findMinMaxStr

Write a C function that reads in words separated by space, finds the first and last words according to ascending alphabetical order (based on ASCII values), and returns them to the calling function through the string parameters first and last. The calling function will then print the first and last strings on the screen. The function prototype is given as follows:

```
void findMinMaxStr(char word[][40], char *first, char *last,
                   int size);
```

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

```
#include <stdio.h>
#include <string.h>
#define SIZE 10
void findMinMaxStr(char word[][40], char *first, char *last, int size);
int main()
{
  char word[SIZE][40];
  char first[40], last[40];
  int i, size;

  printf("Enter size: \n");
  scanf("%d", &size);
  printf("Enter %d words: \n", size);
  for (i=0; i<size; i++)
    scanf("%s", word[i]);
  findMinMaxStr(word, first, last, size);
  printf("First word = %s, Last word = %s\n", first, last);
  return 0;
}
void findMinMaxStr(char word[][40], char *first, char *last, int size)
{
    /* Write your code here */
}
```

Some sample input and output sessions are given below:

(1) Test Case 1:

```
Enter size:
4
Enter 4 words:
Peter Paul John Mary
First word = John, Last word = Peter
```

(2) Test Case 2:

```
Enter size:
1
Enter 1 words:
Peter
First word = Peter, Last word = Peter
```

```
void findMinMaxStr(char word[][40],
char *first, char *last, int size)
{
   int i;
   strcpy(first,word[0]);
   strcpy(last,word[0]);
   for(i=1;i<size;i++)
   {
      if(strcmp(word[i],first)<0)
          strcpy(first,word[i]);
      if(strcmp(word[i],last)>0)
          strcpy(last,word[i]);
   }
}
```

(3) Test Case 3:
Enter size:
2
Enter 2 words:
Peter Mary
First word = Mary, Last word = Peter