(**rNumDigits**) Write a function that counts the number of digits for a non-negative integer. For example, 1234 has 4 digits. Write two versions of the function:

- (1) The function **rNumDigits1()** returns the result.
- (2) The function **rNumDigits2()** returns the result through the parameter *result*. The function prototypes are:

int rNumDigits1(int num);
void rNumDigits2(int num, int \*result);

### Sample input and output sessions:

```
Enter the number:

1234
rNumDigits1(): 4

Enter the number:

13579
rNumDigits2(): 5

Enter the number:

1234
rNumDigits1(): 4

Enter the number:

13579
rNumDigits2(): 5
```

#### Note:

When dealing with numbers, the integer division operator and modulus operator can be used to extract the digit value from the number.

```
#include <stdio.h>
int rNumDigits1(int num);
                                        main()
int main(){
                                                    number=123
   int number;
                                          rNumDigits1(number)
   printf("Enter the number: \n");
   scanf("%d", &number);
   printf("rNumDigits1(): %d\n",
       rNumDigits1(number));
                                         rNumDigits1(int num)
   return 0;
                                          num=123
                                           return rNumDigits1(num/10)+1;
 // By Returning Value
                                          rNumDigits1(int num)
int rNumDigits1(int num)
                                           num=12
                                           return rNumDigits1 (num/10)+1;
   if (num < 10)
      return 1;
   else
                                          rNumDigits1(int num)
      return rNumDigits1(num/10)+1;
                                            num=1
```

return 1;

Enter the number:

123

rNumDigits 1(): 3

```
int rNumDigits1(int num)
{
   if (num < 10)
     return 1;
   else
     return rNumDigits1(num/10)+1;
}</pre>
```

```
Enter the number:

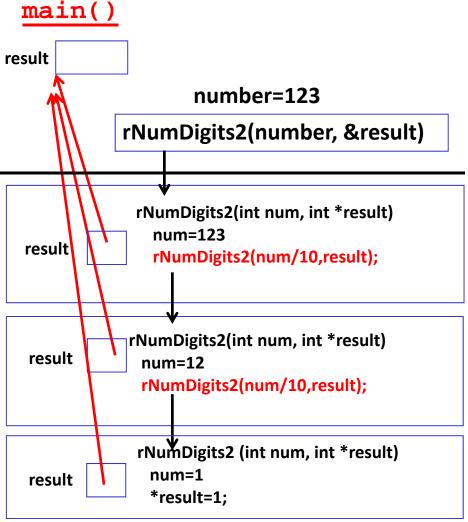
123

rNumDigits (): 3
```

```
number=123
 rNumDigits1(number)
rNumDigits1(int num)
 num=123
  return rNumDigits1(num/10)+1;
 rNumDigits1(int num)
  num=12
  return rNumDigits1 (num/10)+1;
 rNumDigits1(int num)
   num=1
   return 1;
```

```
void rNumDigits2(int num, int *result);
int main()
{
   int number, result;
   printf("Enter the number: \n");
   scanf("%d", &number);
   rNumDigits2(number, &result);
   printf("rNumDigits2(): %d\n", result);
   return 0;
}
   // Call by reference
```

```
void rNumDigits2(int num, int
*result)
{
   if (num < 10)
     *result = 1;
   else {
     rNumDigits2(num/10, result);
     *result = *result + 1;
   }
}</pre>
```



**Enter the number:** 

*123* 

#include <stdio.h>

rNumDigits2(): 3

```
Recursive Functions – Q1<sub>operation</sub>
#include <stdio.h>
void rNumDigits2(int num, int *result);
                                                                         sequence
int main()
                                                main()
                                                                            (i)
                                                              *result = 1:
                                                                            (ii)
                                                              *result = 2:
   int number, result;
                                              result
                                                                            (iii)
                                                              *result = 3;
   printf("Enter the number: \n");
                                                                 number=123
   scanf("%d", &number);
   rNumDigits2(number, &result);
                                                          rNumDigits2(number, &result)
   printf("rNumDigits2(): %d\n", result);
   return 0;
     // Call by reference
                                                            rNumDigits2(int num, int *result)
                                                             num=123
void rNumDigits2(int num, int
                                                result
                                                             rNumDigits2(num/10,result);
*result)
                                                               | *result=*result+1:
    if (num < 10)
                                                           rNumDigits2(int num, int *result)
        *result = 1;
                                                 result
                                                            num=12
    else {
                                                            rNumDigits2(num/10,result);
        rNumDigits2(num/10, result);
                                                                *result=*result+1: *
        *result = *result + 1;
                                                            rNumDigits2 (int num, int *result)
                                                             num=1
                                                 result
                                                                                 (i)
```

\*result=1:

**Enter the number:** 

*123* 

rNumDigits2(): 3

(rDigitPos) Write a function that returns the position of the first appearance of a specified digit in a positive number. The position of the digit is counted from the right and starts from 1. If the required digit is not in the number, the function should return 0. Write two versions of the function:

