strOverWrite

Write a C function strOverWrite() that takes in two strings ***s1*** and ***s2***, and an integer ***pos*** as parameters, inserts ***s1*** into ***s2*** at the position specified by ***pos***, while overwriting the part of ***s1*** that corresponds to the insertion. The function should ensure the overwrite operation does not change the size of ***s1***. If the operation is successful, the function should return the number of characters that were overwritten. For example, if ***s1*** = "this text is short", ***s2*** = "long" and ***pos*** = 13, the resulting ***s1*** would be "this text is longt", and the function returns a value of 4. In this function, there is no need to check input errors.

A sample program template is given below:

```
#include <stdio.h>
#include <string.h>
int strOverWrite(char *s1, char *s2, int pos);
int main()
{
   char s1[40], s2[40], *p;
   int pos, total;

   printf("Enter string 1: \n");
   fgets(s1, 80, stdin);
   if (p=strchr(s1,'\n')) *p = '\0';
   printf("Enter string 2: \n");
   fgets(s2, 80, stdin);
   if (p=strchr(s2,'\n')) *p = '\0';
   printf("Enter position: \n");
   scanf("%d", &pos);
   total = strOverWrite(s1, s2, pos);
   printf("strOverWrite(): %s %d\n", s1, total);
   return 0;
}
int strOverWrite(char *s1, char *s2, int pos)
{
   /* Write your code here */
}
```

Some sample input and output sessions are given below:

(1) Test Case 1:

```
Enter string 1:
this text is short
Enter string 2:
long
Enter position:
6
strOverWrite(): this tlongis short 4
```

(2) Test Case 2:

```
Enter your choice:
1
Enter string 1:
this text is short
Enter string 2:
long
Enter position:
```

```
13
strOverWrite(): this text is longt 4
```

(3) Test Case 3:

```
Enter string 1:
this text is short
Enter string 2:
longg
Enter position:
13
strOverWrite(): this text is longg 5
```

(4) Test Case 4:

```
Enter string 1:
this text is short
Enter string 2:
longgt
Enter position:
13
strOverWrite(): this text is longg 5
```