* outcome:
  1) Transpose of a matrix
  2) Result of addition, substraction and Multiplication of two matrix

* Theory
  2 dimension array
→ 2-D array are data structures that organize element in a tabular formate similar to a spreadsheet. They consists of rows and columns where each element
i) identified by it's row index & column

1) Row - major order

$$index = (i * num\ cols) + j$$

Where num cols is the number of column in the array

2) column Major order

$$index = (j * num\ Rows) + i$$

Where num Rows is the number of rows in the array

Matrix operation

1) Addition
To add two matrix their dimensions must be the same The sum obtained by adding corresponding element of the two
eg

$$A=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix} \qquad B=\begin{bmatrix}5 & 6\\ 7 & 8\end{bmatrix}$$

$$A+B=\begin{bmatrix}6 & 8\\ 10 & 12\end{bmatrix}$$

2) Substraction
substraction of matrix is similar to addition, where corresponding element are substracted

$$A=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix} \qquad B=\begin{bmatrix}5 & 6\\ 7 & 8\end{bmatrix}$$

$$A-B\begin{bmatrix}-4 & -4\\ -4 & -4\end{bmatrix}$$

3) Multiplication
To multiply two matrix A(m×n) and B (m×
The number of column in A must be equal to the number of rows in B The resul product matrix will have dimensional m×p

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \qquad B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$$A \times B \begin{bmatrix} 14 & 22 \\ 45 & 50 \end{bmatrix}$$

5) Transpose of matrix

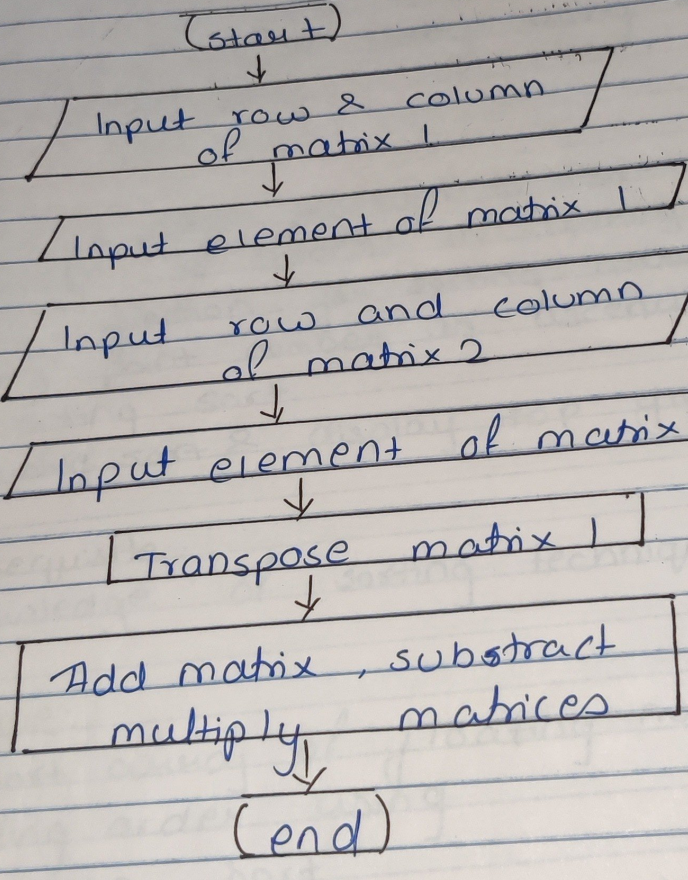
The transpose of a matrix is obtained by interchanging it's row and column if A is an $n \times n$ matrix it's transpose A^T will be an $n \times m$ matrix

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \qquad A^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

* Algorithm:

step 1) start
step 2) Input number of row & column of first matrix
step 3) Input element of first matrix
step 4) Input number of row & column of second
step 5) Input element of second matrix
step 6) function to transpose first i.e the element at row column C in the original is placed at row column r of the transpose
step 7) function to add - substract and multiply two matrix
step 8) stop.

