diagonals2D

Write a C function that accepts a two-dimensional array of integers ar, and the array sizes for the rows and columns as parameters, computes the sum of the elements of the two diagonals, and returns the sums to the calling function through the pointer parameters, sum1 and sum2, using call by reference. For example, if the rowSize is 3, colSize is 3, and the array ar is {1,2,3, 1,1,1, 4,3,2}, then sum1 is computed as 1+1+2=4, and sum2 is 3+1+4=8. The function prototype is given as follows:

```
void diagonals2D(int ar[][SIZE], int rowSize, int colSize, int *sum1, int *sum2);
```

A sample program template is given below to test the function:

```
#include <stdio.h>
#define SIZE 10
void diagonals2D(int ar[][SIZE], int rowSize, int colSize, int *sum1, int *sum2);
int main()
{
  int ar[SIZE][SIZE], rowSize, colSize;
  int i, j, sum1=0, sum2=0;

  printf("Enter row size of the 2D array: \n");
  scanf("%d", &rowSize);
  printf("Enter column size of the 2D array: \n");
  scanf("%d", &colSize);
  printf("Enter the matrix (%dx%d): \n", rowSize, colSize);
  for (i=0; i<rowSize; i++)
    for (j=0; j<colSize; j++)
      scanf("%d", &ar[i][j]);
  diagonals2D(ar, rowSize, colSize, &sum1, &sum2);
  printf("sum1=%d; sum2=%d\n",sum1,sum2);
}
void diagonals2D(int ar[][SIZE], int rowSize, int colSize, int *sum1, int *sum2)
{
     /* Write your code here */
}
```

Some sample input and output sessions are given below:

(1) Test Case 1:

```
Enter row size of the 2D array:
3
Enter column size of the 2D array:
3
Enter the matrix (3x3):
1 2 3
1 1 1
4 3 2
sum1=4; sum2=8
```

(2) Test Case 2:

```
Enter row size of the 2D array:
4
Enter column size of the 2D array:
4
Enter the matrix (4x4):
1 2 3 4
1 1 2 2
2 2 1 1
5 4 3 2
sum1=5; sum2=13
```

(3) Test Case 3:

```
Enter row size of the 2D array:
5
Enter column size of the 2D array:
5
Enter the matrix (4x4):
1 2 3 4 1
1 1 2 2 1
2 2 1 1 1
5 4 3 2 1
5 4 3 2 1
sum1=6; sum2=13
```