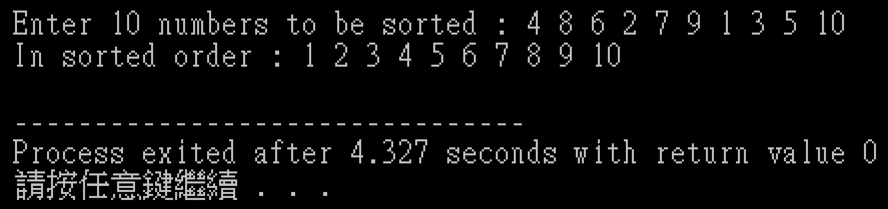
程式設計

助教：李哲源、傅繹嘉

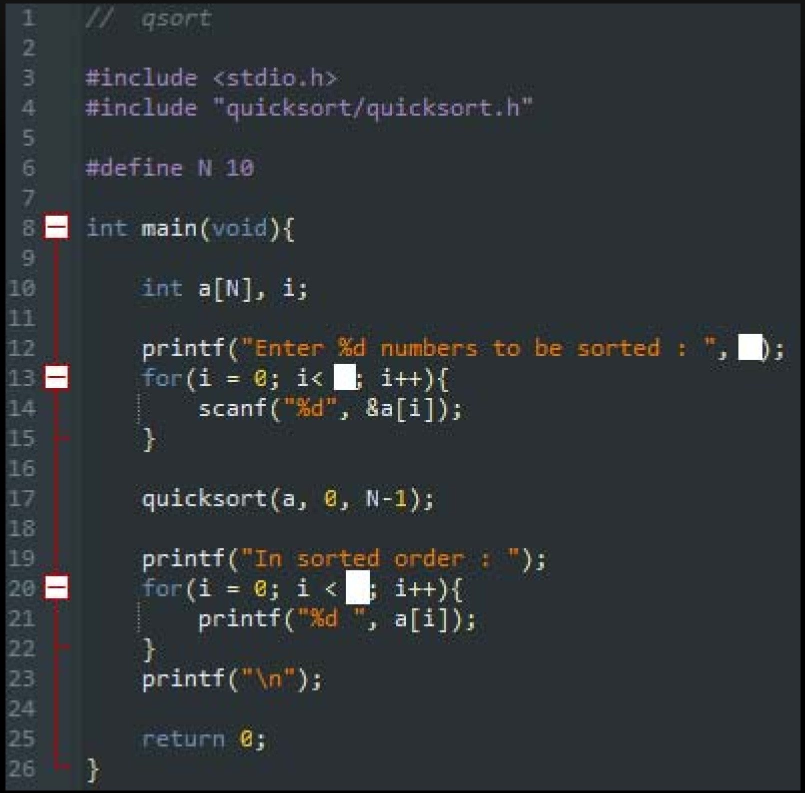
Chapter 15_problem 3

Modify the qsort.c program of Section 9.6 so that the quicksort and split functions are in a separate file named quicksort.c . Create a header file named quicksor.h that contains prototypes for the two functions and have both qsort.c and quicksort.c include this file.



```
Enter 10 numbers to be sorted : 4 8 6 2 7 9 1 3 5 10
In sorted order : 1 2 3 4 5 6 7 8 9 10

--------------------------------
Process exited after 4.327 seconds with return value 0
請按任意鍵繼續 . . .
```

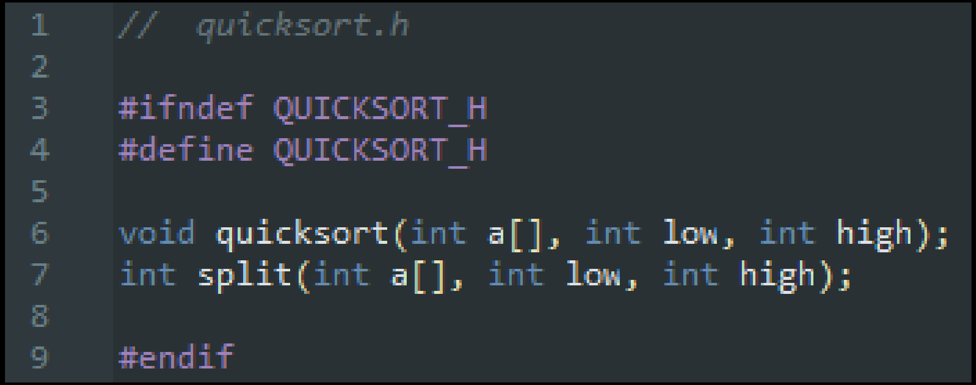
Solution



```c
//  qsort

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

#define N 10

int main(void){

    int a[N], i;

    printf("Enter %d numbers to be sorted : ",  );
    for(i = 0; i<  ; i++){
        scanf("%d", &a[i]);
    }

    quicksort(a, 0, N-1);

    printf("In sorted order : ");
    for(i = 0; i <  ; i++){
        printf("%d ", a[i]);
    }
    printf("\n");

    return 0;
}
```

Solution

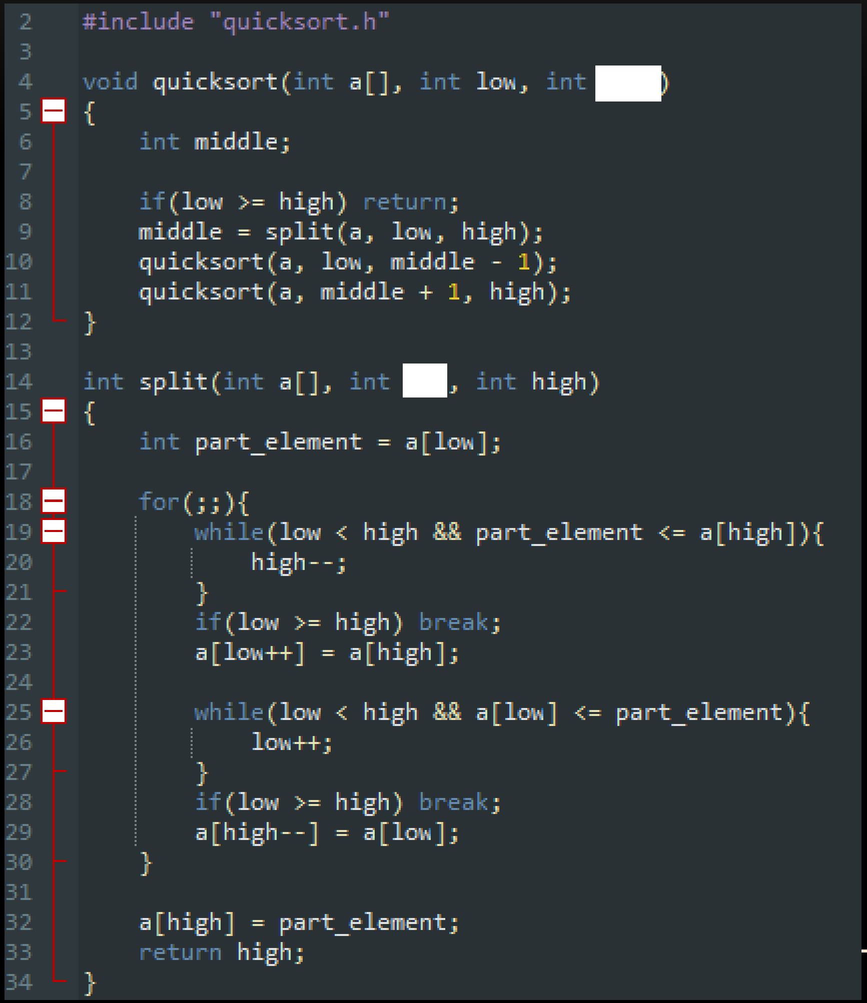
quicksort.h



```c
//  quicksort.h

#ifndef QUICKSORT_H
#define QUICKSORT_H

void quicksort(int a[], int low, int high);
int split(int a[], int low, int high);

#endif
```

Solution

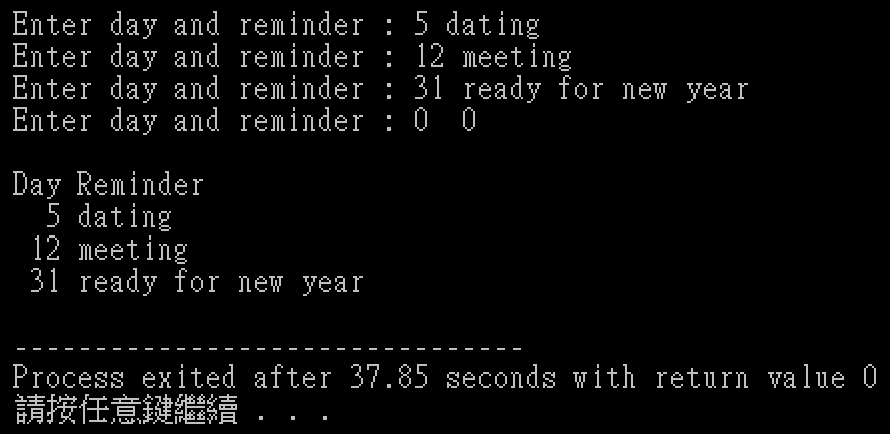
quicksort.c



```c
#include "quicksort.h"

void quicksort(int a[], int low, int [redacted])
{
    int middle;

    if(low >= high) return;
    middle = split(a, low, high);
    quicksort(a, low, middle - 1);
    quicksort(a, middle + 1, high);
}

int split(int a[], int [redacted], int high)
{
    int part_element = a[low];

    for(;;){
        while(low < high && part_element <= a[high]){
            high--;
        }
        if(low >= high) break;
        a[low++] = a[high];

        while(low < high && a[low] <= part_element){
            low++;
        }
        if(low >= high) break;
        a[high--] = a[low];
    }

    a[high] = part_element;
    return high;
}
```

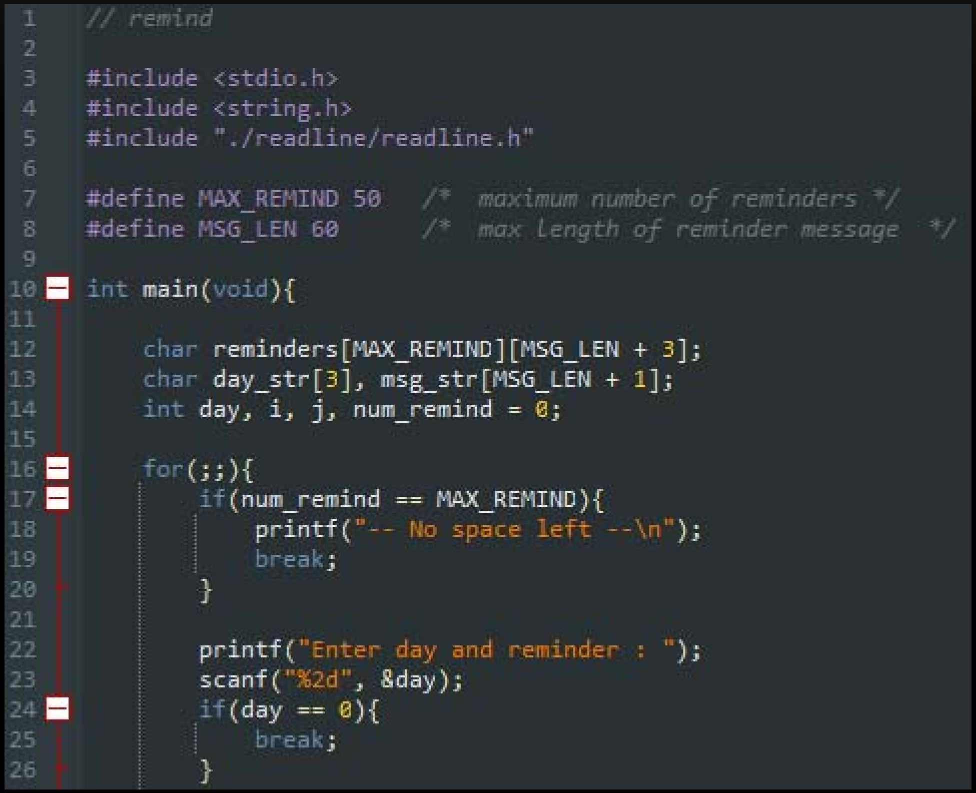
Chapter 15_problem 4

Modify the remind.c program of Section 13.5 so that the read_line function is in a separate file named readline.c . Create a header file named readline.h that contains a prototype for the function and have both remind.c and readline.c include this file.



