**Profesionālās izglītības kompetences centra**

**“Rīgas Tehniskā koledža”**

**profesionālā vidusskola**

Mācību modulis - Matemātikas speciālās nodaļas PB6

**1. patstāvīgais darbs**

Atrisināt vienādojumuu sistēmu ar precizitāti līdz 0,0001, izmantojot Gausa metodi.

**28. variants**

Skolotāja: Vita Balikova\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| audzēkņa | *vārds, uzvārds* | Roberts Legzdiņš |
| *grupa* | P1-2 |
| *e-pasts, tālrunis* | [robzlegz@gmail.com](mailto:robzlegz@gmail.com) 26956611 |
| *jānodod* | 17.02.2023. |

|  |  |
| --- | --- |
| **Skolotāja piezīmes un darba atzīme** | |
| Atzīme: |  |
| Paraksts: |  |
| Nodošanas datums: | 19.02.2023. |
|  |  |

Rīga, 2023

**Saturs**

Uzdevuma formulējums 3

Teorētiskais pamatojums 3

Programmas kods 4,5

Programma darbībā 6, 7

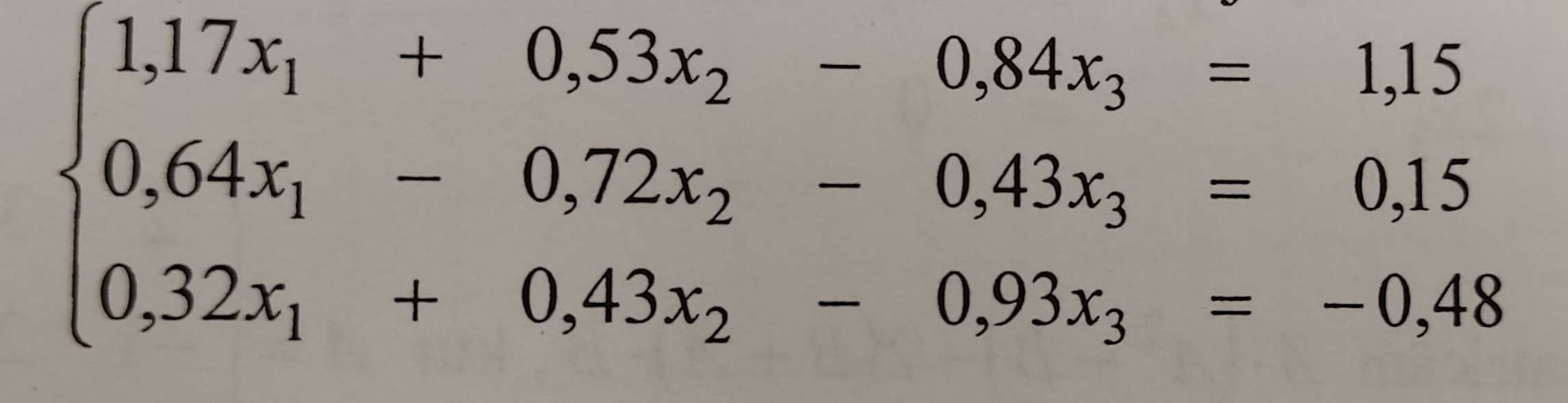
Programma atrisina doto vienādojuma sistēmu 8

Lietotāja ceļvedis 9

**Uzdevuma formulējums**

**28. variants**

Atrisināt vienādojumuu sistēmu ar precizitāti līdz 0,0001, izmantojot Gausa metodi.



### Programmu rakstīju Python programmēšanas valodā.

**Teorētiskais pamatojums**

Matemātikā **Gausa eliminācija** (saukta arī par **rindu reducēšanu)** ir metode, kuru izmanto lineāru vienādojumu sistēmu risināšanai. Tā ir nosaukta slavenā vācu matemātiķa Kārļa Frīdriha Gausa vārdā, kurš rakstīja par šo metodi, bet neizgudroja to.

Lai veiktu Gausa izslēgšanu, lineāro vienādojumu sistēmas locekļu koeficienti tiek izmantoti, lai izveidotu matricas veidu, ko sauc par papildinātu matricu. Tad matricas vienkāršošanai izmanto elementāras rindas operācijas. Izmanto šādus trīs rindu operāciju veidus:

: vienas rindas pārslēgšana ar citu rindu.

