

```
1 // Created by hfwei on 2022/11/25.
2
3 #include <stdio.h>
4 #include <string.h>
5
6 #define LEN 10
7
8 void Swap(char **left, char **right);
9 void PrintStrs(const char *str[], int len);
10 void SelectionSort(char *str[], int len);
11
12 int main() {
13     const char *musicians[LEN] = {
14         "Luo Dayou",
15         "Cui Jian",
16         "Dou Wei",
17         "Zhang Chu",
18         "Li Zhi",
19         "Wan Qing",
20         "WuTiaoRen",
21         "ZuoXiao",
22         "He Mage",
23         "He Yong",
24     };
25
26     PrintStrs(musicians, LEN);
27     SelectionSort(musicians, LEN);
28     PrintStrs(musicians, LEN);
29 }
30
31 void PrintStrs(const char *str[], int len) {
32     printf("\n");
33     for (int i = 0; i < len; i++) {
34         printf("%s\n", str[i]);
35     }
36     printf("\n");
37 }
38
39 // arr: the (copy of the) address of the first element of the `numbers
   ` array
40 void SelectionSort(char *str[], int len) {
41     for (int i = 0; i < len; i++) {
42         // find the minimum of musicians[i .. len - 1]
43         const char *min = str[i];
44         int min_index = i;
45
46         for (int j = i + 1; j < len; j++) {
47             if (strcmp(min, str[j]) > 0) {
48                 min = str[j];
49                 min_index = j;
50             }
51         }
52     }
```

```
53     // swap str[i] and str[min_index]
54     Swap(str + i, str + min_index);
55 }
56 }
57
58 void Swap(char **left, char **right) {
59     char *temp = *left;
60     *left = *right;
61     *right = temp;
62 }
```

```

1 //
2 // Created by hfwei on 2022/11/25.
3 //
4
5 #include <stdio.h>
6
7 /**
8  * @brief Compares the string pointed to by s1 to the string pointed
9  *         to by s2
10 * @param s1 the first string
11 * @param s2 the second string
12 * @return > 0 if s1 > s2; 0 if s1 = s2; < 0 if s1 < s2.
13 */
14 int StrCmp(const char *s1, const char *s2);
15
16 /**
17 * @brief Compare no more than n characters
18 *         (characters that follow '\0' are not compared)
19 *         from the array pointed to by s1 to the array pointed to by
20 *         s2.
21 * @param s1
22 * @param s2
23 * @param n
24 * @return
25 */
26 int StrNCmp(const char *s1, const char *s2, int n);
27
28 int main() {
29     const char *str1 = "hi, C";
30     const char *str2 = "hi, c";
31
32     printf("StrCmp(\"%s\", \"%s\") = %d\n",
33           str1, str2, StrCmp(str1, str2));
34     printf("StrCmpStd(\"%s\", \"%s\") = %d\n",
35           str1, str2, StrCmpStd(str1, str2));
36
37     int n = 2;
38     printf("StrNCmp(\"%s\", \"%s\", %d) = %d\n",
39           str1, str2, n, StrNCmp(str1, str2, n));
40
41     return 0;
42 }
43
44 int StrCmp(const char *s1, const char *s2) {
45     while (*s1 == *s2 && (*s1 != '\0' && *s2 != '\0')) {
46         s1++;
47         s2++;
48     }
49
50     // four cases here
51     if (*s1 == '\0' && *s2 == '\0') {

```

```
52     return 0;
53 }
54
55 return (*(const unsigned char *) s1)
56         < (*(const unsigned char *) s2) ? -1 : 1;
57 }
58
59 int StrCmpStd(const char *s1, const char *s2) {
60     for (; *s1 == *s2; s1++, s2++) {
61         if (*s1 == '\0') {
62             return 0;
63         }
64     }
65
66     return (*(const unsigned char *) s1)
67             < (*(const unsigned char *) s2) ? -1 : 1;
68 }
69
70 int StrNCmp(const char *s1, const char *s2, int n) {
71     for (int i = 0; i < n; i++) {
72         if (*s1 != *s2) {
73             return (*(const unsigned char *) s1)
74                     < (*(const unsigned char *) s2) ? -1 : 1;
75         } else { // *s1 = *s2
76             if (*s1 == '\0') {
77                 return 0;
78             } else {
79                 s1++;
80                 s2++;
81             }
82         }
83     }
84
85     return 0;
86 }
87
88 int StrNCmpStd(const char *s1, const char *s2, int n) {
89     for (; 0 < n; n--, s1++, s2++) {
90         if (*s1 != *s2) {
91             return (*(const unsigned char *) s1)
92                     < (*(const unsigned char *) s2) ? -1 : 1;
93         } else if (*s1 == '\0') {
94             return 0;
95         }
96     }
97
98     return 0;
99 }
```

```

1 //
2 // Created by hfwei on 2022/11/25.