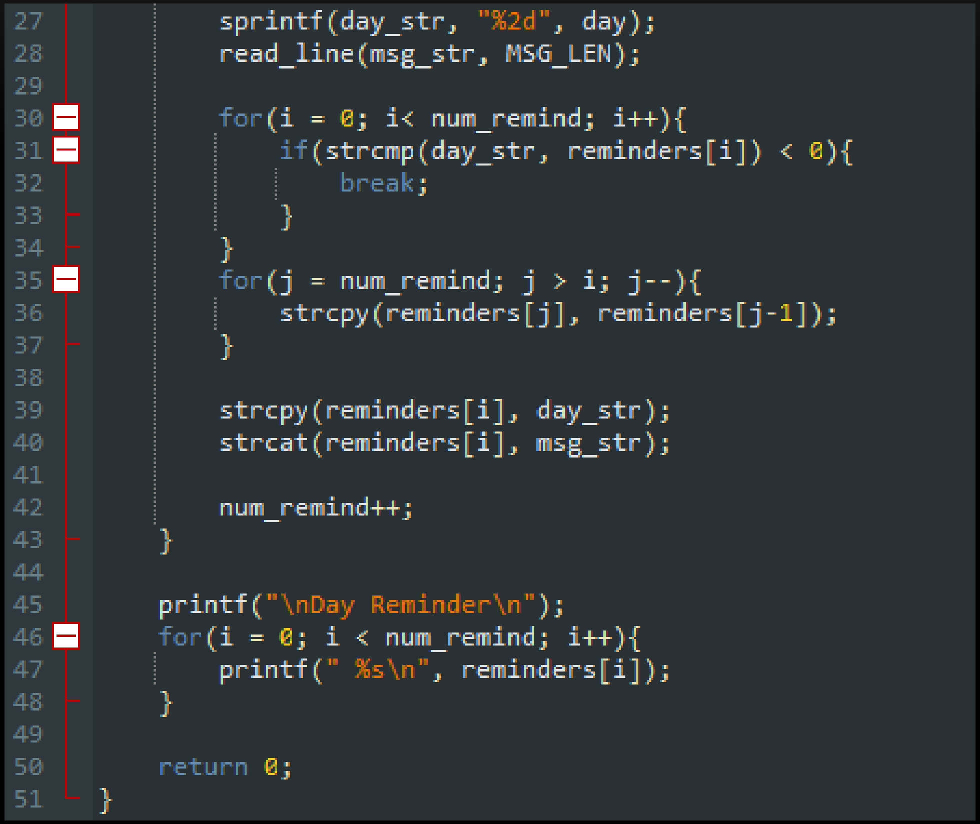
```
Enter day and reminder : 5 dating
Enter day and reminder : 12 meeting
Enter day and reminder : 31 ready for new year
Enter day and reminder : 0  0

Day Reminder
  5 dating
 12 meeting
 31 ready for new year

--------------------------------
Process exited after 37.85 seconds with return value 0
請按任意鍵繼續 . . .
```

Solution



```c
// remind

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

#define MAX_REMIND 50   /*  maximum number of reminders */
#define MSG_LEN 60      /*  max length of reminder message  */

int main(void){

    char reminders[MAX_REMIND][MSG_LEN + 3];
    char day_str[3], msg_str[MSG_LEN + 1];
    int day, i, j, num_remind = 0;

    for(;;){
        if(num_remind == MAX_REMIND){
            printf("-- No space left --\n");
            break;
        }

        printf("Enter day and reminder : ");
        scanf("%2d", &day);
        if(day == 0){
            break;
        }
```

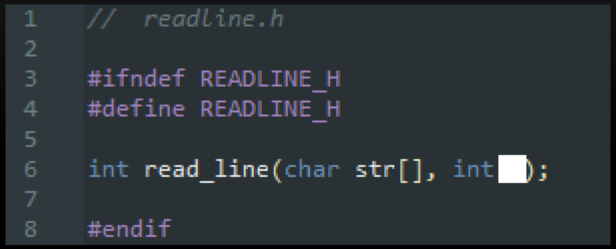
Solution



```
        sprintf(day_str, "%2d", day);
        read_line(msg_str, MSG_LEN);

        for(i = 0; i< num_remind; i++){
            if(strcmp(day_str, reminders[i]) < 0){
                break;
            }
        }
        for(j = num_remind; j > i; j--){
            strcpy(reminders[j], reminders[j-1]);
        }

        strcpy(reminders[i], day_str);
        strcat(reminders[i], msg_str);

        num_remind++;
    }

    printf("\nDay Reminder\n");
    for(i = 0; i < num_remind; i++){
        printf(" %s\n", reminders[i]);
    }

    return 0;
}
```

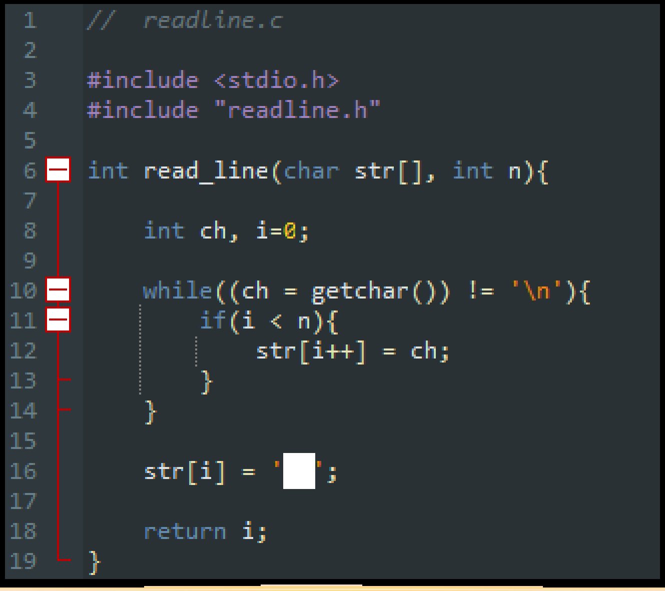
Solution

readline.h



```c
//  readline.h

#ifndef READLINE_H
#define READLINE_H

int read_line(char str[], int  );

#endif
```

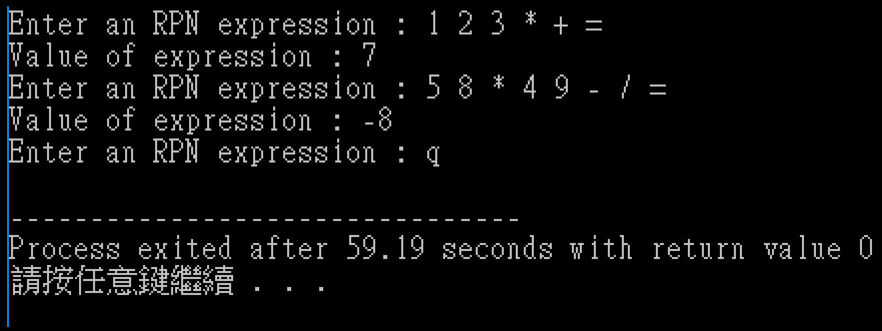
readline.c



```c
//  readline.c

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

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

    int ch, i=0;

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

    str[i] = '  ';

    return i;
}
```

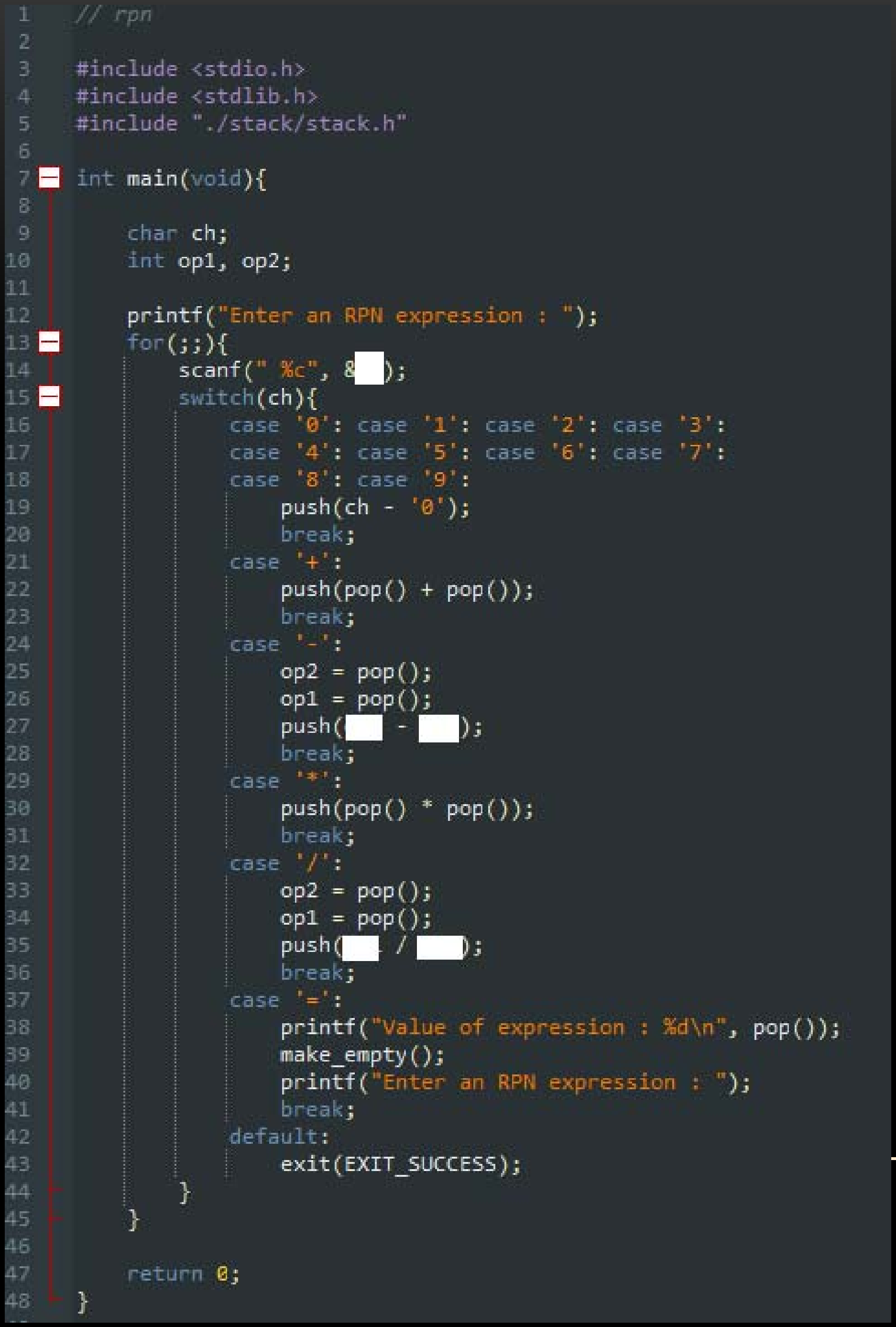
Chapter 15_problem 5

Modify Programming Project 6 from Chapter 10 so that it has separate stack.h and stack.c files, as described in Section 15.2.



```
Enter an RPN expression : 1 2 3 * + =
Value of expression : 7
Enter an RPN expression : 5 8 * 4 9 - / =
Value of expression : -8
Enter an RPN expression : q

--------------------------------
Process exited after 59.19 seconds with return value 0
請按任意鍵繼續 . . .
```

