sumSqDigits

Write a function that takes an integer argument num and returns the sum of the squares of the digits of the integer. For example, given the number 3418, the function should return 90, i.e. 3*3+4*4+1*1+8*8. Write two iterative versions of the function. The function **sumSqDigits1()** returns the computed result, while **sumSqDigits2()** passes the result through the pointer parameter result. The function prototypes are given as follows:

```
int sumSqDigits1(int num);
void sumSqDigits2(int num, int *result)
```

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

```
#include  <stdio.h>
int sumSqDigits1(int num);
void sumSqDigits2(int num, int *result);
int main()
{
  int num, result;

  printf("Enter a number: \n");
  scanf("%d", &num);
  printf("sumSqDigits1(): %d\n", sumSqDigits1(num));
  sumSqDigits2(num, &result);
  printf("sumSqDigits2(): %d\n", result);
  return 0;
}
int sumSqDigits1(int num)
{
  /* Write your code here */
}
void sumSqDigits2(int num, int *result)
{
  /* Write your code here */
}
```

Some sample input and output sessions are given below:

(1) Test Case 1

```
Enter a number:
24
sumSqDigits1(): 20
sumSqDigits2(): 20
```

(2) Test Case 2

```
Enter a number:
3418
sumSqDigits1(): 90
sumSqDigits2(): 90
```

```
int sumSqDigits1(int num)
{
   int sum = 0;
   int remainder;

   while(num!=0)
   {
      remainder = num%10;
      sum = sum +
remainder*remainder;
      num = num/10;
   }
   return sum;
}
void sumSqDigits2(int num, int
*result)
{
   int sum = 0;
   int remainder;

   while(num!=0)
   {
      remainder = num%10;
      sum = sum +
remainder*remainder;
      num = num/10;
   }
   *result = sum;
}
```

(3) Test Case 3

```
Enter a number:
102
sumSqDigits1(): 5
sumSqDigits2(): 5
```