stringcmp

Write a C function that compares the string pointed to by *s1* to the string pointed to by *s2*. If the string pointed to by *s1* is greater than, equal to, or less than the string pointed to by *s2*, then it returns 1, 0 or –1 respectively. Write the code for the function without using any of the standard C string library functions. The function prototype is given as follows:

```
int stringcmp(char *s1, char *s2);
```

A sample template for the program is given below:

```
#include <stdio.h>
#include <string.h>
#define INIT_VALUE 999
int stringcmp(char *s1, char *s2);
int main()
{
  char source[80], target[80], *p;
  int result = INIT_VALUE;

  printf("Enter a source string: \n");
  fgets(source, 80, stdin);
  if (p=strchr(source,'\n')) *p = '\0';
  printf("Enter a target string: \n");
  fgets(target, 80, stdin);
  if (p=strchr(target,'\n')) *p = '\0';
  result = stringcmp(source, target);
  if (result == 1)
    printf("stringcmp(): greater than");
  else if (result == 0)
    printf("stringcmp(): equal");
  else if (result == -1)
    printf("stringcmp(): less than");
  else
    printf("stringcmp(): error");
  return 0;
}
int stringcmp(char *s1, char *s2)
{
    /* Write your code here */
}
```

Some test input and output sessions are given below:

(1) Test Case 1:
Enter a source string:
abc
Enter a target string:

abc
stringcmp(): equal

(2) Test Case 2:
Enter a source string:
abcdefg
Enter a target string:
abcde123
stringcmp(): greater than

(3) Test Case 3:
Enter a source string:
abc123
Enter a target string:
abcdef
stringcmp(): less than

(4) Test Case 4:
Enter a source string:
abcdef
Enter a target string:
abcdefg
stringcmp(): less than