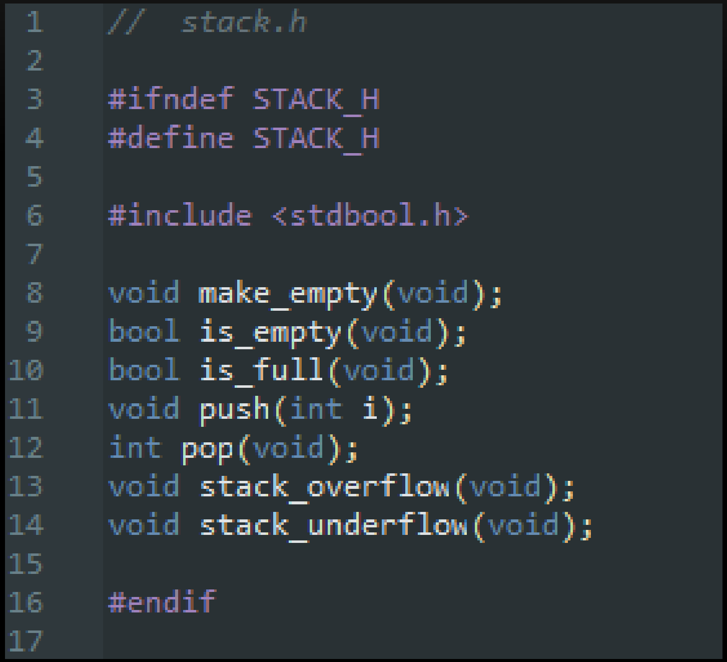
Solution



```c
// rpn

#include <stdio.h>
#include <stdlib.h>
#include "./stack/stack.h"

int main(void){

    char ch;
    int op1, op2;

    printf("Enter an RPN expression : ");
    for(;;){
        scanf(" %c", &[redacted]);
        switch(ch){
            case '0': case '1': case '2': case '3':
            case '4': case '5': case '6': case '7':
            case '8': case '9':
                push(ch - '0');
                break;
            case '+':
                push(pop() + pop());
                break;
            case '-':
                op2 = pop();
                op1 = pop();
                push([redacted] - [redacted]);
                break;
            case '*':
                push(pop() * pop());
                break;
            case '/':
                op2 = pop();
                op1 = pop();
                push([redacted] / [redacted]);
                break;
            case '=':
                printf("Value of expression : %d\n", pop());
                make_empty();
                printf("Enter an RPN expression : ");
                break;
            default:
                exit(EXIT_SUCCESS);
        }
    }

    return 0;
}
```

Solution

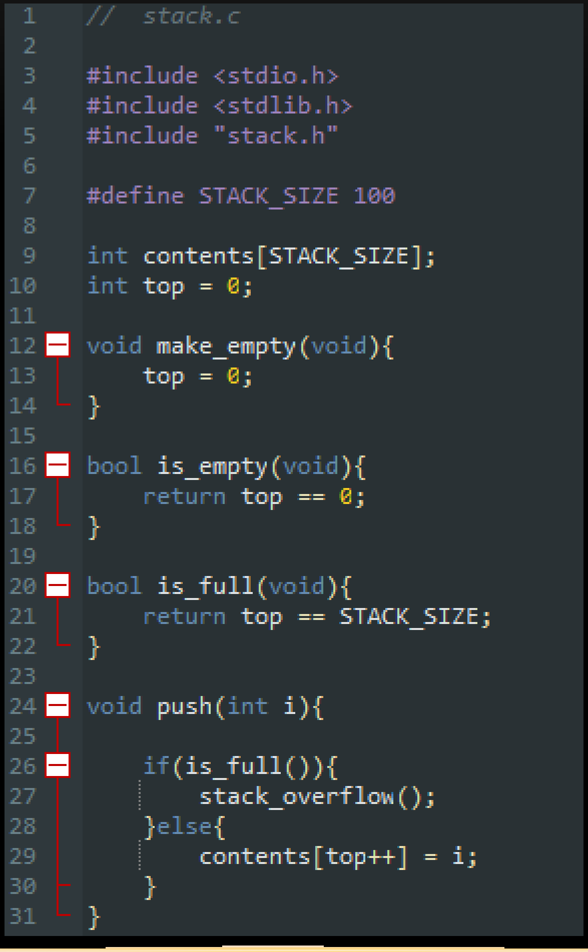
stack.h



```c
//  stack.h

#ifndef STACK_H
#define STACK_H

#include <stdbool.h>

void make_empty(void);
bool is_empty(void);
bool is_full(void);
void push(int i);
int pop(void);
void stack_overflow(void);
void stack_underflow(void);

#endif

```

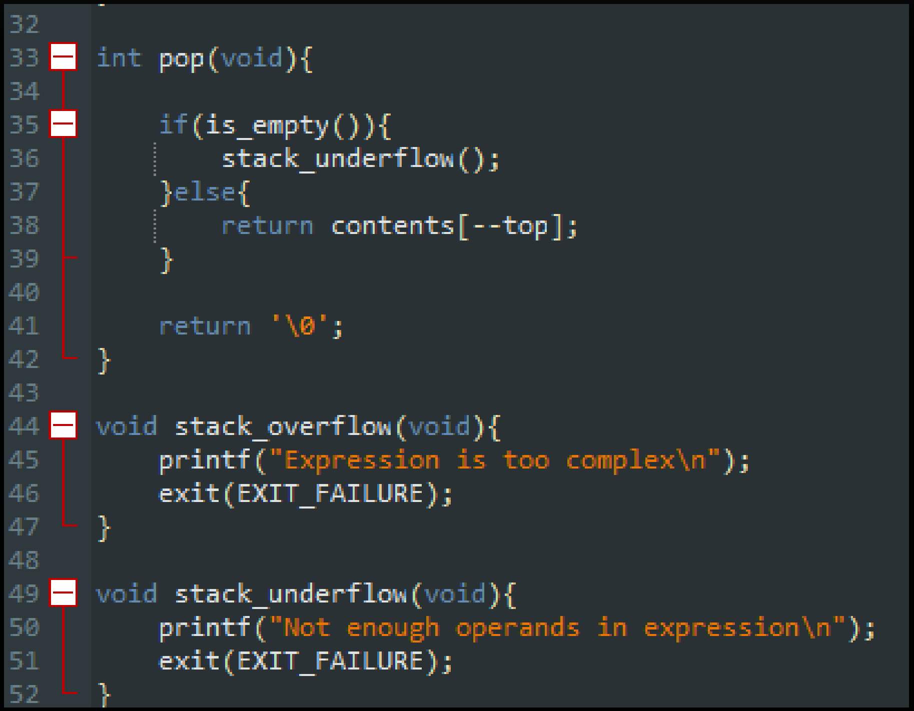
Solution

stack.c



```c
//  stack.c

#include <stdio.h>
#include <stdlib.h>
#include "stack.h"

#define STACK_SIZE 100

int contents[STACK_SIZE];
int top = 0;

void make_empty(void){
    top = 0;
}

bool is_empty(void){
    return top == 0;
}

bool is_full(void){
    return top == STACK_SIZE;
}

void push(int i){

    if(is_full()){
        stack_overflow();
    }else{
        contents[top++] = i;
    }
}
```

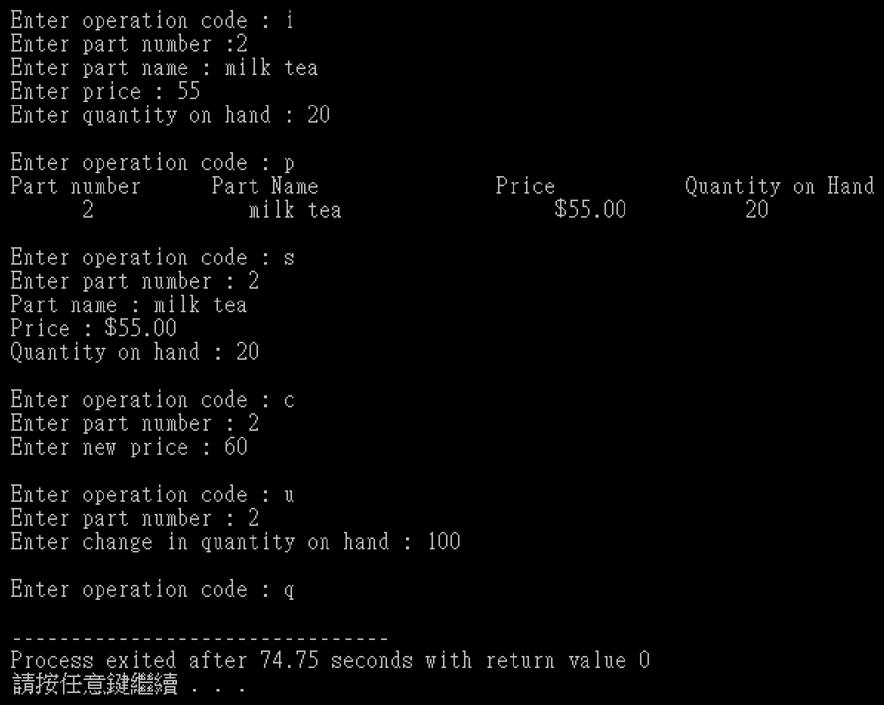
Solution



```c
int pop(void){

    if(is_empty()){
        stack_underflow();
    }else{
        return contents[--top];
    }

    return '\0';
}

void stack_overflow(void){
    printf("Expression is too complex\n");
    exit(EXIT_FAILURE);
}

void stack_underflow(void){
    printf("Not enough operands in expression\n");
    exit(EXIT_FAILURE);
}
```