: rindas reizināšana ar skaitli, kas nav nulle.

: rindas pievienošana vai atņemšana no citas rindas.

Gausa eliminācijas mērķis ir iegūt matricu *rindu-ekelonu formā*. Ja matrica ir rindas-ekelona formā, tas nozīmē, ka, lasot no kreisās uz labo, katra rinda sākas ar vismaz vienu nulles locekli vairāk nekā rinda virs tās.

**Programmas kods**

*import* numpy *as* np

*import* math

class App:

    def \_\_init\_\_(self):

*self*.rows: int = *self*.get\_num("Enter row count: ", integer=True)

*self*.matrix: list[list[float]] = *self*.get\_matrix(*self*.rows)

*self*.result: list[float] = *self*.gauss(*self*.matrix)

        print("Result: ", *self*.result)

    def get\_num(self, txt: str = "Enter int: ", integer=False) -> int:

        num = None

*while* num == None:

*try*:

*if* integer == True:

                    num = int(input(txt))

*else*:

                    num = float(input(txt))

*except*:

                num = *self*.get\_num("Try again: ")

*return* num

    def get\_matrix(self, rows: int) -> list[list[int]]:

        matrix = []

*for* i *in* range(rows):

            row = [*self*.get\_num(f"Enter item ({i}:{j}): ") *for* j *in* range(rows+1)]

            matrix.append(row)

*return* matrix

    def floor\_num(self, x: float) -> float:

        limiter = 10000

        floored = math.floor(x \* limiter)

*return* floored / limiter

    def gauss(self, matrix: list[list[int]]) -> list[float]:

        print("Starting matrix:")

        print(np.matrix(matrix))

        divider = matrix[0][0]

        print(f"Dividing R0 by {divider}")

*for* i *in* range(len(matrix[0])):

            matrix[0][i] = matrix[0][i] / divider

        matrix = np.matrix(matrix)

        n = matrix.shape[0]

        print("Result matrix:")

        print(matrix)

*for* i *in* range(n):

            pivot = matrix[i, i]

*if* pivot == 0:

*raise* ValueError("Can't perform Gaussian elimination: 0 pivot encountered")

*for* j *in* range(i + 1, n):

                factor = matrix[j, i] / pivot

                print(f"{factor}R{i} + R{j}")

                matrix[j, i:] -= factor \* matrix[i, i:]

                print("After elimination:")

                print(matrix)

        x = np.zeros(n)

*for* i *in* range(n - 1, -1, -1):

            s = sum(matrix[i, j] \* x[j] *for* j *in* range(i + 1, n))

