strIntersect

Write the C function `strIntersect()` that takes in three strings *str1*, *str2* and *str3* as parameters, stores the same characters that appeared in both *str1* and *str2* into the string, and returns *str3* to the calling function via call by reference. For example, if *str1* is `"abcdefghijk"` and *str2* is `"123i4bc78h9"`, then *str3* is `"bchi"` will be returned to the calling function after executing the function. If there is no common characters in the two strings, *str3* will be a null string. You may assume that each string contains unique characters in the string, i.e. the characters contained in the same string will not be repeated. The function prototype is:

```
void strIntersect(char *str1, char *str2, char *str3);
```

A sample C program to test the function is given below:

```
#include <stdio.h>
void strIntersect(char *str1, char *str2, char *str3);
int main()
{
   char str1[50],str2[50],str3[50];

   printf("Enter str1: \n");
   scanf("%s",str1);
   printf("Enter str2: \n");
   scanf("%s",str2);
   strIntersect(str1, str2, str3);
   if (*str3 == '\0')
      printf("strIntersect(): null string\n");
   else
      printf("strIntersect(): %s\n", str3);
   return 0;
}
void strIntersect(char *str1, char *str2, char *str3)
{
   /* Write your code here */
}
```

```
void strIntersect(char *str1, char *str2, char *str3)
{
   int i,j;
   int k=0;

   for(i=0;str1[i]!='\0';i++)
   {
      for(j=0;str2[j]!='\0';j++)
      {
         if(str1[i]==str2[j])
         {
            str3[k] = str1[i];
            k++;
         }
      }
   }
   str3[k] = '\0';
}
```

Some sample input and output sessions are given below:

(1) Test Case 1:
```
Enter str1:
abcde
Enter str2:
dec
strIntersect(): cde
```

(2) Test Case 2:
```
Enter str1:
abcdefghijk
Enter str2:
akdhf
strIntersect(): adfhk
```

(3) Test Case 3:
```
Enter str1:
abc
Enter str2:
def
strIntersect(): null string
```