- (1) The function **rDigitPos1()** returns the result.
- (2) The function **rDigitPos2()** returns the result through the parameter *result*. The function prototypes are:

```
int rDigitPos1(int num, int digit);
void rDigitPos2(int num, int digit, int *result);
```

### Sample input and output sessions:

```
Enter the number:

1294567
Enter the digit:
5
rDigitPos1(): 3

Enter the number:

1294567
Enter the digit:
8
rDigitPos2(): 0
```

```
// By Returning Value
#include <stdio.h>
int rDigitPos1(int num, int digit);
int main()
   int number, digit;
  printf("Enter the number: \n");
   scanf("%d", &number);
  printf("Enter the digit: \n");
   scanf("%d", &digit);
  printf("rDigitPos1(): %d\n",
      rDigitPos1(number, digit));
   return 0;
int rDigitPos1(int num, int digit)
   int p;
   if (num % 10 == digit)
      return 1;
   else if (num < 10)
      return 0;
   else {
      p = rDigitPos1(num/10, digit);
      if (p > 0)
         return p + 1;
      else
         return 0;
```

```
main()
              number=1294567, digit=5
  rDigitPos1(number,digit)
rDigitPos1(int num,int digit)
  num=1294567, digit=5; num%10=7!=5
   p= rDigitPos1(num/10,digit);
              return p+1;
 rDigitPos1(int num,int digit)
  num=129456, digit=5; num%10=6!=5
  p= rDigitPos1(num/10,digit);
            return p+1;
 rDigitPos1(int num,int digit)
   num=12945, digit=5; num%10=5 ==5
   return 1;
    Enter the number:
    1294567
    Enter the digit:
    rDigitPos1(): 3
```

```
// By Returning Value
 #include <stdio.h>
 int rDigitPos1(int num, int digit);
 int main()
    int number, digit;
   printf("Enter the number: \n");
    scanf("%d", &number);
   printf("Enter the digit: \n");
    scanf("%d", &digit);
   printf("rDigitPos1(): %d\n",
       rDigitPos1(number, digit));
    return 0;
 int rDigitPos1(int num, int digit)
    int p;
    if (num % 10 == digit)
       return 1;
    else if (num < 10)
       return 0;
    else {
       p = rDigitPos1(num/10, digit);
       if (p > 0)
          return p + 1;
       else
          return 0;
```

```
main()
              number=1294567, digit=5
  rDigitPos1(number,digit)
rDigitPos1(int num,int digit)
  num=1294567, digit=5; num%10=7!=5
   p=rDigitPos1(num/10,digit)
              return p+1;
 rDigitPos1(int num,int digit)
  num=129456, digit=5; num%10=6!=5
  p= rDigitPos1(num/10,digit);
            return p+1;
 rDigitPos1(int num,int digit)
   num=12945, digit=5; num%10=5 ==5
   return 1;
    Enter the number:
    1294567
    Enter the digit:
    rDigitPos1(): 3
```

#### // Call by reference **Recursive Functions – Q2** #include <stdio.h> main() void rDigitPos2(int num, int digit, int \*pos); result int main(){ int number; number=1294567, digit=5 int digit, result=0; printf("Enter the number: \n"); rDigitPos2(number,digit,&result) scanf("%d", &number); printf("Enter the digit: \n"); scanf("%d", &digit); rDigitPos2(int num, int digit, int \*pos) rDigitPos2(number, digit, &result); num=1294567, digit=5; num%10=7!=5 printf("rDigitPos2(): %d\n",result); pos rDigitPos2(num/10,digit,result); return 0; \*pos+=1; void rDigitPos2(int num, int digit, int \*pos) rDigitPos2(int num, int digit, int \*pos) pos num=129456, digit=5; num%10=6!=5 if (num % 10 == digit) rDigitPos2(num/10,digit,result); \*pos = 1;\*pos +=1; else if (num < 10) \*pos = 0;rDigitPos2(int num, int digit, int \*pos) else { num=12945, digit=5; num%10=5 ==5 pos rDigitPos2(num/10, digit, pos); \*pos=1; (i.e. 12345%10==5) if (\*pos > 0)

\*pos += 1;

\*pos = 0;