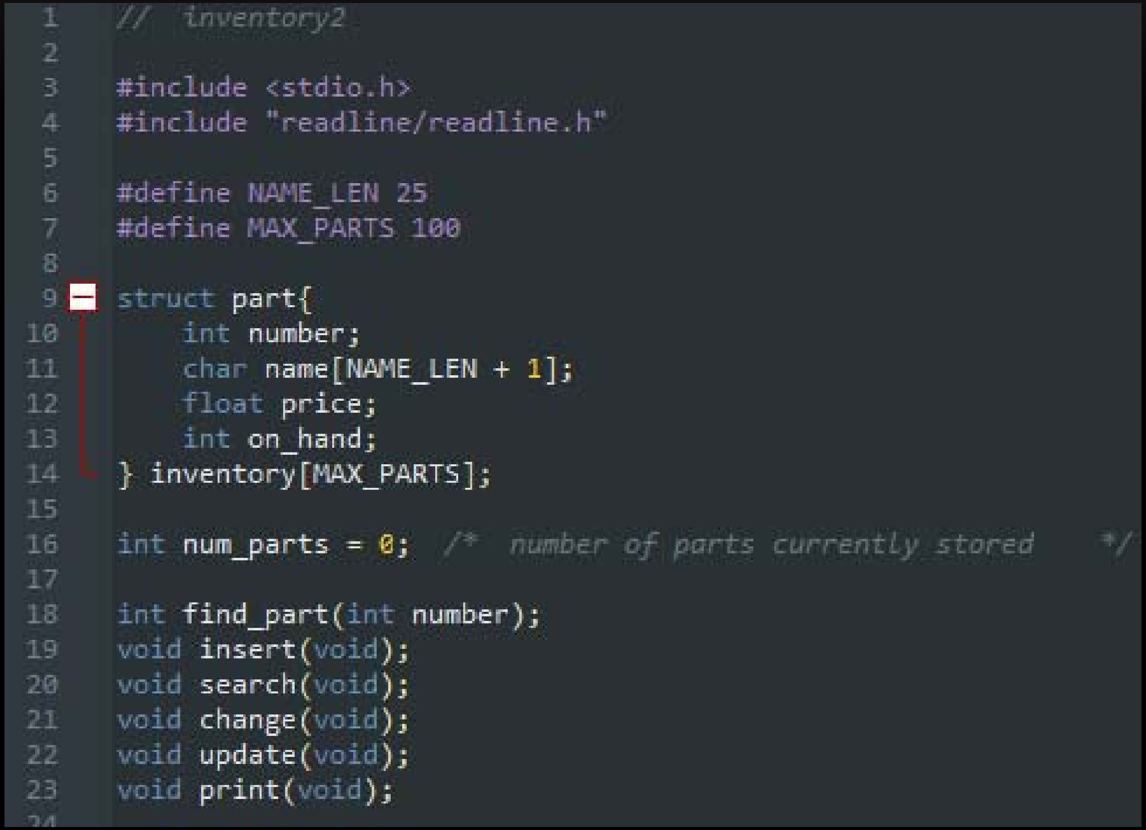
Chapter 16_problem 4

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.



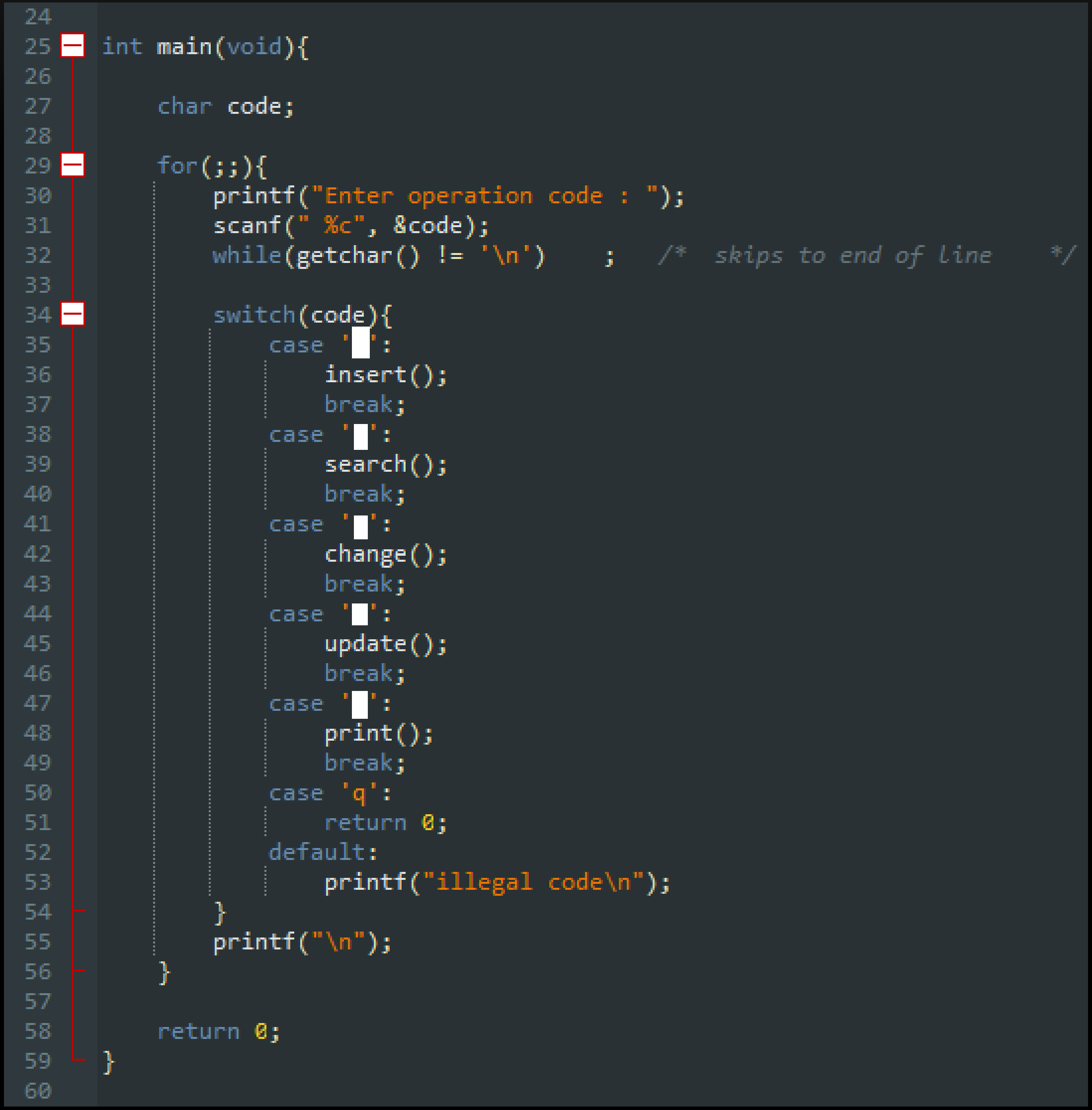
```
Enter operation code : i
Enter part number :2
Enter part name : milk tea
Enter price : 55
Enter quantity on hand : 20

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



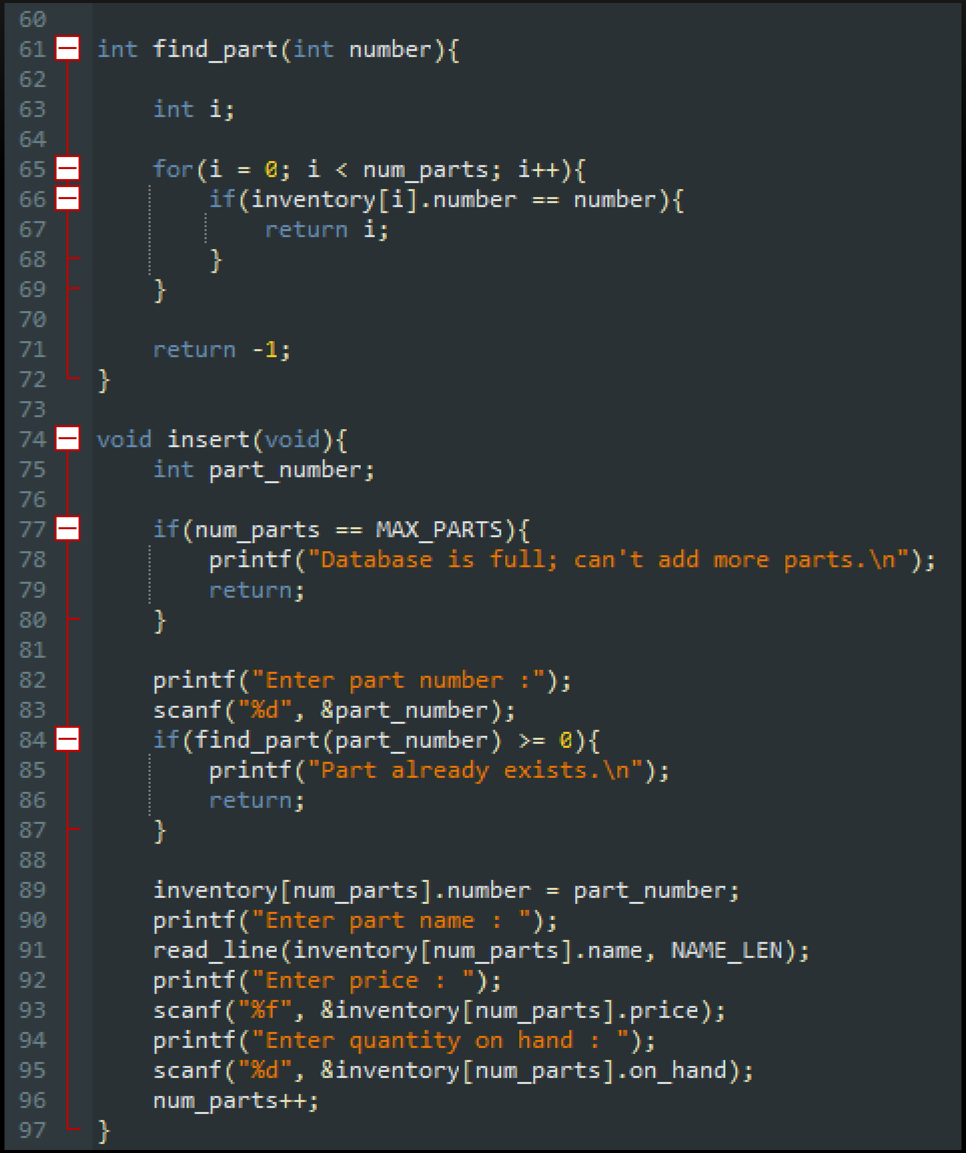
```
//  inventory2

#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



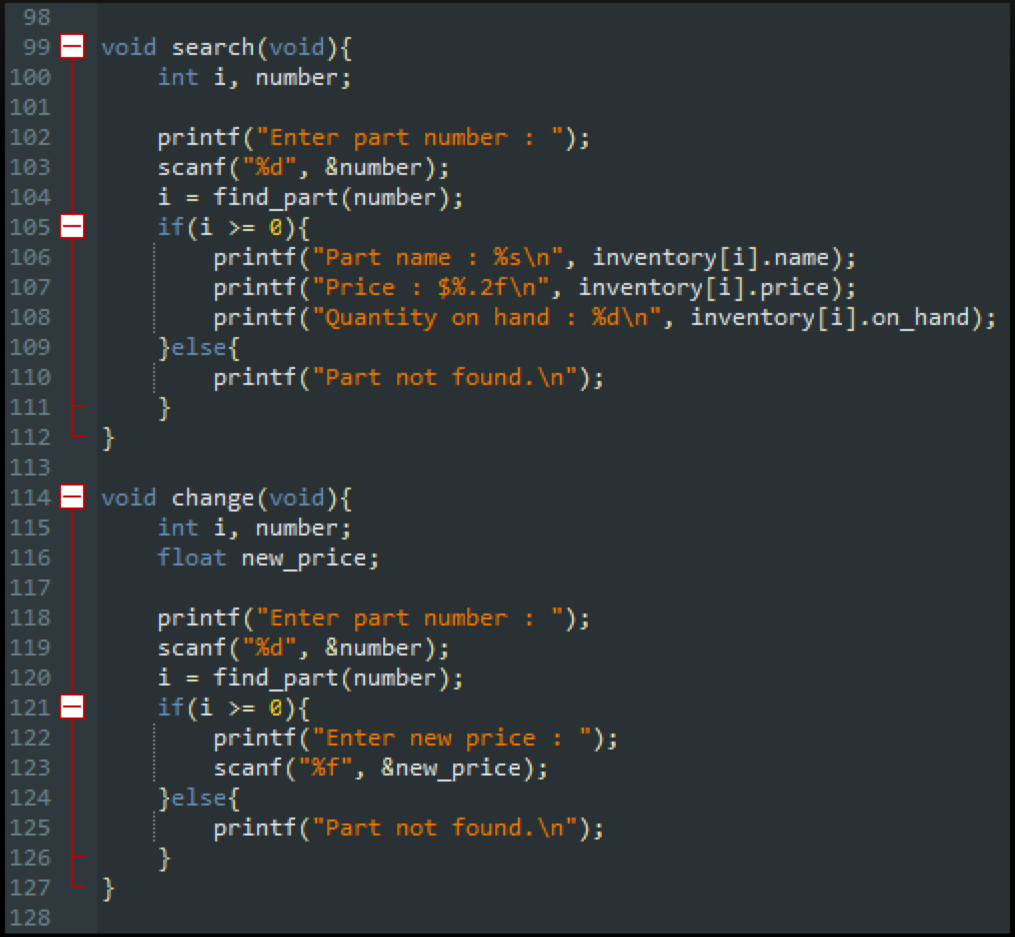
```c

int main(void){

    char code;

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

        switch(code){
            case '[redacted]':
                insert();
                break;
            case '[redacted]':
                search();
                break;
            case '[redacted]':
                change();
                break;
            case '[redacted]':
                update();
                break;
            case '[redacted]':
                print();
                break;
            case 'q':
                return 0;
            default:
                printf("illegal code\n");
        }
        printf("\n");
    }

    return 0;
}

