largestCharStr

Write a function largestCharStr() that takes in an array of strings ***str*** (all characters are in lower letter cases) with ***size*** > 0 and an array ***a*** as parameters, finds the larget character (based on ASCII values) for each string in the array of strings, and stores them into the array ***a*** which is then returned to the calling function via call by reference. For example, if ***size*** is 5 and the array of strings ***str*** is {"peter", "john", "mary", "jane", "kenny"}, then the characters {'t', 'o', 'y', 'n', 'y'} will be stored in the array ***a*** after executing the function.

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

```
#include <stdio.h>
#include <string.h>
#define N 20
void largeCharStr(char str[N][20], char a[N], int size);
int main()
{
   char str[N][20],dummy;
   char a[N],i,j,size;

   printf("Enter number of strings: \n");
   scanf("%d", &size);
   scanf("%c", &dummy);
   for (i=0;i<size;i++){
      printf("Enter string %d: \n", i+1);
      scanf("%s",str[i]);
   }
   largeCharStr(str,a,size);
   printf("largeCharStr(): \n");
   for (i=0;i<size;i++) {
      printf("String %d: ",i+1);
      printf("%c\n",a[i]);
   }
   return 0;
}
void largeCharStr(char str[N][20], char a[N], int size)
{
   /* Write your code here */
}
```

Some sample input and output sessions are given below:

(1) Test Case 1:

```
Enter number of strings:
4
Enter string 1:
kenny
Enter string 2:
mary
Enter string 3:
peter
Enter string 4:
sun
String 1: y
String 2: y
String 3: t
String 4: u
```

(2) Test Case 2:

```
Enter number of strings:
5
Enter string 1:
kenny
Enter string 2:
mary
Enter string 3:
peter
Enter string 4:
sun
Enter string 5:
jane
String 1: y
String 2: y
String 3: t
String 4: u
String 5: n
```