countEvenDigits

Write a C function to count the number of even digits, i.e. digits with values 0,2,4,6,8 in a positive number (>0). For example, if number is 1234567, then 3 will be returned. Write the function in two versions. The function countEvenDigits1() returns the result, while the function countEvenDigits2() returns the result via the pointer parameter, count. The function prototypes are given below:

```
int countEvenDigits1(int number);
void countEvenDigits2(int number, int *count);
```

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

```
#include <stdio.h>
int countEvenDigits1(int number);
void countEvenDigits2(int number, int *count);
int main()
{
  int number, result;

  printf("Enter a number: \n");
  scanf("%d", &number);
  printf("countEvenDigits1(): %d\n", countEvenDigits1(number));
  countEvenDigits2(number, &result);
  printf("countEvenDigits2(): %d\n", result);
  return 0;
}
int countEvenDigits1(int number)
{
  /* Write your code here */
}
void countEvenDigits2(int number, int *count)
{
  /* Write your code here */
}
```

Some sample input and output sessions are given below:

(1) Test Case 1:

```
Enter a number:
1234567
countEvenDigits1(): 3
countEvenDigits2(): 3
```

(2) Test Case 2:

```
Enter a number:
2468
countEvenDigits1(): 4
countEvenDigits2(): 4
```

(3) Test Case 3:

```
int countEvenDigits1(int number)
{
  int count=0;
  int remainder;

  while (number!=0)
  {
     remainder = number%10;
     if(remainder%2==0)
     {
        count++;
     }
     number = number/10;
  }
   return count;
}
void countEvenDigits2(int number, int
*count)
{
  int remainder;
  int cnt=0;
  while (number!=0)
  {
     remainder = number%10;
     if(remainder%2==0)
     {
        cnt++;
     }
     number = number/10;
  }
   *count = cnt;
}
```

```
Enter a number:
1357
countEvenDigits1(): 0
countEvenDigits2(): 0
```