

第 12 次作业

第一题

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #pragma warning(disable:4996)
5  void sort(char* p[20],int n);
6  int main() {
7      int n;
8      printf("input N: ");
9      scanf("%d", &n);
10     char* p[20];
11     for (int i = 0;i < n;i++) {
12         *(p + i) = (char*)malloc(sizeof(char*));
13         scanf("%s", *(p+i));
14     }
15     sort(p,n);
16     printf("output: \n");
17     for (int i = 0;i < n;i++) {
18         printf("%s\n", *(p + i));
19     }
20 }
21
22 void sort(char* p[20],int n)
23 {
24     for (int i = 0;i < n;i++) {
25         char* min = *(p + i);
26         int k = i;
27         for (int j = i;j < n;j++) {
28             if (strcmp(min, *(p + j)) > 0) {
29                 k = j;
30                 min = *(p + j);
31             }
32         }
33         char* temp = *(p + i);
```

```
34         *(p + i) = min;
35         *(p + k) = temp;
36     }
37 }
38
```

```
input N: 3
dsj kf
rew iuqr
zcv
output:
dsj kf
rew iuqr
zcv
```

思路

首先输入 n ，定义一个 `char *`数组，输入 n 个数组，接下来对数组中的元素排序，输出。

第二题

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #pragma warning(disable:4996)
4  int plus(int i, int n);
5  int main() {
6      int a[10][10];
7      int s[10]={0};
8      for (int i = 0;i < 10;i++) {
9          for (int j = 0;j < 10;j++) {
10             *(* (a + i) + j) = i * 10 + j;
11         }
12     }
```

```

13     for (int n = 0;n < 10;n++) {
14         for (int i = 1;i <= 10;i++) {
15             *(s+n) += (*(a + i - 1) + plus(i - 1, n));
16         }
17     }
18     for (int i = 0;i < 10;i++) {
19         printf("第%d 对角线:  %d  \n", i, *(s+i));
20     }
21 }
22
23 int plus(int i, int n)
24 {
25     if ((i + n) <= 9)return i + n;
26     else return (i + n) % 10;
27 }
28

```

```

第0对角线:  495
第1对角线:  495
第2对角线:  495
第3对角线:  495
第4对角线:  495
第5对角线:  495
第6对角线:  495
第7对角线:  495
第8对角线:  495
第9对角线:  495

```

代码思路

首先初始化矩阵 a ，然后定义 $plus$ 函数用于计算 \oplus ，循环计算 n 条对角线的和。

第三题

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #pragma warning(disable:4996)
5  int main() {
6      char start, end;
7      int flag = 0;
8      char* s=(char*)malloc(140*sizeof(char));
9      char* target = (char*)malloc(140 * sizeof(char));
10
11     memset(target, 0, 140 * sizeof(char));
12     printf("start: ");
13     scanf("%c", &start);
14     getchar();
15     printf("end: ");
16     scanf("%c", &end);
17     getchar();
18     printf("input:  \n");
19     gets_s(s, 140);
20     printf("output:\n");
21
22     int index=0;
23     for (int i = 0;i < strlen(s);i++) {
24         if ( * (s + i) == start) {
25             flag = 1;
26             continue;
27         }
28         if (flag == 1) {
29             if (*(s + i) == start) {
30                 memset(target, 0, 140 * sizeof(char));
31                 index = 0;
32             }
33             else if (*(s + i) == end) {
34                 flag = 0;
35                 printf("%s\n", target);
36                 memset(target, 0, 140 * sizeof(char));
37                 index = 0;
38             }
39             else {
40                 *(target + index) = *(s + i);
```

```
41         index++;
42     }
43 }
44 }
45 }
```

```
start: [
end: ]
input:
Emmm... You use [Grin] instead of [Smile] when you are really happy in wechat.
output:
Grin
Smile
```

```
start: *
end: #
input:
Time for lunch. *greedy# Hope a big meal
output:
greedy
```

思路

首先输入 start, end 与 s, 接着遍历 s, 如果碰到 start 则 flag 置为 1。flag 为 1 的情况下, 不断复制 s 中的字符到 target 中直到遇到 end, 此时 flag 置为 0, 打印此时的 target, 然后将其清空, 另外如果再次碰到了 start 则也将 target 清空, index 置为 0。