sortCharStr

Write a C function sortCharStr() that accepts a character string ***str*** as parameter, sorts the characters in the string according to ascending order and returns the resultant string to the calling function via ***str***. You may assume that the character string contains only lower case characters. For example, if ***str*** = "goodies", the resultant string will be "degioos".

A sample program template is given below:

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

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

Some test input and output sessions are given below:

(1) Test Case 1:

```
Enter a string:
big
sortCharStr(): bgi
```

(2) Test Case 2:

```
Enter a string:
goodies
sortCharStr(): degioos
```

(3) Test Case 3:

```
Enter a string:
have
sortCharStr(): aehv
```

(4) Test Case 4:

```
Enter a string:
difficulty
sortCharStr(): cdffiiltuy
```

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

  printf("Enter a string: \n");
  scanf("%s",str);
  sortCharStr(str);
  printf("sortCharStr(): %s\n", str);
  return 0;
}
void sortCharStr(char *str)
{
   char c,*p,*q,*r;
   for(p=str;*p;p++)
   {
     for(q=r=p;*q;q++)
     {
       if(*r>*q)
         r=q;
     }
     if(r!=p)
     {
       c=*r;
       *r=*p;
       *p=c;
     }
   }
}
```

swap the values

str = 'ab'
&p = &str =>*p = 'b'
&q = &r = &p
first iteration for 2nd loop:'b'>'b'->false
q++(*q='a')
second iteration for 2nd for loop:'b'>'a' ->true
&r=&q (char 'a')

since &r(stores char 'a')!=&p(stores char 'b')
'b' and 'a' swap