Write a function that counts the number of digits for a nonnegative integer. For example, 1234 has 4 digits. Write two versions of the function. The function rNumDigits1() returns the result. The function rNumDigits2() returns the result through the parameter *result*. The function prototypes are:

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

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

```
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.

#### By Returning Value

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

**Enter the number:** 

*123* 

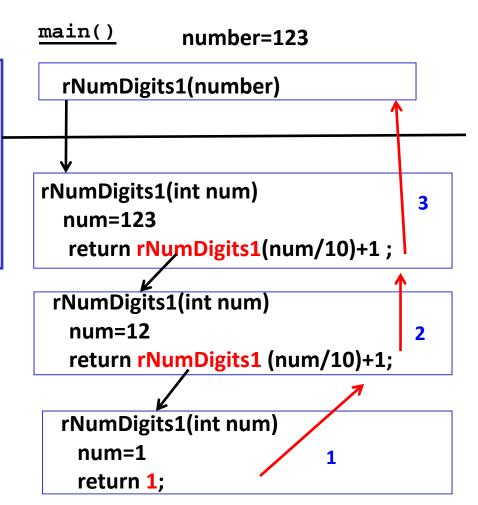
```
main()
             number=123
  rNumDigits1(number)
rNumDigits1(int num)
  num=123
  return rNumDigits1(num/10)+1;
 rNumDigits1(int num)
  num=12
  return rNumDigits1 (num/10)+1;
  rNumDigits1(int num)
   num=1
   return 1;
```

#### By Returning Value

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

#### **Enter the number:**

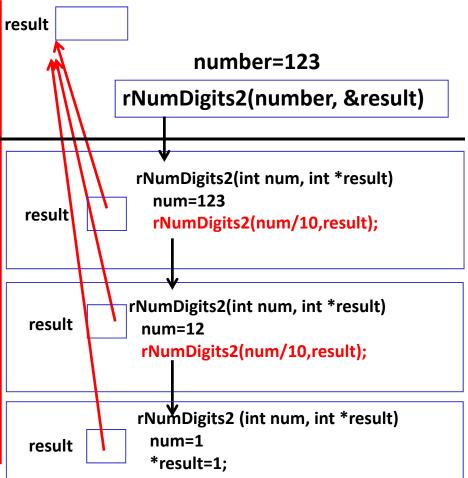
**123** 



#### Call by reference

```
#include <stdio.h>
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;
void rNumDigits2(int num, int *result)
   if (num < 10)
      *result = 1;
   else {
      rNumDigits2(num/10, result);
      *result = *result + 1;
```

### main()



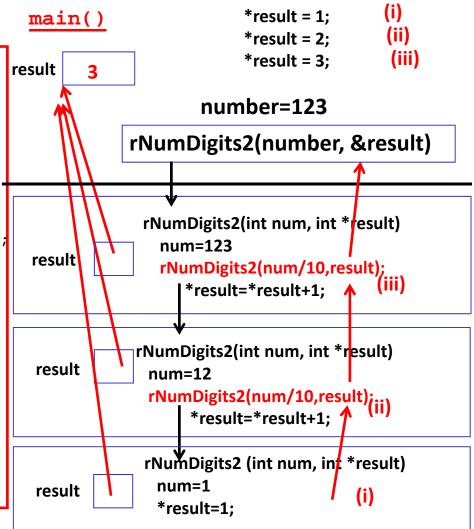
#### **Enter the number:**

**123** 

# operation sequence

#### Call by reference