```

Solution



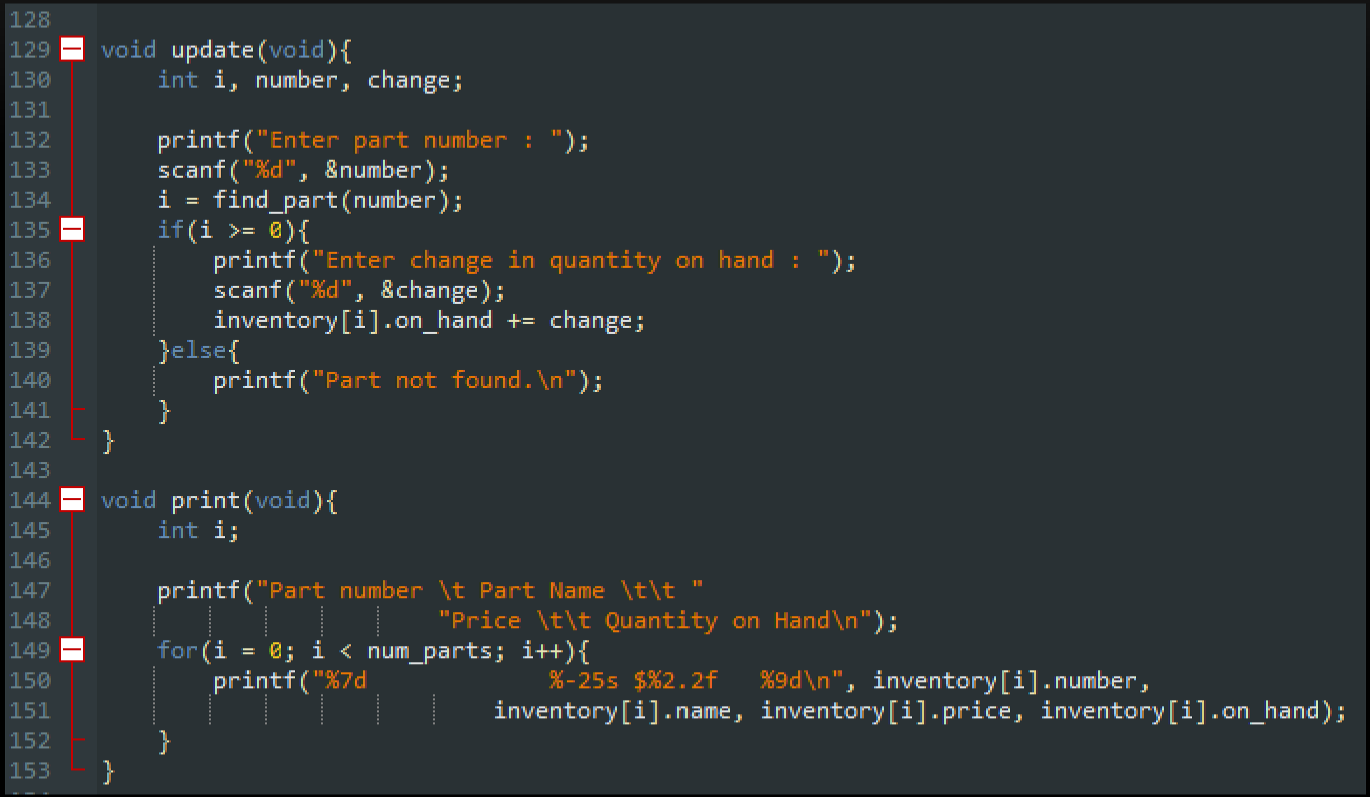
```c

int find_part(int number){

    int i;

    for(i = 0; i < num_parts; i++){
        if(inventory[i].number == number){
            return i;
        }
    }

    return -1;
}

void insert(void){
    int part_number;

    if(num_parts == MAX_PARTS){
        printf("Database is full; can't add more parts.\n");
        return;
    }

    printf("Enter part number :");
    scanf("%d", &part_number);
    if(find_part(part_number) >= 0){
        printf("Part already exists.\n");
        return;
    }

    inventory[num_parts].number = part_number;
    printf("Enter part name : ");
    read_line(inventory[num_parts].name, NAME_LEN);
    printf("Enter price : ");
    scanf("%f", &inventory[num_parts].price);
    printf("Enter quantity on hand : ");
    scanf("%d", &inventory[num_parts].on_hand);
    num_parts++;
}
```

Solution



```c

void search(void){
    int i, number;

    printf("Enter part number : ");
    scanf("%d", &number);
    i = find_part(number);
    if(i >= 0){
        printf("Part name : %s\n", inventory[i].name);
        printf("Price : $%.2f\n", inventory[i].price);
        printf("Quantity on hand : %d\n", inventory[i].on_hand);
    }else{
        printf("Part not found.\n");
    }
}

void change(void){
    int i, number;
    float new_price;

    printf("Enter part number : ");
    scanf("%d", &number);
    i = find_part(number);
    if(i >= 0){
        printf("Enter new price : ");
        scanf("%f", &new_price);
    }else{
        printf("Part not found.\n");
    }
}

```

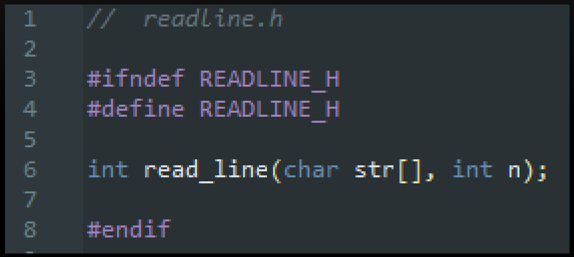
Solution



```c

void update(void){
    int i, number, change;

    printf("Enter part number : ");
    scanf("%d", &number);
    i = find_part(number);
    if(i >= 0){
        printf("Enter change in quantity on hand : ");
        scanf("%d", &change);
        inventory[i].on_hand += change;
    }else{
        printf("Part not found.\n");
    }
}

void print(void){
    int i;

    printf("Part number \t Part Name \t\t "
                        "Price \t\t Quantity on Hand\n");
    for(i = 0; i < num_parts; i++){
        printf("%7d             %-25s $%2.2f   %9d\n", inventory[i].number,
                            inventory[i].name, inventory[i].price, inventory[i].on_hand);
    }
}
```

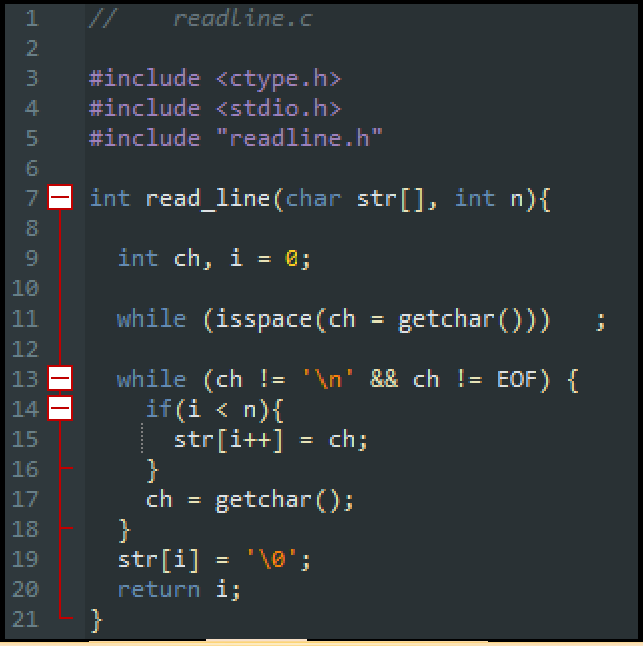
Solution

readline.h



```c
//  readline.h

#ifndef READLINE_H
#define READLINE_H

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

#endif
```

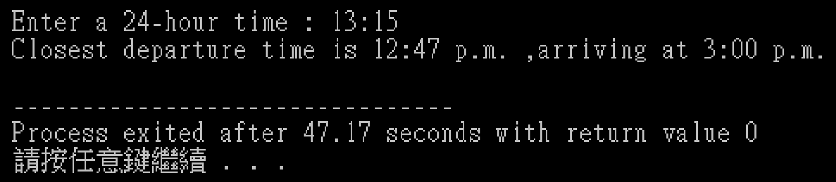
readline.c



```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;
}
```

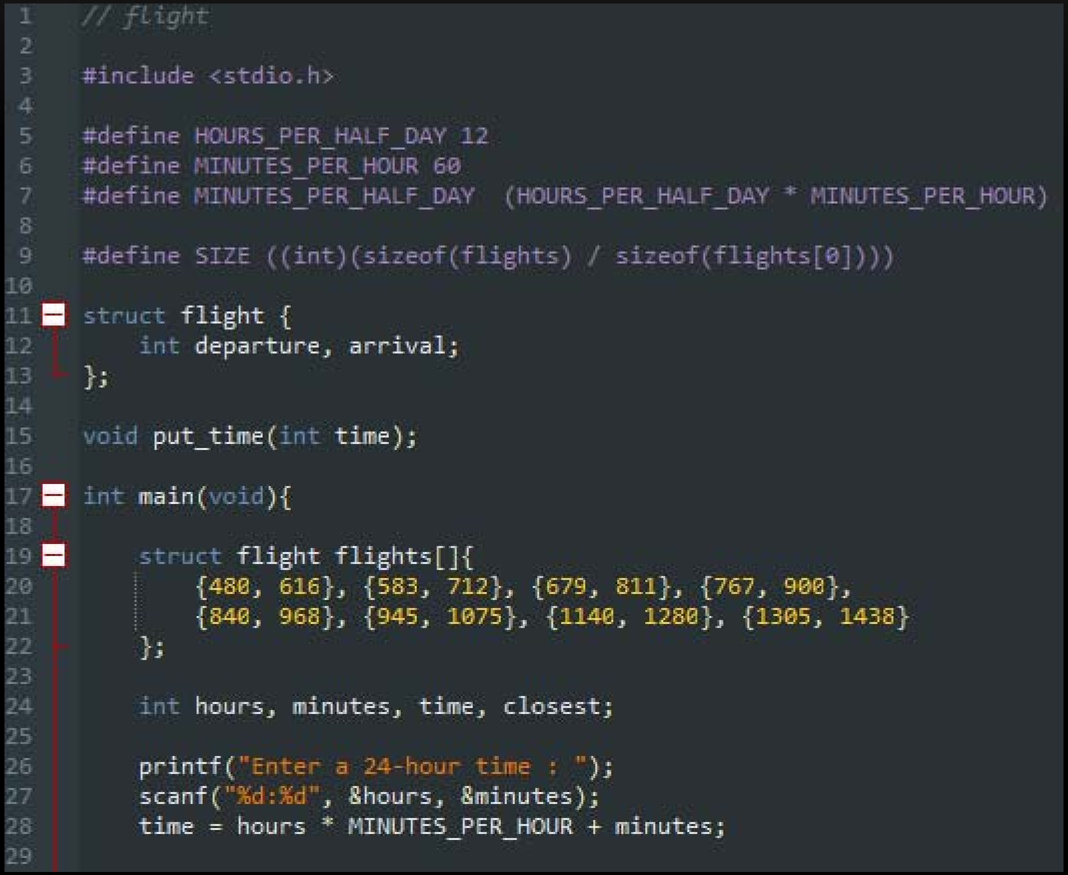
Chapter 16_problem 5

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.



