

rDigitPos

Write a **recursive** 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 pointer parameter `pos`. The function prototypes are given as follows:

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

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

```
#include <stdio.h>
int rDigitPos1(int num, int digit);
void rDigitPos2(int num, int digit, int *pos);
int main()
{
    int number, digit, result=0;

    printf("Enter the number: \n");
    scanf("%d", &number);
    printf("Enter the digit: \n");
    scanf("%d", &digit);
    printf("rDigitPos1(): %d\n", rDigitPos1(number, digit));
    rDigitPos2(number, digit, &result);
    printf("rDigitPos2(): %d\n", result);
    return 0;
}
int rDigitPos1(int num, int digit)
{
    /* Write your code here */
}
void rDigitPos2(int num, int digit, int *pos)
{
    /* Write your code here */
}
```

Some sample input and output sessions are given below:

- (1) Test Case 1:
Enter the number:
1234567
Enter the digit:
6
rDigitPos1(): 2
rDigitPos2(): 2
- (2) Test Case 2:
Enter the number:
1234567
Enter the digit:
8
rDigitPos1(): 0
rDigitPos2(): 0
- (3) Test Case 3:
Enter the number:
1357
Enter the digit:

3
rDigitPos1(): 3
rDigitPos2(): 3

(4) Test Case 4:
Enter the number:
6
Enter the digit:
6
rDigitPos1(): 1
rDigitPos2(): 1