Simplified function

```
void add(int a[][MAX_SIZE], int b[][MAX_SIZE], int c[][MAX_SIZE],
  int rows, int cols)
{
  int i,j;
  for(i=0;i<rows;i++) {
    for(j=0;j<cols;j++)
      count+=2;
    count+=2;
  }
  count++;
}
```

the number of times that each statement is executed. We call this the **frequency**.

The frequency of a nonexecutable stmt is zero. Multiplying s/e by the frequency gives us the **total steps** for each statement.

Summing these totals, gives us the step count for the entire function.

Ex-1.12: Iterative function to sum a list of numbers

Statement	s/e	Frequency	Total steps
float sum(float list[], int n)	0	0	0
{	0	0	0
float tempsum=0;	1	1	1
int i;	0	0	0
for(i=0; i<n; i++)	1	n+1	n+1
tempsum+= list[i];	1	n	n
return tempsum;	1	1	1
}	0	0	0
Total			2n+3

Ex-1.13: Recursive function to sum a list of numbers.

Statement	s/e	Frequency	Total steps
float rsum(float list[], int n)	0	0	0
{	0	0	0
if(n)	1	n+1	n+1
return rsum(list,n-1)+list[n-1];	1	n	n
return list[0];	1	1	1
}	0	0	0