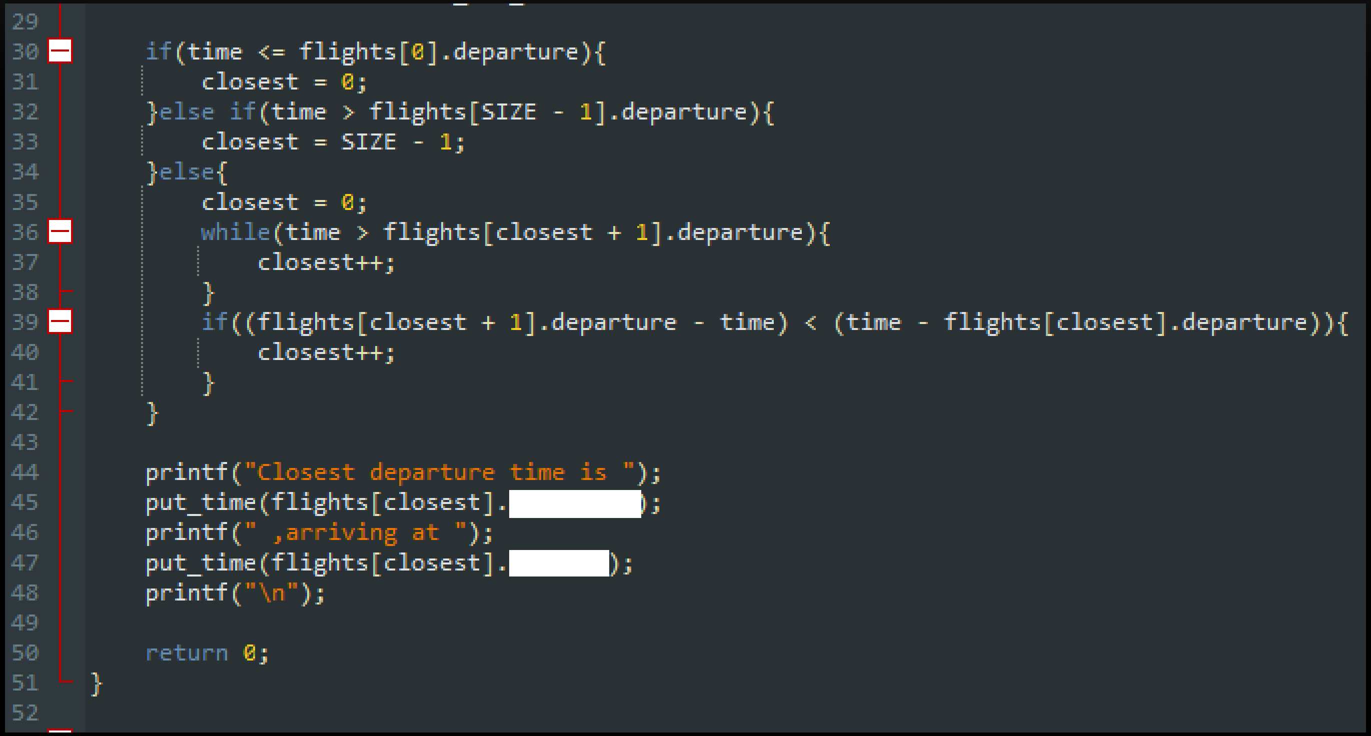
```
Enter a 24-hour time : 13:15
Closest departure time is 12:47 p.m. ,arriving at 3:00 p.m.

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

Solution



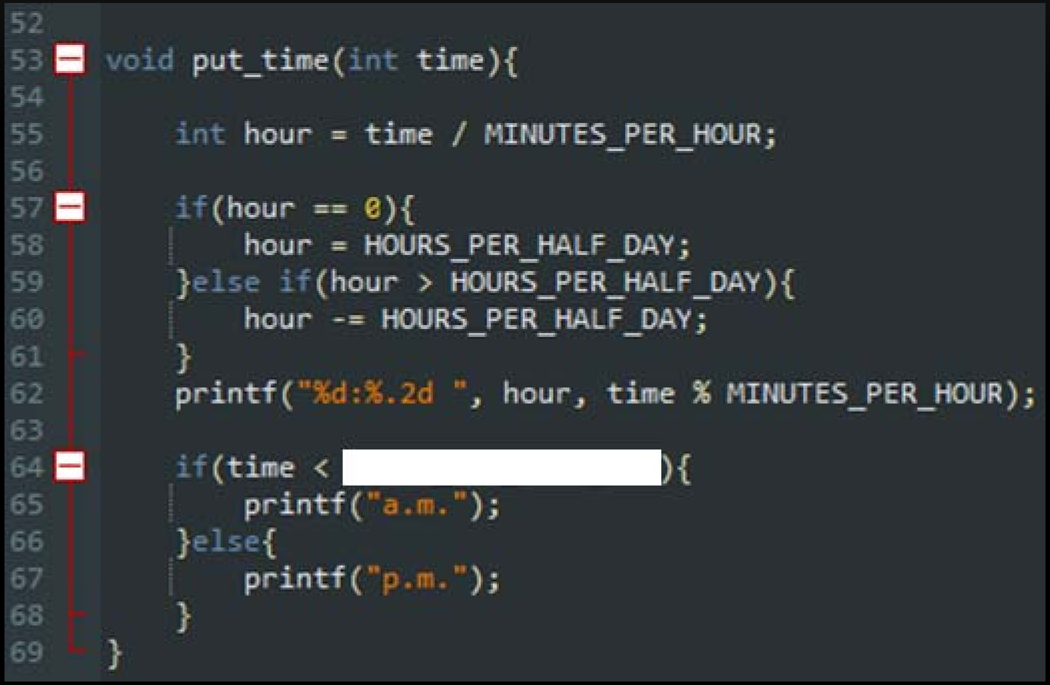
```c
// flight

#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



```c

    if(time <= flights[0].departure){
        closest = 0;
    }else if(time > flights[SIZE - 1].departure){
        closest = SIZE - 1;
    }else{
        closest = 0;
        while(time > flights[closest + 1].departure){
            closest++;
        }
        if((flights[closest + 1].departure - time) < (time - flights[closest].departure)){
            closest++;
        }
    }

    printf("Closest departure time is ");
    put_time(flights[closest].          );
    printf(" ,arriving at ");
    put_time(flights[closest].        );
    printf("\n");

    return 0;
}

```

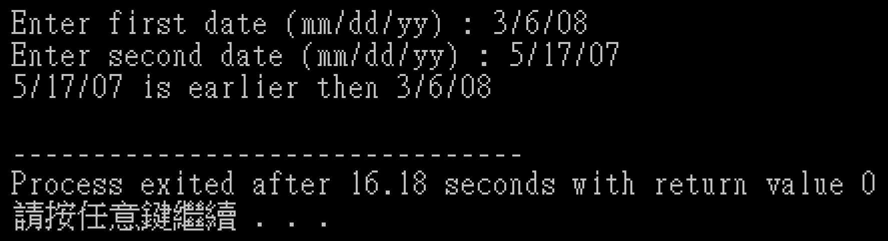
Solution



```c

void put_time(int time){

    int hour = time / MINUTES_PER_HOUR;

    if(hour == 0){
        hour = HOURS_PER_HALF_DAY;
    }else if(hour > HOURS_PER_HALF_DAY){
        hour -= HOURS_PER_HALF_DAY;
    }
    printf("%d:%.2d ", hour, time % MINUTES_PER_HOUR);

    if(time < [redacted]){
        printf("a.m.");
    }else{
        printf("p.m.");
    }
}
```

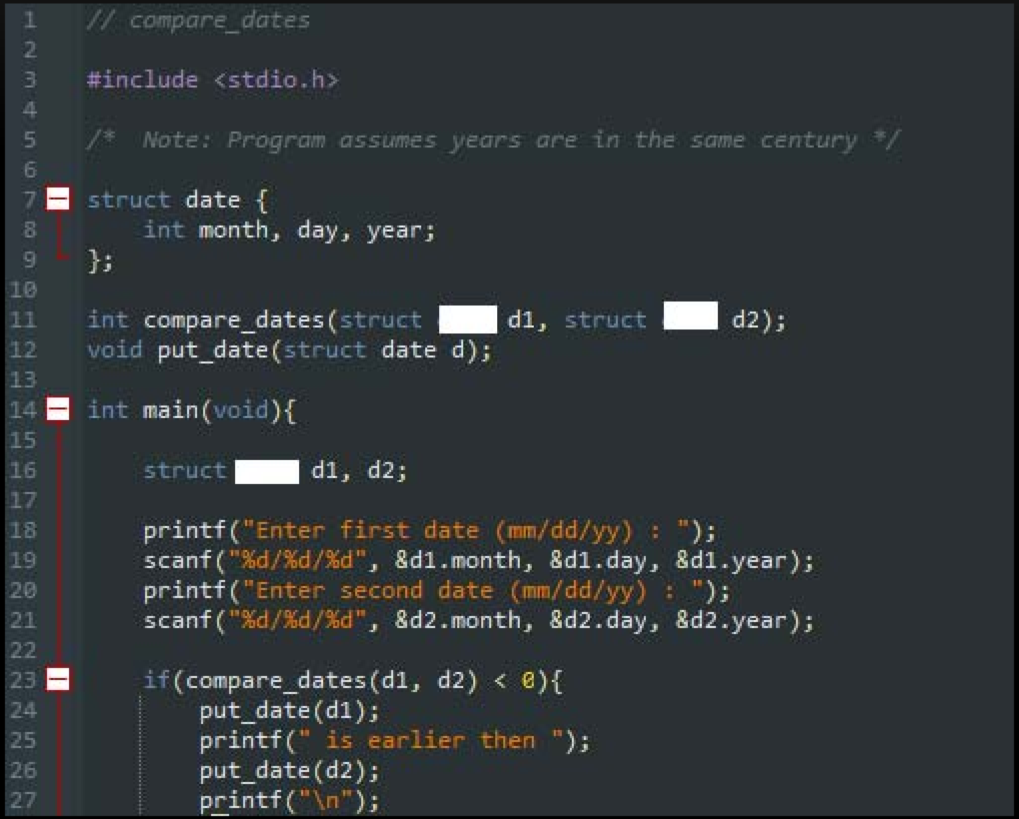
Chapter 16_problem 6

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.



