

The program is for converting a matrix into its sparse representation. Here is the algorithm:-

- Step 1) Define MAX_R and MAX_C as the maximum number of rows and columns respectively.
- Step 2) Define the function `take_input()` and takes a 2D array as input and populates the matrix values by taking input from the user.
- Step 3) Define the function `sparse_input()` that takes the matrix, its number of rows and columns as input and returns the count of non-zero elements in the matrix. The function also populates a 2D array called `sparse` with the indices and values of non-zero elements in the matrix.
- Step 4) Define the function `print()` that takes the sparse matrix and its count as input and prints its representation in the form of three columns: row index, column index and value. In the main function, declare a 2D array called `matrix` with MAX_R rows and MAX_C columns and a 2D array called `sparse` with MAX_R * MAX_C rows and 3 columns (for row index, column index and value of the non-zero elements).
- Step 5) Take input for the number of rows and columns of the matrix from the user.
- Step 6) Call the `take_input()` function to populate the matrix values.
- Step 7) Call the `sparse_input()` function to populate the sparse matrix and get the count of non-zero elements in matrix.
- Step 8) Call the `print()` function to print the sparse matrix representation.

The program is for ~~converting~~ converting a matrix into its sparse representation using a linked list. ~~to~~ ~~as~~

→ Here is the algorithm:

- Step 1) Define ~~MAX_R~~ and ~~MAX_C~~ as the maximum number of rows and columns respectively.
- Step 2) Define a Node structure with row, col and val members representing the row index, column index and value of a non-zero element in the matrix. The structure also has a next pointer to connect nodes.
- Step 3) Define the function `take_input()` that takes a 2D array as input and populates the matrix values by taking input from the user.
- Step 4) Define the function `sparse_input()` that takes the matrix, its number of rows and columns as input, and returns the head of a linked list. The function creates a new node for each non-zero element in the matrix and populates its row, column and value. It then connects the nodes using the next pointer to create the linked list.
- Step 5) Define the function `print()` that takes the head of a linked list as input and prints its representation in the form of three columns: row index, column index and value.
- Step 6) In the main function, declare a 2D array called `matrix` with `MAX_R` rows and `MAX_C` columns.
- Step 7) Take input for the number of rows and columns of the matrix from the user.

- Step 8) Call the take input() function to populate the matrix values.
- Step 9) Call the sparse - input() function to create a linked list representing the sparse matrix.
- Step 10) Call the print() function to print the sparse matrix representation.