computeTotal2D

Write a C function that takes a 4x4 two-dimensional array matrix of integer numbers as a parameter. The function computes the total of the first three elements of each row of the array and stores it at the last element of the row. The function prototype is given as follows:

```
void computeTotal2D(int matrix[SIZE][SIZE]);
```

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

```
#include <stdio.h>
#define SIZE 4
void computeTotal2D(int matrix[SIZE][SIZE]);
int main()
{
  int a[SIZE][SIZE];
  int i,j;
  printf("Enter the matrix data (%dx%d): \n", SIZE, SIZE);
  for (i=0; i<SIZE; i++)
    for (j=0; j<SIZE; j++)
      scanf("%d", &a[i][j]);
  printf("computeTotal2D(): \n");
  computeTotal2D(a);
  for (i = 0; i < SIZE; i++) {
    for (j = 0; j < SIZE; j++)
      printf("%d ", a[i][j]);
    printf("\n");
  }
  return 0;
}
void computeTotal2D(int matrix[SIZE][SIZE])
{
  /* Write your code here */
}
```

Some test input and output sessions are given below:

(1) Test Case 1

```
Enter the matrix data (4x4):
1 2 3 4
2 3 4 5
3 4 5 6
6 7 8 9
computeTotal2D():
1 2 3 6
2 3 4 9
3 4 5 12
6 7 8 21
```

(2) Test Case 2

```
void computeTotal2D(int matrix[SIZE]
[SIZE])
{
   int i,j,sum=0;

   for(i=0;i<SIZE;i++)
   {
      for(j=0;j<=2;j++)
      {
         sum = sum + matrix[i][j];
      }
      matrix[i][SIZE-1] = sum;
      sum=0;
   }
}
```

Enter the matrix data (4x4):
1 2 3 -4
2 3 -4 5
3 -4 5 6
-6 7 8 9
computeTotal2D():
1 2 3 6
2 3 -4 1
3 -4 5 4
-6 7 8 9

(3) Test Case 3
Enter the matrix data (4x4):
1 -2 3 4
2 -3 4 5
3 -4 5 6
6 -7 8 9
computeTotal2D():
1 -2 3 2
2 -3 4 3
3 -4 5 4
6 -7 8 7