divos veidos, t.i., izmantojot faktoriālu aprēķināšanu un veicot pakāpenisku reizināšanu un dalīšanu

# Versija 1.0


import sys


n = int(input("Ievadi naturālu skaitli N (no cik) ===> "))

m = int(input("Ievadi naturālu skaitli M (pa cik) ===> "))


if m > n:

    print("M nevar būt lielākam par N")

    sys.exit(0)


fn = 1


for i in range(2, n+1) :
```

```
    fn = fn * i

fm = 1

for i in range(2, m+1) :
    fm = fm * i

fnm = 1
for i in range(2, n-m+1) :
    fnm = fnm * i

c=fn/fm/fnm



print ("\nC(" + str(m) + "," + str(n) +") = " + str(int(c)) + "\n")

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

nn = int(input("Ievadi no cik (N) ===> "))
mm = int(input("Ievadi pa cik (M) ===> "))

if mm > nn:
    print("M nevar būt lielākam par N")
    sys.exit(0)

n = nn
m = mm

if m > n/2 :
    m=n-m
n=n+1
```

```
c=1

for i in range(1, m+1) :
    c = c * (n-i) / i

print ("\nC(" + str(mm) + "," + str(nn) +") = " + str(int(c)))
```

## Testa piemēri:

1)

```
Ievadi naturālu skaitli N (no cik) ===> 10
Ievadi naturālu skaitli M (pa cik) ===> 10

C(10,10) = 1

Ievadi no cik (N) ===> 10
Ievadi pa cik (M) ===> 10

C(10,10) = 1
```

2)

```
Ievadi naturālu skaitli N (no cik) ===> 10
Ievadi naturālu skaitli M (pa cik) ===> 5

C(5,10) = 252

Ievadi no cik (N) ===> 10
Ievadi pa cik (M) ===> 5

C(5,10) = 252
```

3)

```
Ievadi naturālu skaitli N (no cik) ===> 5
Ievadi naturālu skaitli M (pa cik) ===> 10
M nevar būt lielākam par N
```

4)

```
Ievadi naturālu skaitli N (no cik) ===> 12
Ievadi naturālu skaitli M (pa cik) ===> 6

C(6,12) = 924

Ievadi no cik (N) ===> 6
Ievadi pa cik (M) ===> 12
M nevar būt lielākam par N
```

# 5. uzdevums

Sastādīt programmu, kas nodrukā vārdu otrādi, piemēram, ievada - KARTUPELIS, bet izvada - SILEPUTRAK.

## Kods:

```
# Programmas nosaukums: 5. uzd MPR9

# 5. uzdevums MPR9

# Uzdevuma formulējums: Sastādīt programmu, kas nodrukā vārdu otrādi, piemēram, ievada - KARTUPELIS, bet izvada - SILEPUTRAK

# Versija 1.0


v=input("Ievadiet vārdu : ")


v2=""


for i in v:
  v2=i+v2


print(v2)
```

### Testa piemēri:

1)

```
Ievadiet vārdu : VADO
ODAV
```

2)

```
Ievadiet vārdu : SULA ALUS
SULA ALUS
```

3)

```
Ievadiet vārdu : VANNA TANNA MAMMA
AMMAM ANNAT ANNAV
```

4)

```
Ievadiet vārdu : KARTUPELIS
SILEPUTRAK
```

5)

```
Ievadiet vārdu : SILEPUTRAK
KARTUPELIS
```

# 6. uzdevums

Sastādīt programmu, kas noskaidro vai lietotāja ievadītā simbolu virkne ir palindroma, t.i., no abām pusēm lasās vienādi, piemēram, SMUKUMS, SULA ARI IRA ALUS

### Kods:

```
# Programmas nosaukums: 6. uzd MPR9

# 6. uzdevums MPR9

# Uzdevuma formulējums: Sastādīt programmu, kas noskaidro vai lietotāja ievadītā simbolu virkne ir palindroma, t.i., no abām pusēm lasās vienādi, piemēram, SMUKUMS, SULA ARI IRA ALUS

# Versija 1.0
```

```
x = input("Ievadi vārdu: ")
y = len(x) # Cik ir garums

atb = "Ir palindroms"

for i in range (y//2):
    if x[i] != x[y-1-i]:
        atb ="Nav palindroms"
        break

print(atb)
```

## Testa piemēri:

1)

```
Ievadi vārdu: ANNA
Ir palindroms
```

2)

```
Ievadi vārdu: VANNA
Nav palindroms
```

3)

```
Ievadi vārdu: SMUKUMS
Ir palindroms
```

4)

```
Ievadi vārdu: SULA ARI IRA ALUS
Ir palindroms
```