計算機程式語言

助教：蔡詠聿、吳品頤

Chapter 16_project 4

Write program that asks the user to enter an international dialing code and then looks it up in the country_codes array (see Section 16.3). If it finds the code, the program should display the name of the corresponding country; if not, the program should print an error message.

Chapter 16_project 4

Modify the inventory.c program of Section 16.3 by adding a price member to the part structure. The insert function should ask the user for the price of a new item. The search and print functions should display the price. Add a new command that allows the user to change the price of a part.

```
Enter operation code : i
Enter part number :2
Enter part name : milk tea
Enter price : 55
Enter quantity on hand : 20

Enter operation code : p
Part number      Part Name              Price          Quantity on Hand
      2            milk tea                $55.00          20

Enter operation code : s
Enter part number : 2
Part name : milk tea
Price : $55.00
Quantity on hand : 20

Enter operation code : c
Enter part number : 2
Enter new price : 60

Enter operation code : u
Enter part number : 2
Enter change in quantity on hand : 100

Enter operation code : q

--------------------------------
Process exited after 74.75 seconds with return value 0
請按任意鍵繼續 . . .
```

Solution

```c
//  inventory2

#include <stdio.h>
#include "readline/readline.h"

#define NAME_LEN 25
#define MAX_PARTS 100

struct part{
    int number;
    char name[NAME_LEN + 1];
    float price;
    int on_hand;
} inventory[MAX_PARTS];

int num_parts = 0;  /*  number of parts currently stored    */

int find_part(int number);
void insert(void);
void search(void);
void change(void);
void update(void);
void print(void);
```

Solution

```c

int main(void){

    char code;

    for(;;){
        printf("Enter operation code : ");
        scanf(" %c", &code);
        while(getchar() != '\n')    ;   /*  skips to end of line    */

        switch(code){
            case 'i':
                insert();
                break;
            case 's':
                search();
                break;
            case 'c':
                change();
                break;
            case 'u':
                update();
                break;
            case 'p':
                print();
                break;
            case 'q':
                return 0;
            default:
                printf("illegal code\n");
        }
        printf("\n");
    }

    return 0;
}

```

Solution

```
60
61  int find_part(int number){
62
63      int i;
64
65      for(i = 0; i < num_parts; i++){
66          if(inventory[i].number == number){
67              return i;
68          }
69      }
70
71      return -1;
72  }
73
74  void insert(void){
75      int part_number;
76
77      if(num_parts == MAX_PARTS){
78          printf("Database is full; can't add more parts.\n");
79          return;
80      }
81
82      printf("Enter part number :");
83      scanf("%d", &part_number);
84      if(find_part(part_number) >= 0){
85          printf("Part already exists.\n");
86          return;
87      }
88
89      inventory[num_parts].number = part_number;
90      printf("Enter part name : ");
91      read_line(inventory[num_parts].name, NAME_LEN);
92      printf("Enter price : ");
93      scanf("%f", &inventory[num_parts].price);
94      printf("Enter quantity on hand : ");
95      scanf("%d", &inventory[num_parts].on_hand);
96      num_parts++;
97  }
```

Solution

```
 98
 99  void search(void){
100      int i, number;
101
102      printf("Enter part number : ");
103      scanf("%d", &number);
104      i = find_part(number);
105      if(i >= 0){
106          printf("Part name : %s\n", inventory[i].name);
107          printf("Price : $%.2f\n", inventory[i].price);
108          printf("Quantity on hand : %d\n", inventory[i].on_hand);
109      }else{
110          printf("Part not found.\n");
111      }
112  }
113
114  void change(void){
115      int i, number;
116      float new_price;
117
118      printf("Enter part number : ");
119      scanf("%d", &number);
120      i = find_part(number);
121      if(i >= 0){
122          printf("Enter new price : ");
123          scanf("%f", &new_price);
124      }else{
125          printf("Part not found.\n");
126      }
127  }
128
```

Solution

```
128
129 void update(void){
130     int i, number, change;
131
132     printf("Enter part number : ");
133     scanf("%d", &number);
134     i = find_part(number);
135     if(i >= 0){
136         printf("Enter change in quantity on hand : ");
137         scanf("%d", &change);
138         inventory[i].on_hand += change;
139     }else{
140         printf("Part not found.\n");
141     }
142 }
143
144 void print(void){
145     int i;
146
147     printf("Part number \t Part Name \t\t "
148                         "Price \t\t Quantity on Hand\n");
149     for(i = 0; i < num_parts; i++){
150         printf("%7d             %-25s $%2.2f   %9d\n", inventory[i].number,
151                             inventory[i].name, inventory[i].price, inventory[i].on_hand);
152     }
153 }
```

