計算機程式語言

教課教授 : 謝東儒

助教 :蔡詠聿、吳品頤

Chapter 16_project 1

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.

Solution

```c
#include <stdio.h>

#define ARRAY_LEN(x) ((int) (sizeof((x)) / sizeof((x)[0])))

const struct {
    [redacted]
} country_codes[] =
    {{"Argentina",           54}, {"Bangladesh",     889},
     {"Brazil",              55}, {"Burma (Myanmar)", 95},
     {"China",               86}, {"Colombia",        57},
     {"Congo, Dem. Rep. of", 243}, {"Egypt",          20},
     {"Ethiopia",            251}, {"France",         33},
     {"Germany",             49}, {"India",           91},
     {"Indonesia",           62}, {"Iran",            98},
     {"Italy",               39}, {"Japan",           81},
     {"Mexico",              52}, {"Nigeria",        234},
     {"Pakistan",            92}, {"Philippines",     63},
     {"Poland",              48}, {"Russia",           7},
     {"South Africa",        27}, {"South Korea",     82},
     {"Spain",               34}, {"Sudan",          249},
     {"Thailand",            66}, {"Turkey",          90},
     {"Ukraine",            380}, {"United Kingdom",  44},
     {"United States",        1}, {"Vietnam",         84}};

```

Solution

```
26  int main(void) {
27
28      int code, i;
29
30      printf("Enter an international dialing code: ");
31      scanf("%d", &code);
32
33
34
35
36
37
38
39
40      printf("Error: %d is not a valid country code.\n", code);
41      return 0;
42  }
```

Example

```
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
Enter an international dialing code: 2
Error: 2 is not a valid country code.
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
Enter an international dialing code: 1
1 is the code for United States.
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
Enter an international dialing code: 249
249 is the code for Sudan.
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
Enter an international dialing code: 999
Error: 999 is not a valid country code.
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$
```

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.

Solution

```
1   //  inventory2
2
3   #include <stdio.h>
4   #include "readline/readline.h"
5
6   #define NAME_LEN 25
7   #define MAX_PARTS 100
8
9   struct part{
10      int number;
11      char name[NAME_LEN + 1];
12      float price;
13      int on_hand;
14  } inventory[MAX_PARTS];
15
16  int num_parts = 0;  /*  number of parts currently stored    */
17
18  int find_part(int number);
19  void insert(void);
20  void search(void);
21  void change(void);
22  void update(void);
23  void print(void);
```

Solution

```
24
25  int main(void){
26
27      char code;
28
29      for(;;){
30          printf("Enter operation code : ");
31          scanf(" %c", &code);
32          while(getchar() != '\n')    ;   /*  skips to end of line    */
33
34          switch(code){
35              case 'i':
36                  insert();
37                  break;
38              case 's':
39                  search();
40                  break;
41              case 'c':
42                  change();
43                  break;
44              case 'u':
45                  update();
46                  break;
47              case 'p':
48                  print();
49                  break;
50              case 'q':
51                  return 0;
52              default:
53                  printf("illegal code\n");
54          }
55          printf("\n");
56      }
57
58      return 0;
59  }
60
```

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

```c
//  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;
}
```

Example

```
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop/c15$ ./invent
ory
Enter operation code: i
Enter part number: 2
Enter part name: milk tea
Enter price: 55
Enter quantity on hand: 30

Enter operation code: i
Enter part number: 3
Enter part name: hot dog
Enter price: 20
Enter quantity on hand: 50

Enter operation code: p
Part Number   Part Name                Price   Quantity on Hand
      2       milk tea               $  55.00             30
      3       hot dog                $  20.00             50

Enter operation code: s
Enter part number: 2
Part name: milk tea
Quantity on hand: 30
Price: $55.00

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: p
Part Number   Part Name                Price   Quantity on Hand
      2       milk tea               $  60.00            100
      3       hot dog                $  20.00             50

Enter operation code: q
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop/c15$
```

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.

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  }
```

Example

```
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
Enter a 24-hour time: 12:00
Closest departure time is 11:19 a.m., arriving at 1:31 p.m.
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
Enter a 24-hour time: 8:00
Closest departure time is 8:00 a.m., arriving at 10:16 a.m.
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$
```

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.

Solution

```c
// compare_dates

#include <stdio.h>

/*  Note: Program assumes years are in the same century */

struct date {
    int month, day, year;
};

int compare_dates(struct date d1, struct date d2);
void put_date(struct date d);

int main(void){

    struct date d1, d2;

    printf("Enter first date (mm/dd/yy) : ");
    scanf("%d/%d/%d", &d1.month, &d1.day, &d1.year);
    printf("Enter second date (mm/dd/yy) : ");
    scanf("%d/%d/%d", &d2.month, &d2.day, &d2.year);

    if(compare_dates(d1, d2) < 0){
        put_date(d1);
        printf(" is earlier then ");
        put_date(d2);
        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  }
```

Example

```
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
Enter first date (mm/dd/yy): 1/1/99
Enter second date (mm/dd/yy): 1/2/99
1/1/99 comes before 1/2/99
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
Enter first date (mm/dd/yy): 1/2/99
Enter second date (mm/dd/yy): 1/1/99
1/1/99 comes before 1/2/99
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$
```