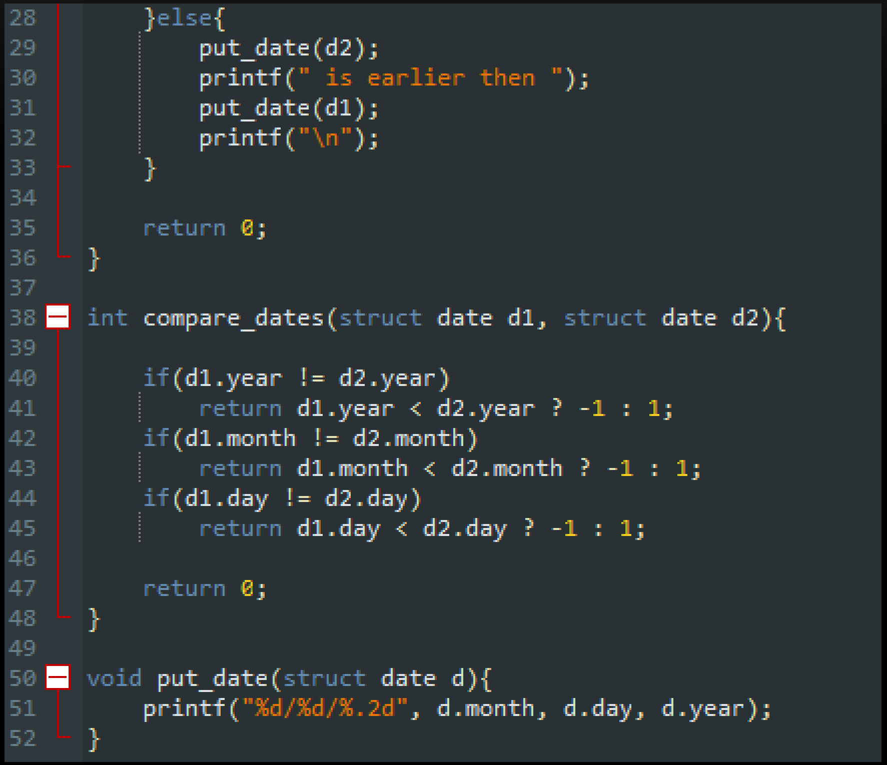
```
Enter first date (mm/dd/yy) : 3/6/08
Enter second date (mm/dd/yy) : 5/17/07
5/17/07 is earlier then 3/6/08

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

Solution



```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       d1, struct       d2);
void put_date(struct date d);

int main(void){

    struct       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

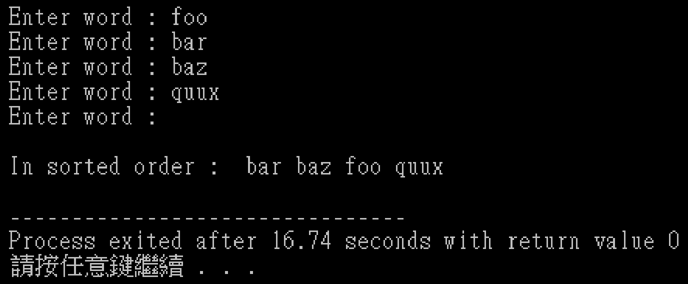


```
    }else{
        put_date(d2);
        printf(" is earlier then ");
        put_date(d1);
        printf("\n");
    }

    return 0;
}

int compare_dates(struct date d1, struct date d2){

    if(d1.year != d2.year)
        return d1.year < d2.year ? -1 : 1;
    if(d1.month != d2.month)
        return d1.month < d2.month ? -1 : 1;
    if(d1.day != d2.day)
        return d1.day < d2.day ? -1 : 1;

    return 0;
}

void put_date(struct date d){
    printf("%d/%d/%.2d", d.month, d.day, d.year);
}
```

Chapter 17_problem 5

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

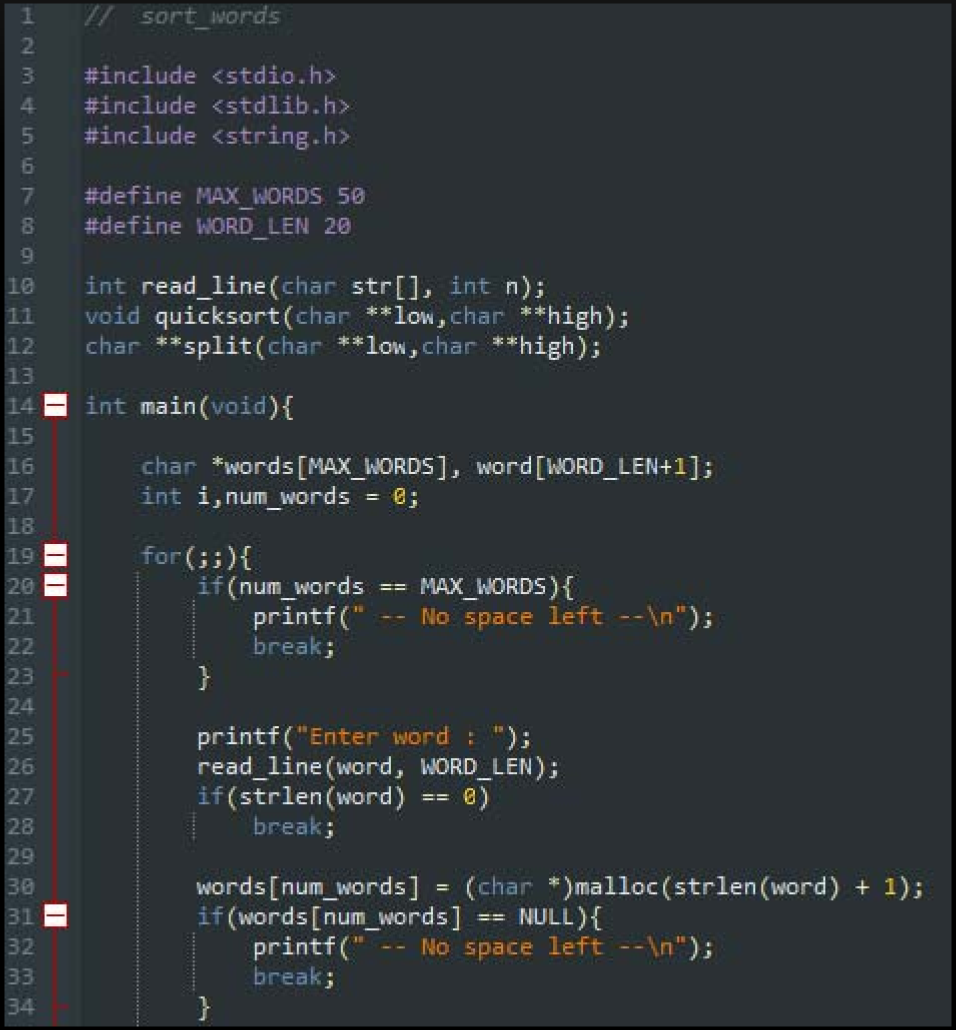
```
Enter word： foo
Enter word： bar
Enter word： baz
Enter word ： quux
Enter word：

In sorted order： bar baz foo quux
```



```
Enter word : foo
Enter word : bar
Enter word : baz
Enter word : quux
Enter word :

In sorted order :  bar baz foo quux

--------------------------------
Process exited after 16.74 seconds with return value 0
請按任意鍵繼續 . . .
```

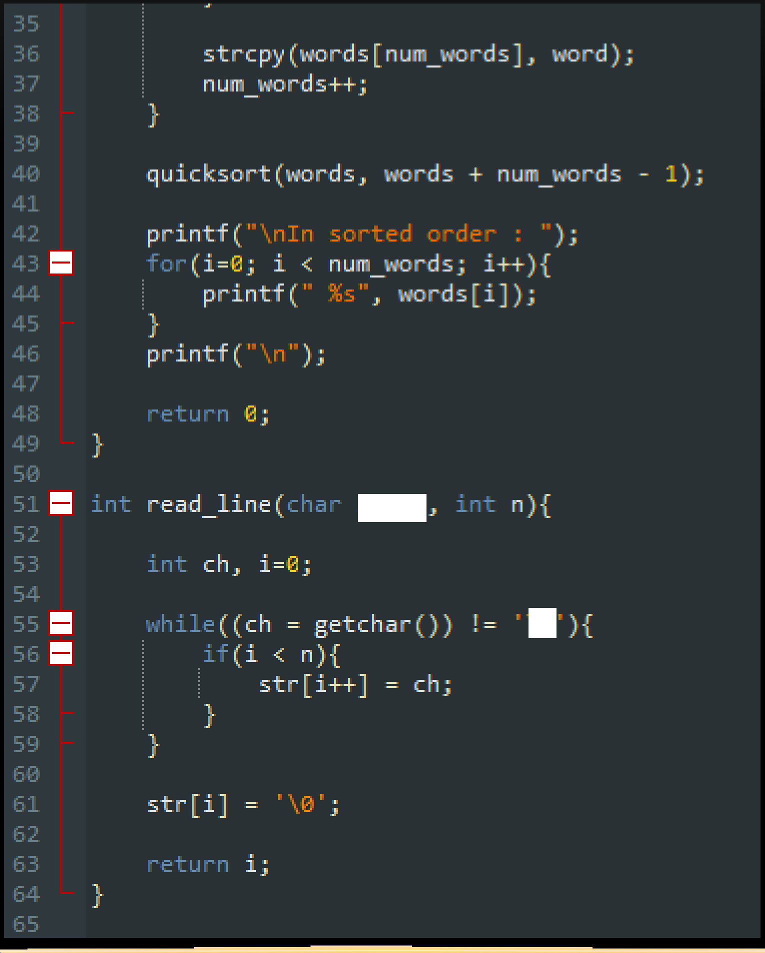
Assume that each word is no more than 20 characters long. Stop reading when the user enters an empty word (i.e., presses Enter without entering a word). Store each word in a dynamically allocated string, using an array of pointers to keep track of the strings, as in the remind2.c program (Section 17.2). 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.