Solution

readline.h

```h
//  readline.h

#ifndef READLINE_H
#define READLINE_H

int read_line(char str[], int n);

#endif
```

readline.c

```c
//    readline.c

#include <ctype.h>
#include <stdio.h>
#include "readline.h"

int read_line(char str[], int n){

  int ch, i = 0;

  while (isspace(ch = getchar()))   ;

  while (ch != '\n' && ch != EOF) {
    if(i < n){
      str[i++] = ch;
    }
    ch = getchar();
  }
  str[i] = '\0';
  return i;
}
```

Chapter 16_project 5

Modify Programming Project 8 from Chapter 5 so that the times are stored in a single array. The elements of the array will be structures, each containing a departure time and the corresponding arrival time. (Each time will be an integer, representing the number of minutes since midnight.) The program will use a loop to search the array for the departure time closest to the time entered by the user.

```
Enter a 24-hour time : 13:15
Closest departure time is 12:47 p.m. ,arriving at 3:00 p.m.

--------------------------------
Process exited after 47.17 seconds with return value 0
請按任意鍵繼續 . . .
```

Solution

```c
// flight

#include <stdio.h>

#define HOURS_PER_HALF_DAY 12
#define MINUTES_PER_HOUR 60
#define MINUTES_PER_HALF_DAY  (HOURS_PER_HALF_DAY * MINUTES_PER_HOUR)

#define SIZE ((int)(sizeof(flights) / sizeof(flights[0])))

struct flight {
    int departure, arrival;
};

void put_time(int time);

int main(void){

    struct flight flights[]{
        {480, 616}, {583, 712}, {679, 811}, {767, 900},
        {840, 968}, {945, 1075}, {1140, 1280}, {1305, 1438}
    };

    int hours, minutes, time, closest;

    printf("Enter a 24-hour time : ");
    scanf("%d:%d", &hours, &minutes);
    time = hours * MINUTES_PER_HOUR + minutes;

```

Solution

```
29
30      if(time <= flights[0].departure){
31          closest = 0;
32      }else if(time > flights[SIZE - 1].departure){
33          closest = SIZE - 1;
34      }else{
35          closest = 0;
36          while(time > flights[closest + 1].departure){
37              closest++;
38          }
39          if((flights[closest + 1].departure - time) < (time - flights[closest].departure)){
40              closest++;
41          }
42      }
43
44      printf("Closest departure time is ");
45      put_time(flights[closest].[redacted]);
46      printf(" ,arriving at ");
47      put_time(flights[closest].[redacted]);
48      printf("\n");
49
50      return 0;
51  }
52
```

Solution

```
52
53  void put_time(int time){
54
55      int hour = time / MINUTES_PER_HOUR;
56
57      if(hour == 0){
58          hour = HOURS_PER_HALF_DAY;
59      }else if(hour > HOURS_PER_HALF_DAY){
60          hour -= HOURS_PER_HALF_DAY;
61      }
62      printf("%d:%.2d ", hour, time % MINUTES_PER_HOUR);
63
64      if(time <                    ){
65          printf("a.m.");
66      }else{
67          printf("p.m.");
68      }
69  }
```

Chapter 16_problem 6

Modify Programming Project 9 from Chapter 5 so that each date entered by the user is stored in a date structure (see Exercise 5). Incorporate the compare_dates function of Exercise 5 into your program.

```
Enter first date (mm/dd/yy) : 3/6/08
Enter second date (mm/dd/yy) : 5/17/07
5/17/07 is earlier then 3/6/08

--------------------------------
Process exited after 16.18 seconds with return value 0
請按任意鍵繼續 . . .
```

Solution

```
1   // compare_dates
2
3   #include <stdio.h>
4
5   /*  Note: Program assumes years are in the same century */
6
7   struct date {
8       int month, day, year;
9   };
10
11  int compare_dates(struct [redacted] d1, struct [redacted] d2);
12  void put_date(struct date d);
13
14  int main(void){
15
16      struct [redacted] d1, d2;
17
18      printf("Enter first date (mm/dd/yy) : ");
19      scanf("%d/%d/%d", &d1.month, &d1.day, &d1.year);
20      printf("Enter second date (mm/dd/yy) : ");
21      scanf("%d/%d/%d", &d2.month, &d2.day, &d2.year);
22
23      if(compare_dates(d1, d2) < 0){
24          put_date(d1);
25          printf(" is earlier then ");
26          put_date(d2);
27          printf("\n");
```

Solution

```
28      }else{
29          put_date(d2);
30          printf(" is earlier then ");
31          put_date(d1);
32          printf("\n");
33      }
34
35      return 0;
36  }
37
38  int compare_dates(struct date d1, struct date d2){
39
40      if(d1.year != d2.year)
41          return d1.year < d2.year ? -1 : 1;
42      if(d1.month != d2.month)
43          return d1.month < d2.month ? -1 : 1;
44      if(d1.day != d2.day)
45          return d1.day < d2.day ? -1 : 1;
46
47      return 0;
48  }
49
50  void put_date(struct date d){
51      printf("%d/%d/%.2d", d.month, d.day, d.year);
52  }
```