

計算機程式語言

教課教授：謝東儒

助教：蔡詠聿、吳品頤

Chapter 17_project 3

Modify the `inventory2.c` program of Section 17.5 by adding an `e(erase)` command that allows the user to remove a part from the database.

請確實實作link list

本次作業只會有一個.c檔，不會有其他的.h或.c檔

Solution

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <ctype.h>
4
5  #define NAME_LEN 25
6
7  struct part {
8      int number;
9      char name[NAME_LEN+1];
10     int on_hand;
11     struct part *next;
12 };
13
14 struct part *inventory = NULL;
15
16 struct part *find_part(int number);
17 void insert(void);
18 void search(void);
19 void update(void);
20 void print(void);
21 void erase(void);
22 int read_line(char str[], int n);
23
```

Solution

```
24 int main(void)
25 {
26     char code;
27
28     for (;;) {
29         printf("Enter operation code: ");
30         scanf(" %c", &code);
31         while (getchar() != '\n')
32             ;
33         switch (code) {
34             case 'i': insert();
35                     break;
36             case 'e': erase();
37                     break;
38             case 's': search();
39                     break;
40             case 'u': update();
41                     break;
42             case 'p': print();
43                     break;
44             case 'q': return 0;
45             default: printf("Illegal code\n");
46         }
47         printf("\n");
48     }
49 }
50
```



Solution

```
51 struct part *find_part(int number)
52 {
53     struct part *p;
54
55     for (p = inventory;
56         p != NULL && number > p->number;
57         p = p->next)
58         ;
59     if (p != NULL && number == p->number)
60         return p;
61     return NULL;
62 }
63
```

Solution

```
64 void insert(void)
65 {
66     struct part *cur, *prev, *new_node;
67
68     new_node = malloc(sizeof(struct part));
69     if (new_node == NULL) {
70         printf("Database is full; can't add more parts.\n");
71         return;
72     }
73
74     printf("Enter part number: ");
75     scanf("%d", &new_node->number);
76
77     for (cur = inventory, prev = NULL;
78         cur != NULL && new_node->number > cur->number;
79         prev = cur, cur = cur->next)
80         ;
81     if (cur != NULL && new_node->number == cur->number) {
82         printf("Part already exists.\n");
83         free(new_node);
84         return;
85     }
86
87     printf("Enter part name: ");
88     read_line(new_node->name, NAME_LEN);
89     printf("Enter quantity on hand: ");
90     scanf("%d", &new_node->on_hand);
91     new_node->next = cur;
92     if (prev == NULL)
93         inventory = new_node;
94     else
95         prev->next = new_node;
96 }
97
```

Solution

```
98 void erase(void)
99 {
100     struct part **pp = ;
101     struct part *temp;
102     int n;
103     printf("Enter part number: ");
104     scanf("%d", &n);
105
106     
107
108
109
110
111
112
113
114
115     printf("Part number %d not found in database\n", n);
116     return;
117 }
118
```

Solution

```
119 void search(void)
120 {
121     int number;
122     struct part *p;
123
124     printf("Enter part number: ");
125     scanf("%d", &number);
126     p = find_part(number);
127     if (p != NULL) {
128         printf("Part name: %s\n", p->name);
129         printf("Quantity on hand: %d\n", p->on_hand);
130     } else
131         printf("Part not found.\n");
132 }
133
```


Solution

```
134 void update(void)
135 {
136     int number, change;
137     struct part *p;
138
139     printf("Enter part number: ");
140     scanf("%d", &number);
141     p = find_part(number);
142     if (p != NULL) {
143         printf("Enter change in quantity on hand: ");
144         scanf("%d", &change);
145         p->on_hand = change;
146     } else
147         printf("Part not found.\n");
148 }
149
```

Solution

```
150 void print(void)
151 {
152     struct part *p;
153
154     printf("Part Number    Part Name                "
155           "Quantity on Hand\n");
156     for (p = inventory; p != NULL; p = p->next)
157         printf("%7d          %-25s%11d\n", p->number, p->name,
158               p->on_hand);
159 }
160
161 int read_line(char str[], int n)
162 {
163     int ch, i = 0;
164
165     while (isspace(ch = getchar()))
166         ;
167     while (ch != '\n' && ch != EOF) {
168         if (i < n)
169             str[i++] = ch;
170         ch = getchar();
171     }
172     str[i] = '\0';
173     return i;
174 }
```

Example