* Flowchart



* Conclusion

By this way we can perform various operation on matrix successfully.

the elements of a list or array in a specific order typically in ascending or decending order sorting can be cutering into main types
1) Comparision based sorting
2) non - comparison based sorting

- Advantages of sorting
1) efficiency in searching
2) Improved Data management
3) facilities other algorithm

- disadvantage of sorting
1) Time complexity
2) space complexity
3) overhead

- sorting is a fundamental operation in computer science with various algorithm switched of different contexts

* selection sort

Selection sort divides the array into the two part. a sorted and un sorted part. It repeatedly select the smallest (or largest) element from the unsorted part and moves it to the end of the sorted part

Intial Array : [29. 10. 14. 37. 13]
minimum is 10 swap with 29

Array: [10. 29. 14. 37. 13]
minimum is 13 swap with 29

Array: [10. 13. 14. 37. 29]
minimum is 24. No swap Needed

Array: [10. 13. 14. 29. 37]

Sorted Array

Array: [10, 13, 14, 29, 37]

* Bubble Sort

Bubble sort repeatedly steps though the list compares adjacent elements & swaps them if they are in the wrong order This process repeats until the list sorted

Initial Array: [3, 1, 2]
compare 3 & 1 : swap

Array : [1, 3, 2]
compare 3 & 2 : swap

Array : [1, 2, 3]

Array : [1, 2, 3]
compare 2 & 3 : No swap

Array : [1, 2, 3]

* Algorithm

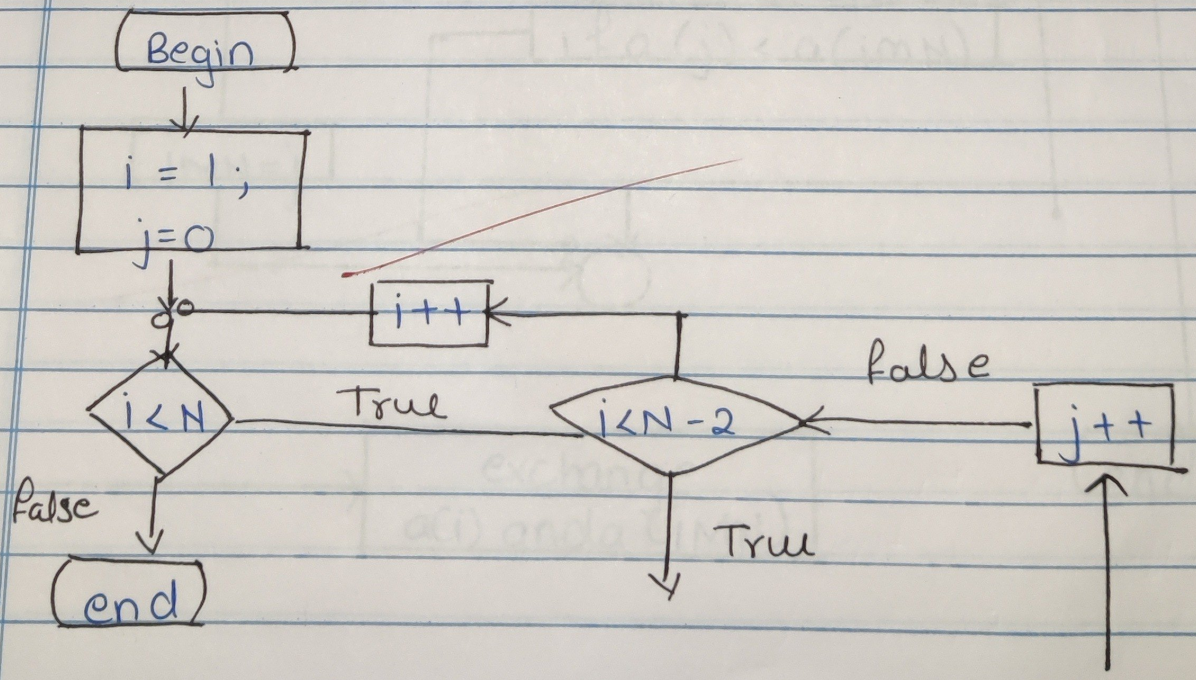
1) Bubble sort
step 1) We start from the first element
step 2) If the second element is greater than first elements swap them
step 3) Then compare second element and third element swap them then are not in the order
step 4) Continue the process until the last element
step 5) After the first iteration lock the last element Assuming the last element is the largest so that element doesn't follow the next iteration

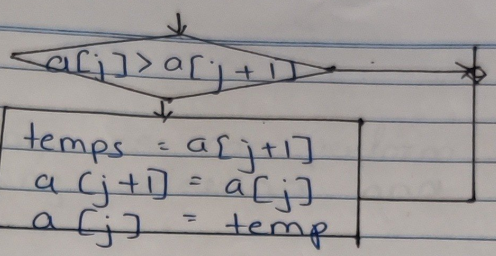
2) selection sort
step 1) place pointer on the first element & set as smallest value & is the smallest value
step 2) Compare the second element with he smallest. if the second element is smallest than the value, assign the second element as the

smallest Because 8 > 4. 4 still the smallest
step 3) Compare to third element Because
1<4 then 1 is the smallest
step4) continue compare to last element
because 5>1. then 1 still the smallest
step5) swap 1 to the first place That way first iteration has been completed and the smallest value already saved on the first place. which the will not participate in the Next iteration
step6) Repeat the process of finding the next smallest element and swapping it into the correct position until the entire array is sorted.

