digitValue

Write a function that returns the value of the k^{th} digit (k>0) from the right of a non-negative integer *num*. For example, if num is1234567 and k is 3, the function will return 5 and if num is 1234 and k is 8, the function will return 0. Write the function in two versions. The function **digitValue1()** returns the result, while **digitValue2()** passes the result through pointer parameter result. The prototypes of the function are given below:

```
int digitValue1(int num, int k);
void digitValue2(int num, int k, int *result);
```

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

```
#include <stdio.h>
int digitValue1(int num, int k);
void digitValue2(int num, int k, int *result);
int main()
{
  int num, digit, result=-1;

  printf("Enter the number: \n");
  scanf("%d", &num);
  printf("Enter k position: \n");
  scanf("%d", &digit);
  printf("digitValue1(): %d\n",  digitValue1(num, digit));
  digitValue2(num, digit, &result);
  printf("digitValue2(): %d\n", result);
  return 0;
}
int digitValue1(int num, int k)
{
  /* Write your code here */
}
void digitValue2(int num, int k, int *result)
{
  /* Write your code here */
}
```

Some sample input and output sessions are given below:

(1) Test Case 1:

```
Enter the number:
234567
Enter k position:
3
digitValue1(): 5
digitValue2(): 5
```

(2) Test Case 2:

```
Enter the number:
```

```
234567
Enter k position:
1
digitValue1(): 7
digitValue2(): 7
```

(3) Test Case 3:

```
Enter the number:
123
Enter k position:
8
digitValue1(): 0
digitValue2(): 0
```

```
int digitValue1(int num, int k)
{
    int count =0;
    int bcount=0;
    int remainder;
    int i;
    i=num;
    while(i!=0)
    {
        i = i/10;
        count++;
    }
    if(count<k)
    {
        return 0;
    }
    else
    {
        while(num!=0)
        {
            bcount++;
            remainder = num%10;
            num=num/10;
            if(bcount==k)
            {
                return remainder;
            }
        }
    }
}
void digitValue2(int num, int k, int *result)
{
    int count =0;
    int bcount=0;
    int remainder;
    int i;
    i=num;
    while(i!=0)
    {
        i = i/10;
        count++;
    }
    if(count<k)
    {
        *result = 0;
    }
    else
    {
        while(num!=0)
        {
            bcount++;
            remainder = num%10;
            num=num/10;
            if(bcount==k)
            {
                *result = remainder;
            }

        }
    }
}
```