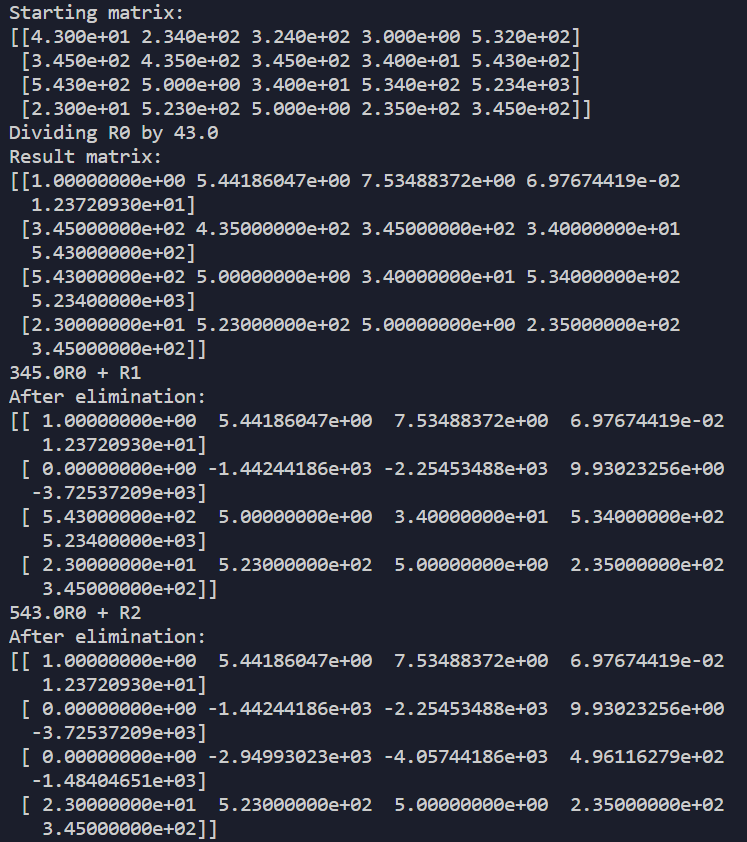
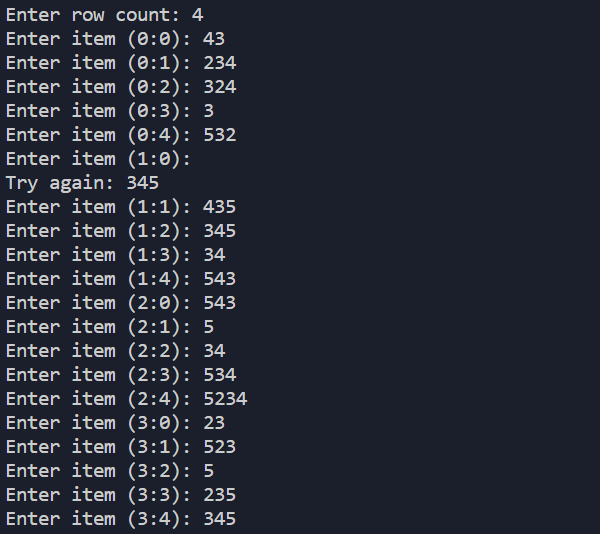
            result = (matrix[i, -1] - s) / matrix[i, i]

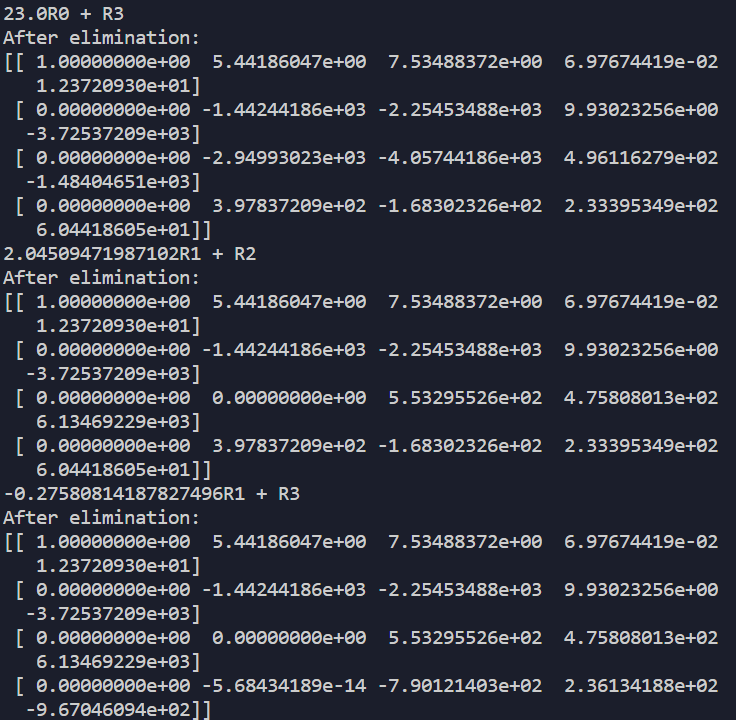
            x[i] = *self*.floor\_num(result)

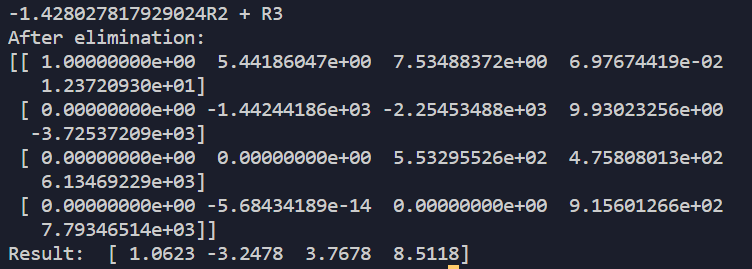
*return* x

\_ = App()

