rSquare

Write a **recursive** function that returns the square of a positive integer number *num*, by computing the sum of odd integers starting with 1. The result is returned to the calling function. For example, if *num* = 4, then $4^2 = 1 + 3 + 5 + 7 = 16$ is returned; if *num* = 5, then $5^2 = 1 + 3 + 5 + 7 + 9 = 25$ is returned. Write two versions of the function. The function `rSquare1()` returns the result. The function `rSquare2()` returns the result through the parameter *result*. The function prototypes are:

```
int rSquare1(int num);
void rSquare2(int num, int *result);
```

For separate program testing: The following sample program template is given for testing the functions:

```
#include <stdio.h>
int rSquare1(int num);
void rSquare2(int num, int *result);
int main()
{
   int number, result=0;

   printf("Enter the number: \n");
   scanf("%d", &number);
   printf("rSquare1(): %d\n", rSquare1(number));
   rSquare2(number, &result);
   printf("rSquare2(): %d\n", result);
   return 0;
}
int rSquare1(int num)
{
   /* Write your code here */
}
void rSquare2(int num, int *result)
{
   /* Write your code here */
}
```

Some sample input and output sessions are given below:

(1) Test Case 1:

```
Enter a number:
4
rSquare1(): 16
rSquare2(): 16
```

(2) Test Case 2:

```
Enter a number:
1
rSquare1(): 1
rSquare2(): 1
```

(3) Test Case 3:

```
Enter a number:
12
rSquare1(): 144
rSquare2(): 144
```

(4) Test Case 4:

```
Enter a number:
5
rSquare1(): 25
```

```
rSquare2(): 25
```