**strNRChr**

Write a C function strNRChr() that locates the **last n$^{th}$ occurrence** of ***ch*** in the string pointed to by ***str***. The function returns a pointer to the character, or a **NULL** pointer if ***ch*** does not occur in the string. For example, if ***str*** = "abacadae", ***ch*** = 'a' and ***n*** = 2, then the function returns the address of the substring "adae" within "abacadae". In this function, there is no need to check input error.

A sample program template for testing the function is given below:

```
#include <stdio.h>
#include <string.h>
char *strNRChr(char *str, int n, char ch);
int main()
{
   char str[80], ch, dummy;
   char *temp=NULL;
   int n;

   printf("Enter a string: \n");
   scanf("%s",str);
   scanf("%c",&dummy);
   printf("Enter a char: \n");
   scanf("%c",&ch);
   printf("Enter the occurrence: \n");
   scanf("%d", &n);
   temp = strNRChr(str, n, ch);
   if (temp!=NULL)
      printf("strNRChr(): %s\n", temp);
   else
      printf("strNRChr(): null string\n");
   return 0;
}
char *strNRChr(char *str, int n, char ch)
{
   /* Write your code here */
}
```

Some test input and output sessions are given below:

(1) Test Case 1

```
Enter a string:
abacadae
Enter a char:
z
Enter the occurrence:
1
strNRChr(): null string
```

(2) Test Case 2

```
Enter a string:
abacadae
Enter a char:
```

```
a
Enter the occurrence:
1
strNRChr(): ae
```

(3) Test Case 3

```
Enter a string:
abacadae
Enter a char:
a
Enter the occurrence:
2
strNRChr(): adae
```

(4) Test Case 4

```
Enter a string:
abacadae
Enter a char:
a
Enter the occurrence:
3
strNRChr(): acadae
```

```c
#include <stdio.h>
#include <string.h>
char *strNRChr(char *str, int n, char ch);
int main()
{
  char str[80], ch, dummy;
  char *temp=NULL;
  int n;

  printf("Enter a string: \n");
  scanf("%s",str);
  scanf("%c",&dummy);
  printf("Enter a char: \n");
  scanf("%c",&ch);
  printf("Enter the occurrence: \n");
  scanf("%d", &n);
  temp = strNRChr(str, n, ch);
  if (temp!=NULL)
    printf("strNRChr(): %s\n", temp);
  else
    printf("strNRChr(): null string\n");
  return 0;
}
char *strNRChr(char *str, int n, char ch)
{
   int len;
   int i;
   char *name;
   int k =0;
   int j;
   int count = 0;
   int a;

   len = strlen(str);

   for(i=0;i<len;i++)
   {
      if(ch == str[len-1-i])
      {
         count+=1;
         if(count == n)
         {
            a=len-1-i;
            for(j=a;j<len;j++)
            {
               name[k] = str[j];
               k++;
            }
            return name;
         }
      }
   }
   return NULL;

}
```

retrun (str+i)