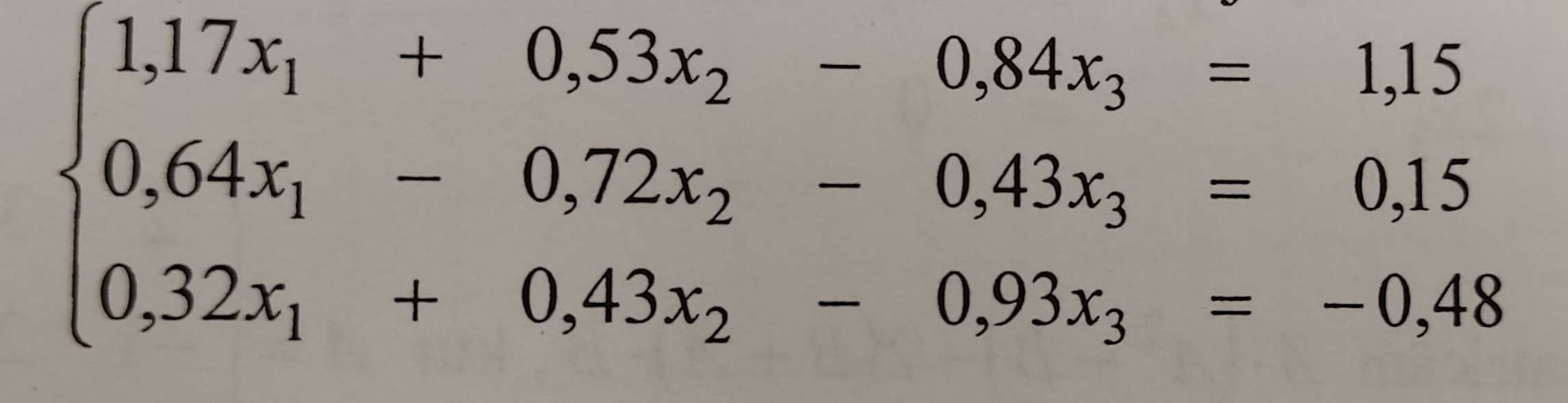
**Programma darbībā:**

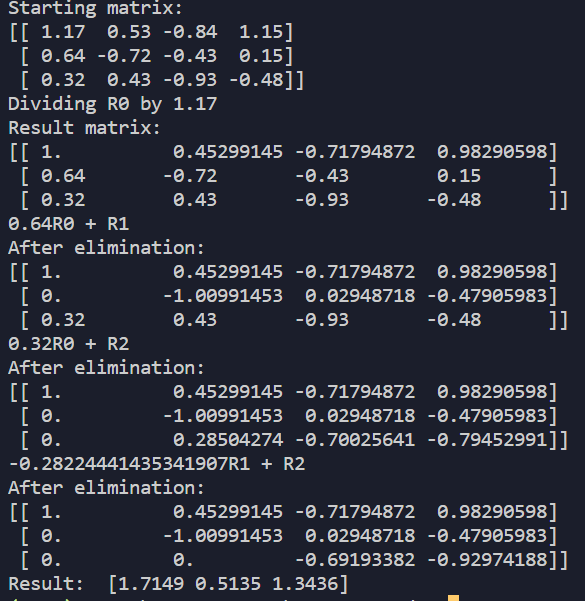






**Programma atrisina doto vienādojumu:**





**Lietotāja ceļvedis**

Programma ir paredzēta jebkāda izmēra vienādojumu sistēmu atrisināšanai un tā ir rakstīta Python programmēšanas valodā, izmanotjot numpy un math bibleotēkas. Kodu pārkopēt, instalēt numpy bibleotēku un palaist.

No sākuma lietotājam jāievada, cik rindas ir vienādojuma sistēmā, tad, pēc kārtas ievadīt visus nezināmos un vienādojuma rezultātu, kad tas izdarīts tiek izprintēta ievadītā matrica, risināšanas soļi un rezultāts.