matShifting

Write a C function matShifting() that takes in a two-dimensional array ***a*** (***n***x***n*** square matrix with ***n***<=10) as parameter, shifts the column data of the array ***a*** to the right by one column position. The last column of the array will be shifted to the first column. The resultant data will be stored into another two-dimensional square array ***b*** (as parameter) which will be returned to the calling function via call by reference. For example, if the matrix of the array ***a*** (3x3 matrix) is:

$\longrightarrow$ Shift column data by one position

1 2 3
4 5 6
7 8 9

After program execution, the matrix of the array ***b*** (3x3) is:

3 1 2
6 4 5
9 7 8

A sample program template is given below:

```
#include <stdio.h>
#define M 10
void matShifting(int a[M][M], int b[M][M], int n);
int main()
{
   int a[M][M], b[M][M];
   int n,i,j;

   printf("Enter array (nxn) size (n<=10): \n");
   scanf("%d",&n);
   for (i=0; i<n; i++) {
      printf("Enter row %d: \n", i);
      for (j=0; j<n; j++)
         scanf("%d",&a[i][j]);
   }
   matShifting(a,b,n);
   printf("Array b: \n");
   for (i=0;i<n;i++) {
      for (j=0;j<n;j++)
         printf("%d ",b[i][j]);
      printf("\n");
   }
   return 0;
}
void matShifting(int a[M][M], int b[M][M], int n)
{
   /* Write your code here */
}
```

Some sample input and output sessions are given below:

(1) Test Case 1:

```
Enter array (nxn) size (n<=10):
3
Enter row 0:
1 2 3
Enter row 1:
4 5 6
Enter row 2:
7 8 9
Array b:
3 1 2
6 4 5
9 7 8
```

(2) Test Case 2:

```
Enter array (nxn) size (n<=10):
4
Enter row 0:
1 2 3 4
Enter row 1:
3 4 5 6
Enter row 2:
2 3 4 5
Enter row 3:
3 4 5 6
Array b:
4 1 2 3
6 3 4 5
5 2 3 4
6 3 4 5
```

(3) Test Case 3:

```
Enter array (nxn) size (n<=10):
5
Enter row 0:
1 2 3 4 5
Enter row 1:
3 4 5 6 7
Enter row 2:
2 3 4 5 6
Enter row 3:
3 4 5 6 7
Enter row 4:
3 -4 5 -6 -7
Array b:
5 1 2 3 4
7 3 4 5 6
6 2 3 4 5
7 3 4 5 6
-7 3 -4 5 -6
```