

## Assignment 2 - Algorithm and flowchart

### 1. Write an Algorithm to add two matrices

Step 1: Start

Step 2: Input rows and columns for the first matrix.

Step 3: Input elements to first matrix.

Step 4: Input rows and columns for the second matrix.

Step 5: Input elements to second matrix.

Step 6: Loop through each element of the matrix & add them together.

Step 7: Store the sum of each pair in the resultant matrix at the corresponding position.

Step 8: Repeat step 6 until all the elements are added.

Step 9: Return the resultant matrix as the sum of two matrices.

Step 10: Stop

### 2. Write an Algorithm to find the sum of elements in each row and column.

Step 1: Start

Step 2: Create an empty list to hold the total for each row.

Step 3: Create another empty list to hold the total for each column.

Step 4: Loop over every row in the matrix:

- 4.1: Set up a variable to track the sum of the current row.
- 4.2: Loop through each item in the current row:
- 4.3: Add each item to the current row's sum.
- 4.4: Store the total sum of the row in the list of row sums.

Step 5: Loop over every column in the matrix:

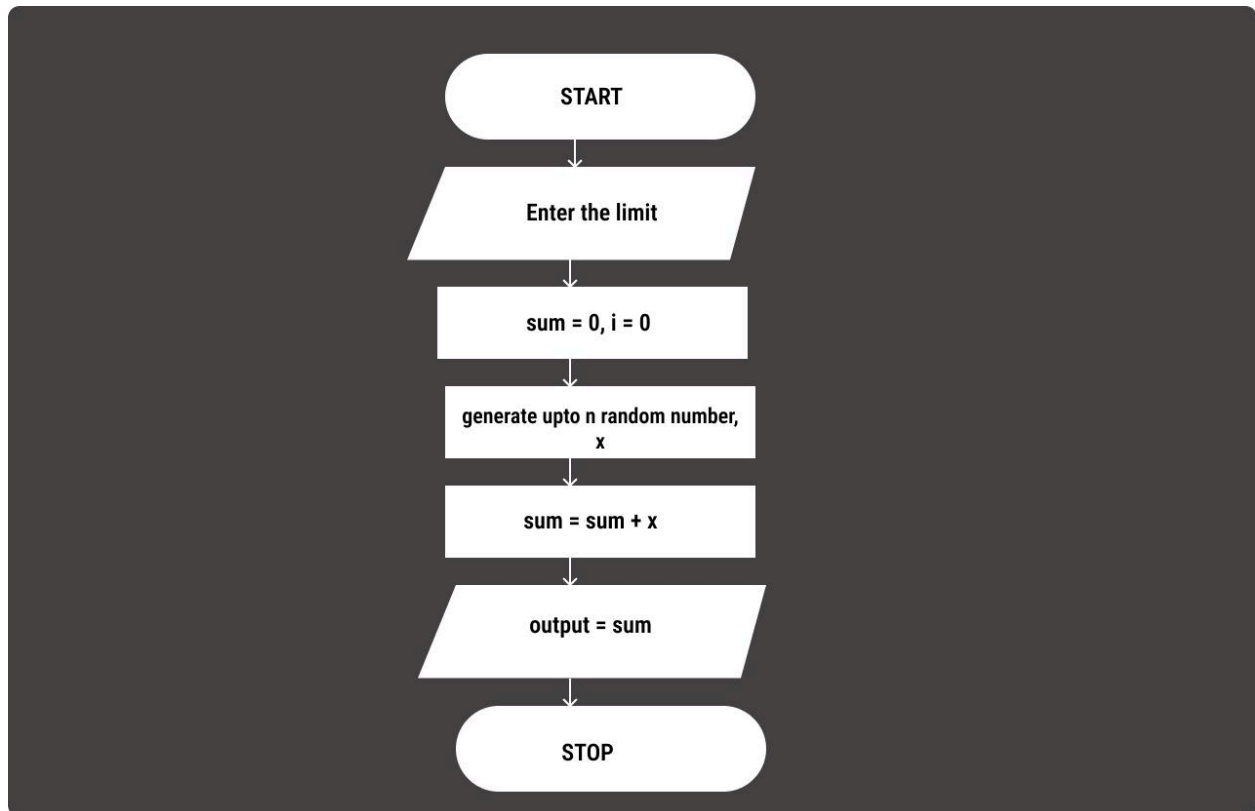
- 5.1: Set up a variable to track the sum of the current column.
- 5.2: Loop through each row to access the current column's value:
- 5.3: Add each value in the current column to the column's total.
- 5.4: Store the total sum of the column in the list of column sums.

Step 6: Display the list of row sums.

Step 7: Display the list of column sums.

Step 8: Stop

3. Draw a flowchart to find the sum of n random numbers.



4. One M x N grid is provided. Each cell has either grass (G-cell) or sand (S-cell). Among the four neighbours of an S-cell (say Csample), if at least three neighbours are G-cell, then next month, 'Csample' will become a G-cell. Similarly, if 'Csample' is a G-cell and at least three neighbours are S-cell, then next month 'Csample' will become an S-cell. In other cases, the status of a cell will not change. Illustrate an algorithm to determine any given grid's status after X months (X will be given as input).

Step 1: Start

Step 2: Initialize an array representing the initial landscape as grid

Step 3: Initialize a variable X to store the number of months to simulate.

Step 4: Create a temporary array of same size of the initial array to store the updated status of the cells.

Step 5: Iterate through each cell in grid

Step 6: If count the number of G-cell and S-cell near it, if the G-cell and it has 3 surrounding S-cell update it to temporary cell as S-cell

Step 7: If count the number of S-cell and G-cell near it, if the S-cell and it has 3 surrounding G-cell update it to temporary cell as G-cell

Step 8: Else keep the cell's status unchanged in the temporary grid

Step 9: Print the temporary grid which shows the final stage after X months.