```
#include <stdio.h>
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;
void rNumDigits2(int num, int *result)
   if (num < 10)
      *result = 1;
   else {
      rNumDigits2(num/10, result);
      *result = *result + 1;
```



#### **Enter the number:**

**123** 

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. The function rDigitPos1() returns the result. 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);
```

```
Enter the number:

1234567
Enter the digit:

6
rDigitPos1(): 2
Enter the number:

1234567
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",
      rDigitPosl(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=1234567, digit=5
       rDigitPos1(number,digit)
     rDigitPos1(int num,int digit)
       num=1234567, digit=5
        p= rDigitPos1(num/10,digit);
                   return p+1;
      rDigitPos1(int num,int digit)
       num=123456, digit=5
       p= rDigitPos1(num/10,digit);
                 return p+1;
      rDigitPos1(int num,int digit)
        num=12345, digit=5
        return 1; (12345%10==5)
Enter the number:
1234567
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",
      rDigitPosl(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=1234567, digit=5
       rDigitPos1(number,digit)
     rDigitPos1(int num,int digit)
       num=1234567, digit=5
                                         3
        p= rDigitPos1(num/10,digit);
                   return p+1;
      rDigitPos1(int num,int digit)
       num=123456, digit=5
                                        2
       p= rDigitPos1(num/10,digit);
                 return p+1;
      rDigitPos1(int num,int digit)
        num=12345, digit=5
                                      1
        return 1; (12345%10==5)
Enter the number:
1234567
Enter the digit:
rDigitPos1(): 3
```

#### Call by reference

### **Recursive Functions – Q2**

```
#include <stdio.h>
void rDigitPos2(int num, int digit, int
                                                  main()
*pos);
int main()
                                                result
   int number;
                                                                     number=1234567, digit=5
   int digit, result=0;
                                                             rDigitPos2(number,digit,&result)
   printf("Enter the number: \n");
   scanf("%d", &number);
   printf("Enter the digit: \n");
   scanf("%d", &digit);
                                                               rDigitPos2(int num, int digit, int *result)
   rDigitPos2(number, digit, &result);
                                                                num=1234567, digit=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 *result)
                                                    pos
                                                               num=123456, digit=5
   if (num % 10 == digit)
                                                               rDigitPos2(num/10,digit,result);
       *pos = 1i
                                                                    *pos +=1;
   else if (num < 10)
       *pos = 0;
                                                               rDigitPos2(int num, int digit, int *result)
   else {
                                                                num=12345, digit=5
                                                    pos
       rDigitPos2(num/10, digit, pos);
                                                                *pos=1; (i.e. 12345%10==5)
       if (*pos > 0)
          *pos += 1;
                                                  Enter the number:
       else
                                                  1234567
                                                  Enter the digit:
          *pos = 0;
                                                  rDigitPos1(): 3
```

#### Call by reference

#include <stdio.h>

### **Recursive Functions – Q2**

1234567

**Enter the digit:** 

rDigitPos1(): 3

```
operation sequence
```

10

```
void rDigitPos2(int num, int digit, int
*pos);
int main()
   int number;
   int digit, result=0;
  printf("Enter the number: ");
   scanf("%d", &number);
  printf("Enter the digit: ");
  scanf("%d", &digit);
  rDigitPos2(number, digit, &result);
  printf("rDigitPos2(): %d", result);
  return 0;
void rDigitPos2(int num, int digit, int
*pos)
   if (num % 10 == digit)
      *pos = 1i
   else if (num < 10)
      *pos = 0;
   else {
      rDigitPos2(num/10, digit, pos);
      if (*pos > 0)
         *pos += 1;
      else
         *pos = 0;
```

```
(i)
                              *result = 1:
  main()
                                               (ii)
                              *result = 2:
                                                (iii)
                              *result = 3;
result
                        number=1234567, digit=5
               rDigitPos2(number,digit,&result)
                 rDigitPos2(int num, int digit, int *result)
                   num=1234567, digit=5
    pos
                   rDigitPos2(num/10,digit,result);
                      *pos+=1;
                                              (iii)
                rDigitPos2(int num, int digit, int *result)
    pos
                 num=123456, digit=5
                 rDigitPos2(num/10,digit,result); (ii)
                       *pos +=1;
                 rDigitPos2(int num, int digit, int *result)
                  num=12345, digit=5
    pos
                                                 (i)
                   *pos=1; (i.e. 12345%10==5)
  Enter the number:
```

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. 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);
```

