```
//40(a)Sparse matrix
/**
 * C program to check sparse matrix
 */

#include <stdio.h>
#define SIZE 3

int main()
{
    int A[SIZE][SIZE];
    int row, col, total=0;

    /* Input elements in matrix from user */
    printf("Enter elements in matrix of size 3x3: \n");
    for(row=0; row<SIZE; row++)
    {
        for(col=0; col<SIZE; col++)
        {
            scanf("%d", &A[row][col]);
        }
    }

    /* Count total number of zero elements in the matrix */
    for(row=0; row<SIZE; row++)
    {
        for(col=0; col<SIZE; col++)
        {
            /* If the current element is zero */
            if(A[row][col] == 0)
            {
                total++;
            }
        }
    }

    if(total >= (row * col)/2)
    {
        printf("\nThe given matrix is a Sparse matrix.");
    }
    else
    {
        printf("\nThe given matrix is not Sparse matrix.");
    }

    return 0;
}
Output
Enter elements in matrix of size 3x3:
1 0 0
4 5 0
0 0 0

The given matrix is a Sparse matrix.
```

```
//40(b)Transpose of matrix
#include <stdio.h>
int main() {
  int a[10][10], transpose[10][10], r, c;
  printf("Enter rows and columns: ");
  scanf("%d %d", &r, &c);

  // asssigning elements to the matrix
  printf("\nEnter matrix elements:\n");
  for (int i = 0; i < r; ++i)
  for (int j = 0; j < c; ++j) {
    printf("Enter element a%d%d: ", i + 1, j + 1);
    scanf("%d", &a[i][j]);
  }

  // printing the matrix a[][]
  printf("\nEntered matrix: \n");
  for (int i = 0; i < r; ++i)
  for (int j = 0; j < c; ++j) {
    printf("%d  ", a[i][j]);
    if (j == c - 1)
    printf("\n");
  }

  // computing the transpose
  for (int i = 0; i < r; ++i)
  for (int j = 0; j < c; ++j) {
    transpose[j][i] = a[i][j];
  }

  // printing the transpose
  printf("\nTranspose of the matrix:\n");
  for (int i = 0; i < c; ++i)
  for (int j = 0; j < r; ++j) {
    printf("%d  ", transpose[i][j]);
    if (j == r - 1)
    printf("\n");
  }
  return 0;
}
```

Output

```
Enter rows and columns: 2
3

Enter matrix elements:
Enter element a11: 1
Enter element a12: 4
Enter element a13: 0
Enter element a21: -5
Enter element a22: 2
Enter element a23: 7

Entered matrix:
```

```
1  4  0
-5  2  7

Transpose of the matrix:
1  -5
4  2
0  7
```