The Problem includes:

* Dynamic memory allocation.
* Lower Triangle
* Upper triangle
* Sum of Matrix
* Coloum and Row sum.
* 90 degree rotation

#include <stdio.h>  
#include<stdlib.h>  
int \*\* createMatrix(int row,int col){  
    int \*\* mat = (int \*\*) malloc(sizeof(int\*)\*row);  
    for(int i=0;i<row;i++){  
        \*(mat+i) = (int \*)malloc(sizeof(int)\*col);  
    }  
    return mat;  
}  
  
void getValue(int \*\* mat,int row,int col){  
    printf("Enter the Values\n");  
    for(int i=0;i<row;i++){  
        for(int j=0;j<col;j++){  
            scanf("%d",\*(mat+i)+j);  
        }  
    }  
   }  
  
  
void printMatrix(int \*\* mat,int row,int col){  
    printf("The Value inside the matrix are:\n");  
    for(int i=0;i<row;i++){  
        for(int j=0;j<col;j++){  
            printf("%d ",\*(\*(mat+i)+j));  
        }  
        printf("\n");  
    }  
}  
  
int checkLowerTriangularMatrix(int \*\* mat,int row,int col){  
    for(int i=0;i<row;i++){  
        for(int j=0;j<col;j++){  
            if(j>i && mat[i][j]!=0){  
            return 0;  
            }  
        }  
    }  
    return 1;  
}  
  
  
int checkUpperTriangularMatrix(int \*\* mat,int row,int col){  
    for(int i=0;i<row;i++){  
        for(int j=0;j<col;j++){  
            if(j<i && mat[i][j]!=0){  
            return 0;  
            }  
        }    
    }  
    return 1;  
}  
  
int sumOfMatrix(int \*\* mat,int row,int col){  
    int sum = 0;  
    for(int i=0;i<row;i++){  
        for(int j=0;j<col;j++){  
            sum = sum+mat[i][j];  
        }  
    }  
    return sum;  
}  
  
  
void printArray(int \* arr,int size){  
    for(int i=0;i<size;i++){  
        printf("%d ",\*(arr+i));  
    }  
    printf("\n");  
}  
  
int \* createArray(int size){  
    return (int \*)malloc(sizeof(int)\*size);  
}  
     
int \* colSum(int \*\* mat,int row,int col){  
    int \* res = createArray(col);  
    int sum = 0;  
    for(int j=0;j<col;j++){  
        sum = 0;  
        for(int i=0;i<row;i++){  
            sum = sum+mat[i][j];  
        }  
       \*(res+j) = sum;  
    }  
    return res;  
}  
  
int \* rowSum(int \*\* mat,int row,int col){  
    int \* res = createArray(row);  
    int sum = 0;  
    for(int i=0;i<row;i++){  
        sum = 0;  
        for(int j=0;j<col;j++){  
            sum = sum+mat[i][j];  
        }  
        \*(res+i) = sum;  
    }  
    return res;  
}  
int pd(int \*\*mat,int row,int col){  
    int sum=0;  
    for(int i=0;i<row;i++){  
        for(int j=0;j<col;j++){  
            if(i==j){  
                sum+=mat[i][j];  
            }  
        }  
    }  
    return sum;  
}  
int sd(int \*\*mat,int row,int col){  
    int sum=0;  
    for(int i=0;i<row;i++){  
        for(int j=0;j<col;j++){  
            if(i+j==col-1){  
                sum+=mat[i][j];  
            }  
        }  
    }  
    return sum;  
}  
int \*\*rotate90right(int \*\*mat,int row,int col){  
    int \*\*rotatedMat=createMatrix(row,col);  
    for(int i=0;i<row;i++){  
        for(int j=col-1;j>col;j--){  
            rotatedMat[i][j]=mat[j][i];  
        }  
    }  
    printf("AFTER ROTATION:\n");  
    printMatrix(rotatedMat,row,col);  
     
}  
  
int main(void) {  
    int row,col;  
    int \*\* mat;  
    printf("Enter the row and cols value\n");  
    scanf("%d%d",&row,&col);  
    mat = createMatrix(row,col);  
    getValue(mat,row,col);  
    printMatrix(mat,row,col);  
    if(checkLowerTriangularMatrix(mat,row,col)){  
        printf("The Given Matrix is a Lower Traingular Matrix\n");  
        printf("The sum of given Lower triangular Matrix is : %d\n",sumOfMatrix(mat,row,col));  
    } else {  
        printf("The Give Matrix is not a Lower Traingular Matrix\n");  
    }  
  
    if(checkUpperTriangularMatrix(mat,row,col)){  
        printf("The Give Matrix is a Upper Traingular Matrix\n");  
        printf("The sum of given Upper triangular Matrix is : %d\n",sumOfMatrix(mat,row,col));  
    } else {  
        printf("The Give Matrix is not a Upper Traingular Matrix:\n");  
    }  
    int \* rowSumArr = rowSum(mat, row, col);  
    printf("The row sum of the given matrix is:\n");  
    printArray(rowSumArr, row);  
    int \* colSumArr = colSum(mat, row, col);  
    printf("The col sum of the given matrix is:\n");  
    printArray(colSumArr, col);  
   
    printf("SUM OF PRIMARY DIAGONAL IS %d\n",pd(mat,row,col));  
    printf("SUM OF SECONDARY DIAGONAL IS %d\n",sd(mat,row,col));  
    rotate90right(mat,row,col);  
       return 0;  
}