```
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");
   scanf("%d", &x);
                                               num=3
   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
                                                return 1;
```

**Enter a number:** 

rSquare1(): 9

# By Returning Value Recursive Functions - Q3

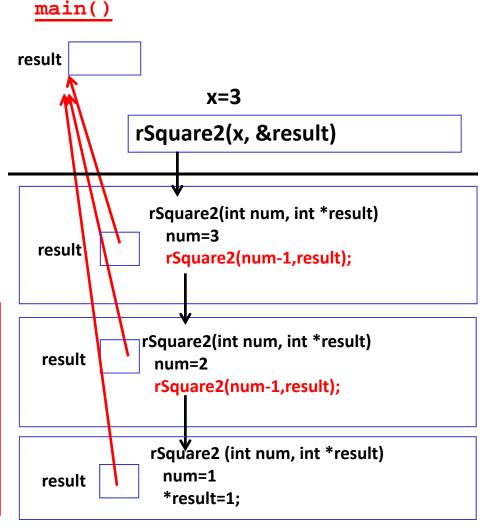
```
#include <stdio.h>
                                             main()
                                                          x=3
int rSquare1(int num);
                                               rSquare1(x)
int main()
   int x;
                                             rSquare1(int num)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

#### **Enter a number:**

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);
```



#### **Enter the number:**

*123* 

# 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 = 1i
   else {
      rSquare2(num-1, result);
      *result += (2*num -1);
```

```
(i)
                           *result = 1:
 main()
                                           (ii)
                           *result = 4:
                                            (iii)
                           *result = 9:
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 the number:**

123

```
#include <stdio.h>
#define BLANK ' '
void saveChar();
int main()
   printf("Enter your word and end it
with a space => ");
   saveChar();
  putchar('\n');
   return 0;
void saveChar()
   char ch;
   ch = getchar();
   if (ch != BLANK)
      saveChar();
   else
      putchar('\n');
   putchar(ch);
```

Enter your word and end it with a space => ward

### What is the output?

You may try to enter the code and run the code to see the output.

```
#include <stdio.h>
#define BLANK ' '
void saveChar();
int main()
  printf("Enter your word and end it
with a space => ");
  saveChar();
  putchar('\n');
  return 0;
void saveChar()
   char ch;
  ch = getchar();
   if (ch != BLANK)
      saveChar();
   else
      putchar('\n');
  putchar(ch);
```

Enter your word and end it with a space => ward

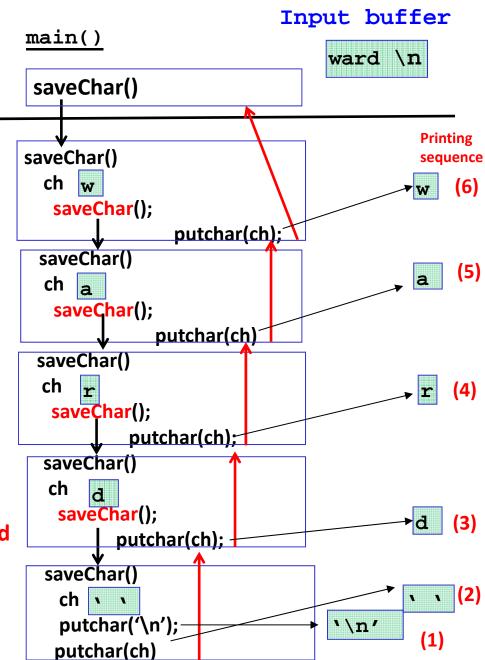
draw

Input buffer main() ward \n saveChar() saveChar() ch w saveChar(); saveChar() ch a saveChar(); saveChar() ch saveChar(); saveChar() ch saveChar(); saveChar() ch v v putchar('\n'); '\n' **(1)** putchar(ch)

```
#include <stdio.h>
#define BLANK ' '
void saveChar();
int main()
  printf("Enter your word and end it
with a space => ");
   saveChar();
  putchar('\n');
  return 0;
void saveChar()
   char ch;
  ch = getchar();
   if (ch != BLANK)
      saveChar();
   else
      putchar('\n');
  putchar(ch);
```

Enter your word and end it with a space => ward

draw



```
#include <stdio.h>
#define BLANK ' '
void saveChar();
int main()
   printf("Enter your word and end it
with a space => ");
   saveChar();
  putchar('\n');
   return 0;
void saveChar()
   char ch;
   ch = getchar();
   if (ch != BLANK)
      saveChar();
   else
      putchar('\n');
   putchar(ch);
```

Enter your word and end it with a space => ward

#### draw

Please note that there is a blank character at the end of the input word before the "enter" key is pressed.

Basically, this program prints an input string, which ends with a space character, in the reversed order.