3 //
4
5 #include <string.h>
6 #include <stdio.h>
7
8 /**
9  * @brief Copy the string pointed to by src (including the terminating
10  *    null character)
11  *    into the array pointed to by dest.
12  *    If copying takes place between objects that overlap,
13  *    the behavior is undefined.
14  *
15  *    We assume that there is enough room in dest for storing src.
16  *    Otherwise, it is an undefined behavior.
17  * @param dest
18  * @param src
19  */
20 void StrCpy(char *dest, const char *src);
21 void StrCpy1(char *dest, const char *src);
22 void StrCpy2(char *dest, const char *src);
23 void StrCpy3(char *dest, const char *src);
24 void StrCpy4(char *dest, const char *src);
25 void StrCpy5(char *dest, const char *src);
26 char *StrCpyStd(char *dest, const char *src);
27
28 int main() {
29     const char *src = "Hello World";
30     char dest[strlen(src) + 1];
31
32     // StrCpy(dest, src);
33     // printf("dest = %s\n", dest);
34
35     strcpy_s(dest, sizeof dest / sizeof dest[0], src);
36     printf("dest = %s\n", dest);
37
38     return 0;
39 }
40
41 void StrCpy(char *dest, const char *src) {
42     int i = 0;
43     while (src[i] != '\0') {
44         dest[i] = src[i];
45         i++;
46     }
47
48     dest[i] = '\0';
49 }
50
51 void StrCpy1(char *dest, const char *src) {
52     int i = 0;

```

```
53 while ((dest[i] = src[i]) != '\0') {
54     i++;
55 }
56 }
57
58 void StrCpy2(char *dest, const char *src) {
59     int i = 0;
60     while ((*dest + i) = *(src + i)) != '\0') {
61         i++;
62     }
63 }
64
65 void StrCpy3(char *dest, const char *src) {
66     while ((*dest = *src) != '\0') {
67         src++;
68         dest++;
69     }
70 }
71
72 void StrCpy4(char *dest, const char *src) {
73     while ((*dest++ = *src++) != '\0');
74 }
75
76 // NOT recommended!
77 void StrCpy5(char *dest, const char *src) {
78     while ((*dest++ = *src++));
79 }
80
81 char *StrCpyStd(char *dest, const char *src) {
82     for (char *s = dest; (*s++ = *src++) != '\0');
83     return dest;
84 }
```

```
1 //
2 // Created by hfwei on 2022/11/25.
3 //
4
5 #include <stdio.h>
6
7 int StrLen(const char *str);
8 size_t StrLenStd(const char *str);
9
10 int main() {
11     char msg[20] = "Hello World!";
12     msg[0] = 'N';
13     printf("%s\n", msg);
14
15     // create an anonymous array of `char`
16     // that may be stored in read-only memory
17     char *ptr_msg = "Hello World!";
18     // undefined behavior
19     // interpreted by signal 11: SIGSEGV
20     // SIG: Signal; SEGV: segmentation violation
21     *ptr_msg = 'N';
22     // ptr_msg[0] = 'N';
23     printf("%s\n", msg);
24
25     printf("StrLen(%s) = %d\n", msg, StrLen(msg));
26     printf("StrLenStd(%s) = %d\n", msg, StrLen(msg));
27
28     return 0;
29 }
30
31 int StrLen(const char *str) {
32     int len = 0;
33     while (str[len] != '\0') {
34         len++;
35     }
36     // printf("%d", '\0');
37
38     return len;
39 }
40
41 size_t StrLenStd(const char *str) {
42     const char *sc;
43     for (sc = str; *sc != '\0'; sc++);
44
45     return (sc - str);
46 }
```