Tallinn University of Technology

School of Engineering

Integrated Engineering

Emma Hansson

243487MVEB

**ARRAYS**

Second homework in course IAX0583

Supervisor: Vladimir Viies

Tallinn 2024

# AUTHORS DECLARATION

I confirm that I have prepared this homework independently and that it has not been previously submitted for defense by someone else. All the works of other authors used in the preparation of the work, important points of view, data from literary sources and elsewhere are cited in the work.

Author: Emma Hansson

Date: 10.12.2024

**CONTENTS**

[AUTHORS DECLARATION 2](#_Toc184746273)

[1. TASK STATEMENT 4](#_Toc184746274)

[1.1. Specific task 4](#_Toc184746275)

[1.1.1. Method 4](#_Toc184746281)

[2. SOLUTION DESCRIPTION 5](#_Toc184746282)

[2.1. Workflow 5](#_Toc184746283)

[2.2. The algorithm 7](#_Toc184746284)

[2.3. The code in C 9](#_Toc184746285)

[2.4. Special cases 11](#_Toc184746286)

[3. CONCLUSION 12](#_Toc184746287)

[EXTRA MATERIALS 13](#_Toc184746288)

[REFERENCES 14](#_Toc184746289)

# TASK STATEMENT

The task is to construct the algorithm of the task and the corresponding program in C language. All raw data are real numbers and are entered from the keyboard. The results are displayed on the screen

## Specific task

# Specific task is option 9:

# ,,Write an algorithm and its jointly corresponding program by which:

# 1. the integer number of rows and columns n (1 < n < 10) of the two-dimensional array (square matrix) A and the real number elements row by row are entered;

# 2. a real number array B is formed by transposing (NB! see what it means) matrix A;

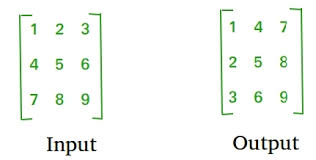
# 3. the array B is displayed (displayed on the screen) row by row.”

### 1.1.1. Method

The initial and final values n and matrix A entries ​​are given. Valid condition: 1 < n < 10.

The new matrix B is found by transposing given matrix A. The transpose of a matrix is obtained by changing the rows into columns and columns into rows for a given matrix A.

Example:



# SOLUTION DESCRIPTION

The goal of this programme is to transpose the given square matrix A, with dimensions [n][n] and output it as matrix B. Dimension n and matrix entriees will be determined by the user, where the user inserts n and entries to matrix one by one. All inserted values are integers for simple representation.

The user enters:

* n (where 1<n<10);
* entries for matrix A[n][n],

which all are real numbers.

All values ​​are entered separately and in case of incorrect form or parameters not being met, the user is asked to re-enter the values.

## Workflow

Simple description of the workflow of programme:

1. Defines the original data
2. Introduces the programme to the user
3. Asks for inputs
4. Check’s inputs and asks again if needed
5. Calculates transposed matrix
6. Outputs the results

In code, the **prototypes** declare functions. The **main** initializes arrays and variables needed for the program's calculations (e.g., **n, matrix[n][n]**).The **main** also calls other functions to perform specific tasks:

* + **PrintIntroduction**: Displays an introductory message.
  + **InputForDimensions/InputForEntries**: Prompts the user for input values.
  + **TransposeMatrix:** Transposes matrix A.
  + **OutputResults:** Displays the results to the user.

## 2.2. The algorithm

Based on the workflow, I conducted an algorithm, using “yEd” graph editor. Alrogithm has 3 separate parts: input, process and output, which are shown visually by columns.

**A screenshot of a diagram

Description automatically generated**

Breakdown of the algorithm:

1. **Ask user input for n:** The program starts by asking the user to input the size of the matrix. The value of n must be between 1 and 10.
2. **Check if n is valid:** The program checks if the input value of n is within the valid range of 1 to 10.
3. **Ask user input for matrix A[n][n]:** If n is valid, the program prompts the user to input the elements of the matrix A.
4. **Check if inserted numbers are real numbers:** The program verifies that the elements entered by the user are real numbers.
5. **Transpose the matrix:** If the input is valid, the program proceeds to transpose the matrix A. This involves creating a new matrix B of the same size as A and assigning the elements of A to B in a transposed manner.
6. **Display matrix B on screen:** Once the matrix A is transposed into matrix B, the program displays the transposed matrix B on the screen.

The code works with loops for i and j.

In the flowchart, i and j are variables that are used to iterate through the rows and columns of a matrix.

* i is the row index, and it starts at 0 and increments by 1 until it reaches the number of rows in the matrix.
* j is the column index, and it starts at 0 and increments by 1 until it reaches the number of columns in the matrix.

## The code in C

**A screen shot of a computer program

Description automatically generated**

**A computer screen shot of text

Description automatically generated**

**A computer screen shot of a program

Description automatically generated**

**A screen shot of a computer code

Description automatically generated**

## 2.4. Special cases

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Description** | **State** | **Solution** |
| 1. | User inserts non-numeric values as inputs | solved | Checking the return value of *scanf*to confirm that a number was successfully read.  Clearing the input buffer if the input is invalid, allowing the user to try again without causing an infinite loop. |
| 4. | User inserts 1>n | solved | Program won’t accept it, due to the validation check and asks user to re-enter value. |
| 5. | User inserts 10<n | solved | Program won’t accept it, due to the validation check and asks user to re-enter value. |

Table 1. Special cases

# CONCLUSION

I conducted a program that transposes square matrix A and outputs it as matrix B. The algorithm was put together in “yEd” software and the main parts were input, process and output. Input asks for values n and entries for A[n][n], and checks if 1<n<10. If inserted values are not valid, then the programme asks to insert values again. Process puts to array the input matrix, interchanges corresponding columns and rows and outputs the result.

Conducting the algorithm was not that difficult, but I used the help of a programming website to confirm my ideas. Conducting the prototypes and main was also not difficult, since I had done similar exercises beforehand. I used BlackBox.ai to get functions based on my prototypes and main. The code was not prepared to deal with special cases. I had to figure out the missing parts and asked artificial intelligence to modify the functions such that the programme would not go to infinite loop, when non-number is inserted by user.

# EXTRA MATERIALS

**A computer screen shot of a black screen

Description automatically generated**

Picture 1. Code excecution, where inputs were valid.

**A computer screen with white text

Description automatically generated**

Picture 2. Code excecution, where inputs were invalid twice (n was invalid number: n=1; n was non-mumeric symbol: n=&).

# REFERENCES

GeeksForGeeks. (2024). <https://www.geeksforgeeks.org/program-to-find-transpose-of-a-matrix/> (9.12.2024).

Vladimir Viies. (2022). Homework formatting requirements. <https://www.tud.ttu.ee/im/Vladimir.Viies/materials/IAX0583_Programmeerimine%20I/pR0583en/Homeworks%20formatting.pdf> (15.11.2024).

Jason J. Sosa. (2019). <https://www.blackbox.ai/> (9.12.2024).