```
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
Enter operation code: i
Enter part number: 3
Enter part name: hot dog
Enter quantity on hand: 20

Enter operation code: i
Enter part number: 4
Enter part name: soup
Enter quantity on hand: 50

Enter operation code: p
Part Number  Part Name          Quantity on Hand
      3      hot dog             20
      4      soup              50

Enter operation code: u
Enter part number: 4
Enter change in quantity on hand: 80

Enter operation code: e
Enter part number: 3

Enter operation code: p
Part Number  Part Name          Quantity on Hand
      4      soup             80

Enter operation code: s
Enter part number: 4
Part name: soup
Quantity on hand: 80

Enter operation code: w
Illegal code

Enter operation code: q
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$
```

Chapter 17_project 5

Write a program that sorts a series of words entered by the user:

Assume that each word is no more than 20 characters long. Stop reading when the user enters an empty word (i.e., presses Enter without entering a word). Store each word in a dynamically allocated string, using an array of pointers to keep track of the strings, as in the remind2.c program (Section 17.2). After all words have been read, sort the array (using any sorting technique) and then use a loop to print the words in sorted order. Hint: Use the read_line function to read each word. as in remind2 . c.

請不要使用**qsort()**

Solution

```
1 // sort_words
2
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include <string.h>
6
7 #define MAX_WORDS 50
8 #define WORD_LEN 20
9
10 int read_line(char str[], int n);
11 void quicksort(char **low, char **high);
12 char **split(char **low, char **high);
13
14 int main(void){
15
16     char *words[MAX_WORDS], word[WORD_LEN+1];
17     int i, num_words = 0;
18
19     for(;;){
20         if(num_words == MAX_WORDS){
21             printf(" -- No space left --\n");
22             break;
23         }
24
25         printf("Enter word : ");
26         read_line(word, WORD_LEN);
27         if(strlen(word) == 0)
28             break;
29
30         words[num_words] = (char *)malloc(strlen(word) + 1);
31         if(words[num_words] == NULL){
32             printf(" -- No space left --\n");
33             break;
34         }
```

Solution

```
35     strcpy(words[num_words], word);
36     num_words++;
37 }
38
39 quicksort(words, words + num_words - 1);
40
41 printf("\nIn sorted order : ");
42 for(i=0; i < num_words; i++){
43     printf(" %s", words[i]);
44 }
45 printf("\n");
46
47 return 0;
48 }
49
50
51 int read_line(char str, int n){
52     int ch, i=0;
53
54     while((ch = getchar()) != '\n'){
55         if(i < n){
56             str[i++] = ch;
57         }
58     }
59
60     str[i] = '\0';
61
62     return i;
63 }
64
65
```

Solution

```
66 void quicksort(char **low, char **high)
67 {
68     char **middle;
69
70     if(low >= high) return;
71     middle = split(low, high);
72     quicksort(low, middle - 1);
73     quicksort(middle + 1, high);
74 }
75
76 char **split(char **low, char **high)
77 {
78     char *part_element = *low;
79
80     for(;;){
81         while(low < high && strcmp(part_element, *high) <= 0){
82             high--;
83         }
84         if(low >= high) break;
85         *low++ = *high;
86
87         while(low < high && strcmp(*low, part_element) <= 0){
88             low++;
89         }
90         if(low >= high) break;
91         *high-- = *low;
92     }
93
94     *high = part_element;
95     return high;
96 }
```

Example

```
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
Enter a word: foo
Enter a word: bar
Enter a word: baz
Enter a word: quux
Enter a word:

In sorted order: bar baz foo quux
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$
```


Chapter 17_project 6

Modify Programming Project 5 so that it uses qsort to sort the array of pointers.

Solution

