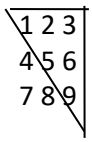
matTriSum

Write a C function matTriSum() that takes in a two-dimensional array ***x*** and size ***n*** (in nxn square matrix with *n*<=10) as parameters, computes the sum of the elements of the upper triangular matrix of the array, and returns the sum to the calling function. For example, for the following 3x3 (***n***=3) square matrix:



the sum of the upper triangular matrix in the array is 26 (=1+2+3+5+6+9) will be returned to the calling function.

A sample program template is given below:

```
#include <stdio.h>
#define M 10
int matTriSum(int x[M][M], int n);
int main()
{
   int x[M][M];
   int n,i,j,s;

   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",&x[i][j]);
   }
   s=matTriSum(x,n);
   printf("The sum is: %d\n",s);
   return 0;
}
int matTriSum(int x[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:
3 4 5
Enter row 2:
6 7 8
The sum is: 23
```

(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
The sum is: 40
```

(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
The sum is: 58
```