minCharToEnd

Write a C function minCharToEnd() that accepts a character string ***str*** as parameter, finds the smallest character from the string (based on ASCII values), and moves it to the end of the string. E.g., if the string is "adecb", then the string will be "decba" after executing the function. You may assume that the string contains only lower case characters. The string will be passed to the caller via call by reference. If more than one smallest characters are in the string, then the **first appearance** of the smallest character will be moved to the end of the string. For example, if the string is "adeabe", then the resultant string will be "deabea".

A sample program template is given below:

```
#include <stdio.h>
#include <string.h>
void minCharToEnd(char *str);
int main()
{
   char str[80];

   printf("Enter a string: \n");
   scanf("%s",str);
   minCharToEnd(str);
   printf("minCharToEnd(): %s",str);
   return 0;
}
void minCharToEnd(char *str)
{
    /* Write your code here */
}
```

Some test input and output sessions are given below:

(1) Test Case 1:

```
Enter a string:
adecb
minCharToEnd(): decba
```

(2) Test Case 2:

```
Enter a string:
adeabe
minCharToEnd(): deabea
```

(3) Test Case 3:

```
Enter a string:
cba
minCharToEnd(): cba
```

(4) Test Case 4:

```
Enter a string:
ab
minCharToEnd(): ba
```

swap values

```
#include <stdio.h>
#include <string.h>
void minCharToEnd(char *str);
int main()
{
  char str[80];

  printf("Enter a string: \n");
  scanf("%s",str);
  minCharToEnd(str);
  printf("minCharToEnd(): %s",str);
  return 0;
}
void minCharToEnd(char *str)
{
   char temp;
   int i;

   for(i=0;str[i]!='\0';i++)
   {
      if(str[i]<str[i+1])
      {
         temp = str[i];
         str[i] = str[i+1];
         str[i+1] = temp;
      }
   }
}
```