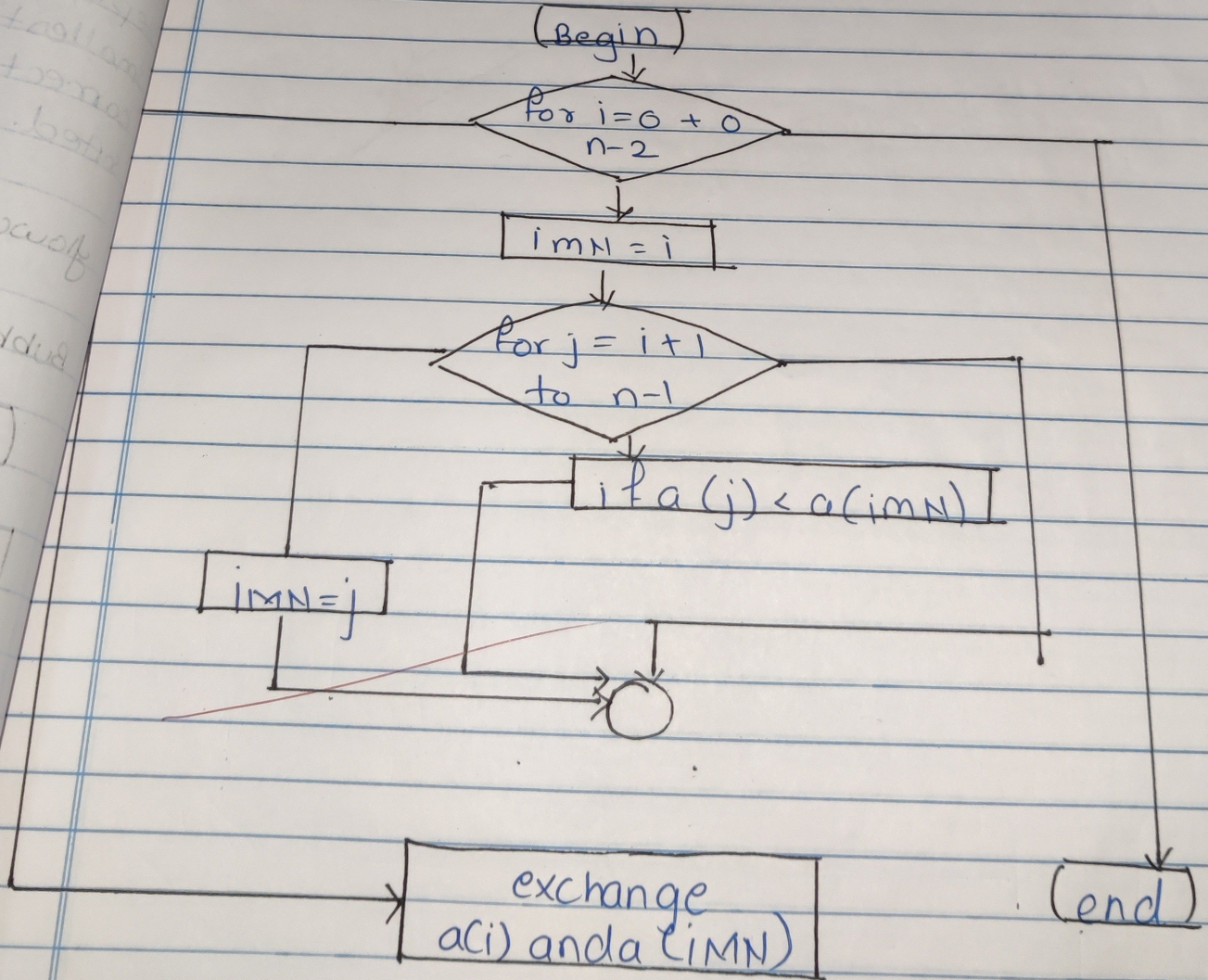
* flowchart

1) Bubble Sort





t Selection Sort



* Conclusion

By this way. we can perform sorting of array using seleetion and bubble sort