rGcd

Write a **recursive** C function that computes the greatest common divisor and returns the result to the calling function via call by reference. For example, if `num1` is 4 and `num2` is 7, then `result` is 1; and if `num1` is 4 and `num2` is 32, then `result` is 4. Write the recursive function in two versions. The function `rGcd1()` computes and returns the result. The function `rGcd2()` computes and returns the result through the parameter `result` using call by reference. The function prototypes are given as follows:

```
int rGcd1(int num1, int num2);
void rGcd2(int num1, int num2, int *result);
```

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

```
#include <stdio.h>
int rGcd1(int num1, int num2);
void rGcd2(int num1, int num2, int *result);
int main()
{
   int n1, n2, result;

   printf("Enter 2 numbers: \n");
   scanf("%d %d", &n1, &n2);
   printf("rGcd1(): %d\n", rGcd1(n1, n2));
   rGcd2(n1, n2, &result);
   printf("rGcd2(): %d\n", result);
   return 0;
}
int rGcd1(int num1, int num2)
{
   /* Write your code here */
}
void rGcd2(int num1, int num2, int *result)
{
   /* Write your code here */
}
```

Some sample input and output sessions are given below:

(1) Test Case 1:

```
Enter 2 numbers:
4 7
rGcd1(): 1
rGcd2(): 1
```

(2) Test Case 2:

```
Enter 2 numbers:
32 4
rGcd1(): 4
rGcd2(): 4
```

(3) Test Case 3:

```
Enter 2 numbers:
4 38
rGcd1(): 2
rGcd2(): 2
```

(4) Test Case 4:

```
Enter 2 numbers:
32 38
rGcd1(): 2
rGcd2(): 2
```