else

Enter the number: 1294567
Enter the digit: 5
rDigitPos2(): 3

9

```
// Call by reference Recursive Functions - Q2
                                                                            operation
                                                                            sequence
#include <stdio.h>
void rDigitPos2(int num, int digit, int
                                                 main()
                                                                               (i)
                                                                 *result = 1:
*pos);
                                                                               (ii)
                                                                 *result = 2;
int main(){
                                               result
                                                                               (iii)
                                                                 *result = 3;
   int number;
                                                                    number=1294567, digit=5
   int digit, result=0;
   printf("Enter the number: ");
                                                            rDigitPos2(number,digit,&result)
   scanf("%d", &number);
   printf("Enter the digit: ");
   scanf("%d", &digit);
                                                              rDigitPos2(int num, int digit, int *pos)
   rDigitPos2(number, digit, &result);
                                                               num=1294567, digit=5; num%10=7!=5
   printf("rDigitPos2(): %d", result);
                                                   pos
                                                               rDigitPos2(num/10,digit,result);
   return 0;
                                                                  *pos+=1;
                                                                                      (iii)
void rDigitPos2(int num, int digit,
int *pos)
                                                             rDigitPos2(int num, int digit, int *pos)
                                                   pos
                                                              num=129456, digit=5; num 10=6!=5
   if (num % 10 == digit)
                                                              rDigitPos2(num/10,digit,result); (ii)
       *pos = 1;
                                                                  *pos +=1;
   else if (num < 10)
       *pos = 0;
                                                              rDigitPos2(int num, int digit, int *pos)
   else {
                                                               num=12945, digit=5; num%10=5 ==5
                                                   pos
      rDigitPos2(num/10, digit, pos);
                                                               *pos=1; (i.e. 12945%10==5)
       if (*pos > 0)
          *pos += 1;
                                                       Enter the number:
       else
                                                       1294567
          *pos = 0;
                                                       Enter the digit:
                                                                                            10
                                                       rDigitPos2(): 3
```

(**rSquare**) Write a 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:

- (1) The function **rSquare1()** returns the result.
- (2) The function **rSquare2()** returns the result through the parameter *result*. The function prototypes are:

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

### Sample input and output sessions:

```
Enter a number:

4
rSquare1(): 16

Enter a number:

5
rSquare2(): 25
```

# // By Returning Value Recursive Functions - Q3

```
#include <stdio.h>
                                            main()
                                                          x=3
int rSquare1(int num);
                                               rSquare1(x)
int main()
   int x;
                                             rSquare1(int num)
   printf("Enter a number: \n");
                                               num=3
   scanf("%d", &x);
  printf("rSquare1(): %d\n", rSquare1(x));
                                                return rSquare1(num-1)+(2*num-1);;
   return 0;
int rSquare1(int num)
                                             rSquare1(int num)
                                               num=2
   if (num == 1)
                                               return rSquare1(num-1)+(2*num-1);
      return 1;
   else
      return rSquare1(num-1)+(2*num -1);
                                              rSquare1(int num)
                                               num=1
           Enter a number:
                                               return 1;
```

rSquare1(): 9

# // By Returning Value Recursive Functions - Q3

```
#include <stdio.h>
                                            main()
                                                          x=3
int rSquare1(int num);
                                               rSquare1(x)
int main()
   int x;
                                             rSquare1(int num)
                                                                          1+3+5=9
   printf("Enter a number: \n");
                                               num=3
   scanf("%d", &x);
                                                return rSquare1(num-1)+(2*num-1);
  printf("rSquare1(): %d\n", rSquare1(x));
   return 0;
int rSquare1(int num)
                                             rSquare1(int num)
                                                                           1+3=4
                                               num=2
   if (num == 1)
                                               return rSquare1(num-1)+(2*num-1);
      return 1;
   else
      return rSquare1(num-1)+(2*num -1);
                                              rSquare1(int num)
                                               num=1
           Enter a number:
                                               return 1;
```

rSquare1(): 9

```
// Call by reference
#include <stdio.h>
void rSquare2(int num, int *result);
int main()
   int x, result;
   printf("Enter a number: \n");
   scanf("%d", &x);
   rSquare2(x, &result);
   printf("rSquare2(): %d\n", result);
   return 0;
void rSquare2(int num, int *result)
   if (num == 1)
      *result = 1;
   else {
      rSquare2(num-1, result);
      *result += (2*num -1);
```

```
main()
result
                      x=3
              rSquare2(x, &result)
                rSquare2(int num, int *result)
                 num=3
  result
                 rSquare2(num-1,result);
               rSquare2(int num, int *result)
   result
                num=2
                rSquare2(num-1,result);
                rSquare2 (int num, int *result)
                 num=1
  result
                 *result=1:
```

Enter a number: 3 rSquare2(): 9

# operation sequence

```
// Call by reference
#include <stdio.h>
void rSquare2(int num, int *result);
int main()
   int x, result;
   printf("Enter a number: \n");
   scanf("%d", &x);
   rSquare2(x, &result);
   printf("rSquare2(): %d\n", result);
   return 0;
void rSquare2(int num, int *result)
   if (num == 1)
      *result = 1;
   else {
      rSquare2(num-1, result);
      *result += (2*num -1);
```

```
main()
                 *result = 1; (i)
                 *result = 4; (ii)
                 *result = 9; (iii)
result
                     x=3
             rSquare2(x, &result)
               rSquare2(int num, int *result)
                 num=3
  result
                 rSquare2(num,result);
                                          (iii)
                    *result+=(2*num-1)
              rSquare2(int num, int *result)
  result
               num=2
               rSquare2(num,result);
                    rSquare2 (int num, int *result)
                num=1
  result
                                        (i)
                 *result=1:
```

Enter a number: 3 rSquare2(): 9