```
1 // sort_words2
2
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include <string.h>
6
7 #define MAX_WORDS 50
8 #define WORD_LEN 20
9
10 int read_line(char str[], int n);
11 int compare_strings(const void *p, const void *q);
12
13 int main(void){
14
15     char *words[MAX_WORDS], word[WORD_LEN+1];
16     int i,num_words = 0;
17
18     for(;;){
19         if(num_words == MAX_WORDS){
20             printf(" -- No space left --\n");
21             break;
22         }
23
24         printf("Enter word : ");
25         read_line(word, WORD_LEN);
26         if(strlen(word) == 0)
27             break;
28
29         words[num_words] = (char *)malloc(strlen(word) + 1);
30         if(words[num_words] == NULL){
31             printf(" -- No space left --\n");
32             break;
33         }
34     }
```

Solution

```
34
35     strcpy(words[num_words], word);
36     num_words++;
37 }
38
39 qsort(words, num_words, sizeof(char *), compare_strings);
40
41 printf("\nIn sorted order : ");
42 for(i=0; i < num_words; i++){
43     printf(" %s", words[i]);
44 }
45 printf("\n");
46
47 return 0;
48 }
49
50 int read_line(char str[], int n){
51     int ch, i=0;
52
53     while((ch = getchar()) != '\n'){
54         if(i < n){
55             str[i++] = ch;
56         }
57     }
58
59     str[i] = '\0';
60
61     return i;
62 }
63
64
65 int compare_strings(const void *p, const void *q){
66     return strcmp(*(char **)p, *(char **)q);
67 }
```

Example

```
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
Enter a word: foo
Enter a word: bar
Enter a word: baz
Enter a word: quux
Enter a word:

In sorted order: bar baz foo quux
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$
```

Chapter 17_project 7

(C99) Modify the remind2.c program of Section 17.2 so that each element of the reminders array is a pointer to a vstring structure (see Section 17.9) rather than a pointer to an ordinary string.

Solution

```
1 // remind2
2
3 #include <stdio.h>
4 #include <stdlib.h>
5 #include <string.h>
6
7 #define MAX_REMIND 50 /* maximum number of reminders */
8 #define MSG_LEN 60 /* max length of reminder message */
9
10 struct vstring {
11     int len;
12     char chars[];
13 };
14
15 int read_line(char str[], int n);
16
17 int main(void){
18
19     struct vstring *reminders[MAX_REMIND];
20     char day_str[3], msg_str[MSG_LEN + 1];
21     int day, i, j, num_remind = 0;
22
23     for(;;){
24         if(num_remind == MAX_REMIND){
25             printf("-- No space left --\n");
26             break;
27         }
28
29         printf("Enter day and reminder : ");
30         scanf("%2d", &day);
31         if(day == 0){
32             break;
33         }
34         sprintf(day_str, "%2d", day);
35         read_line(msg_str, MSG_LEN);
36
37         for(i = 0; i < num_remind; i++){
38             if(strcmp(day_str, reminders[i]->chars) < 0){
39                 break;
40             }
41         }
```

Solution

```
42 for(j = num_remind; j > i; j--){
43     reminders[j] = reminders[j-1];
44 }
45
46 reminders[i] = (vstring *)malloc(sizeof(struct vstring) + 2 + strlen(msg_str));
47 if(reminders[i] == NULL){
48     printf("-- No space left --\n");
49     break;
50 }
51
52 reminders[i]->len = 2 + strlen(msg_str);
53 memcpy(reminders[i]->chars, day_str, 2);
54 memcpy(reminders[i]->chars + 2, msg_str, strlen(msg_str));
55
56 num_remind++;
57 }
58
59 printf("\nDay Reminder\n");
60 for(i = 0; i < num_remind; i++){
61     printf(" %s\n", reminders[i]->len, reminders[i]->chars);
62 }
63
64 return 0;
65 }
66
67 int read_line(char str[], int n){
68     int ch, i=0;
69
70     while((ch = getchar()) != '\n'){
71         if(i < n){
72             str[i++] = ch;
73         }
74     }
75
76     str[i] = '\0';
77
78     return i;
79 }
80 }
```

Example

```
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
Enter day and reminder: 5 dating
Enter day and reminder: 12 meeting
Enter day and reminder: 31 ready for new year
Enter day and reminder: 0 0

Day Reminder
 5 dating
12 meeting
31 ready for new year
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$
```