Solution



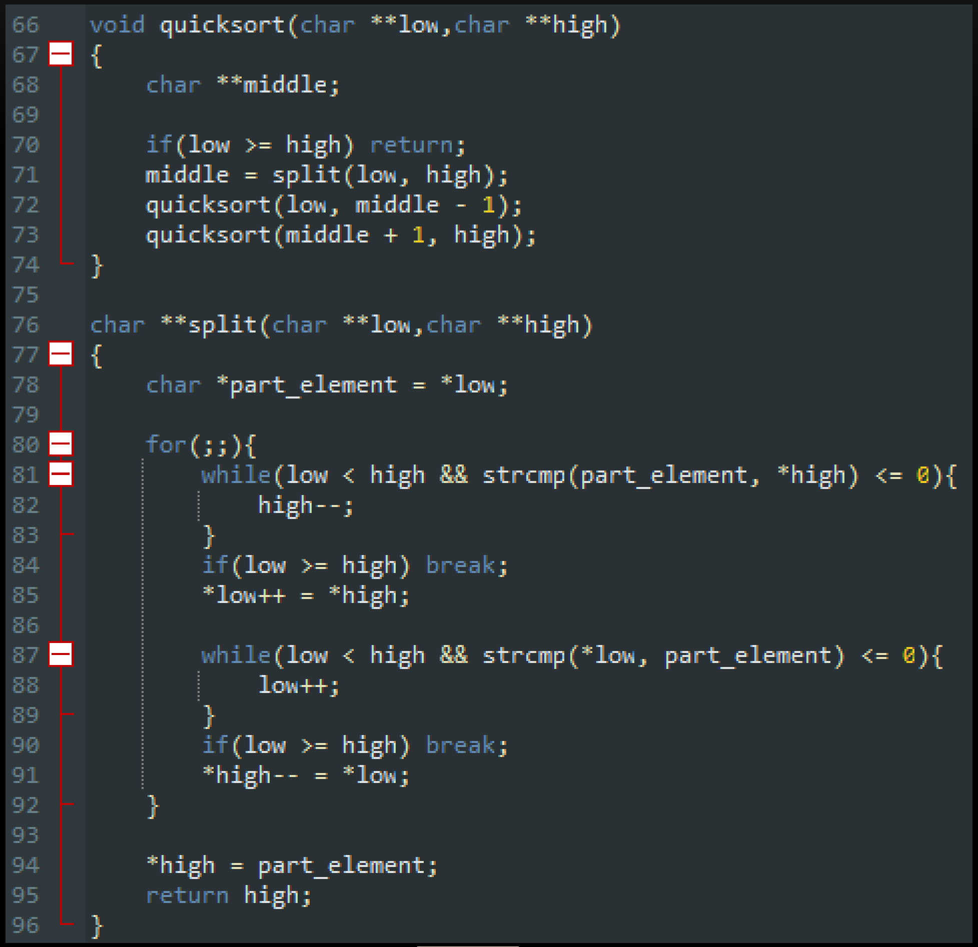
```
//  sort_words

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_WORDS 50
#define WORD_LEN 20

int read_line(char str[], int n);
void quicksort(char **low,char **high);
char **split(char **low,char **high);

int main(void){

    char *words[MAX_WORDS], word[WORD_LEN+1];
    int i,num_words = 0;

    for(;;){
        if(num_words == MAX_WORDS){
            printf(" -- No space left --\n");
            break;
        }

        printf("Enter word : ");
        read_line(word, WORD_LEN);
        if(strlen(word) == 0)
            break;

        words[num_words] = (char *)malloc(strlen(word) + 1);
        if(words[num_words] == NULL){
            printf(" -- No space left --\n");
            break;
        }
```

Solution



```c
        strcpy(words[num_words], word);
        num_words++;
    }

    quicksort(words, words + num_words - 1);

    printf("\nIn sorted order : ");
    for(i=0; i < num_words; i++){
        printf(" %s", words[i]);
    }
    printf("\n");

    return 0;
}

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

    int ch, i=0;

    while((ch = getchar()) != '[redacted]'){
        if(i < n){
            str[i++] = ch;
        }
    }

    str[i] = '\0';

    return i;
}
```

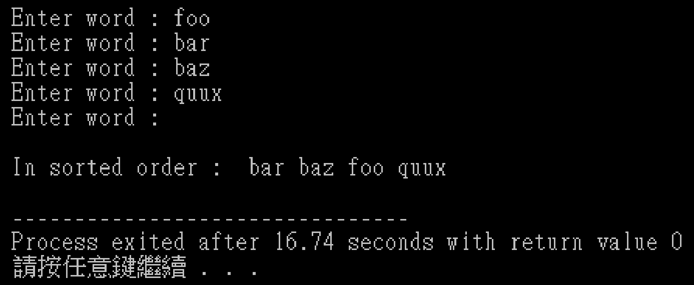
Solution



```
void quicksort(char **low,char **high)
{
    char **middle;

    if(low >= high) return;
    middle = split(low, high);
    quicksort(low, middle - 1);
    quicksort(middle + 1, high);
}

char **split(char **low,char **high)
{
    char *part_element = *low;

    for(;;){
        while(low < high && strcmp(part_element, *high) <= 0){
            high--;
        }
        if(low >= high) break;
        *low++ = *high;

        while(low < high && strcmp(*low, part_element) <= 0){
            low++;
        }
        if(low >= high) break;
        *high-- = *low;
    }

    *high = part_element;
    return high;
}
```

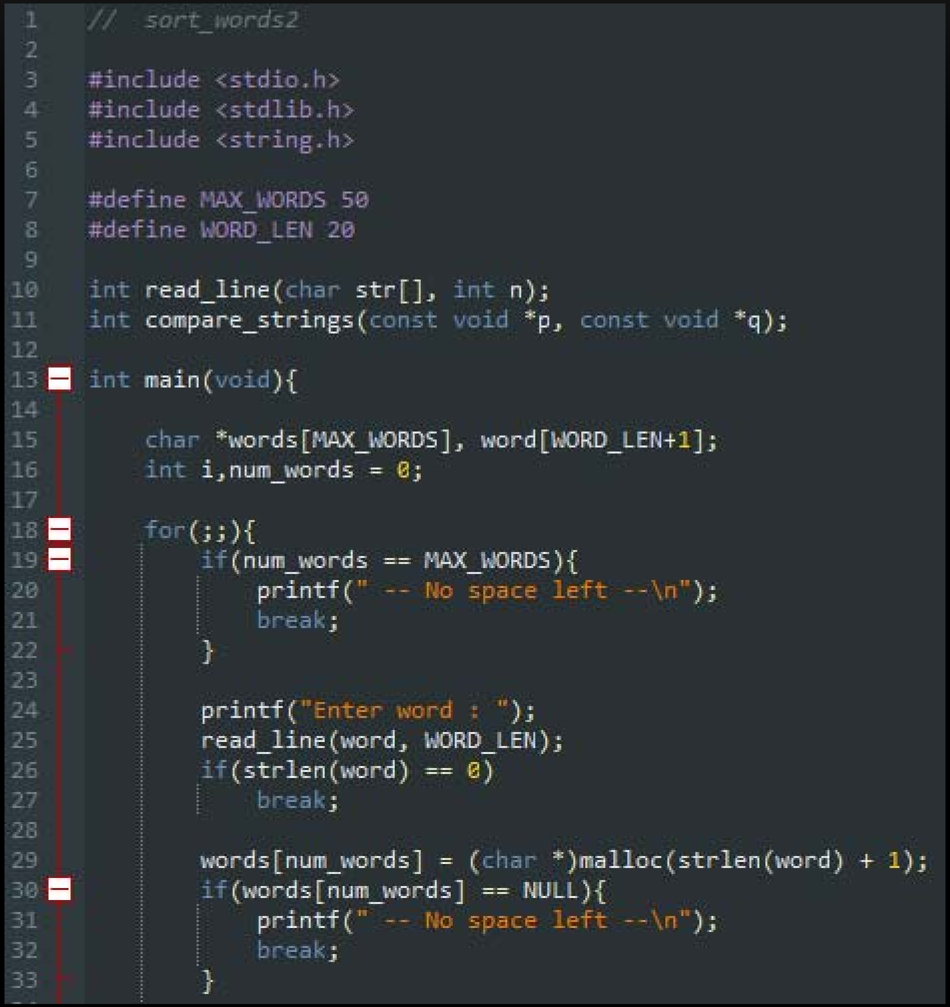
Chapter 17_problem 6

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



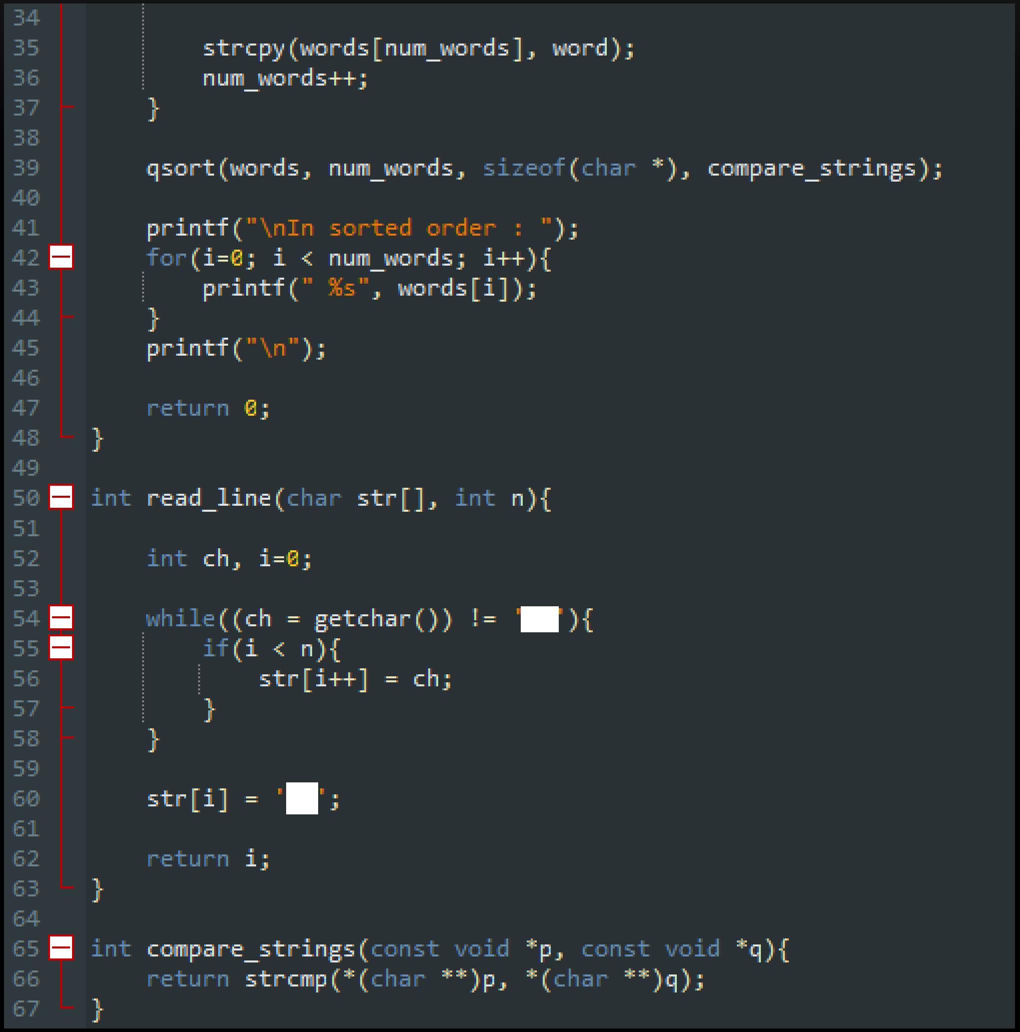
```
Enter word : foo
Enter word : bar
Enter word : baz
Enter word : quux
Enter word :

In sorted order :  bar baz foo quux

--------------------------------
Process exited after 16.74 seconds with return value 0
請按任意鍵繼續 . . .
```

Solution



```c
//  sort_words2

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_WORDS 50
#define WORD_LEN 20

int read_line(char str[], int n);
int compare_strings(const void *p, const void *q);

int main(void){

    char *words[MAX_WORDS], word[WORD_LEN+1];
    int i,num_words = 0;

    for(;;){
        if(num_words == MAX_WORDS){
            printf(" -- No space left --\n");
            break;
        }

        printf("Enter word : ");
        read_line(word, WORD_LEN);
        if(strlen(word) == 0)
            break;

        words[num_words] = (char *)malloc(strlen(word) + 1);
        if(words[num_words] == NULL){
            printf(" -- No space left --\n");
            break;
        }
```

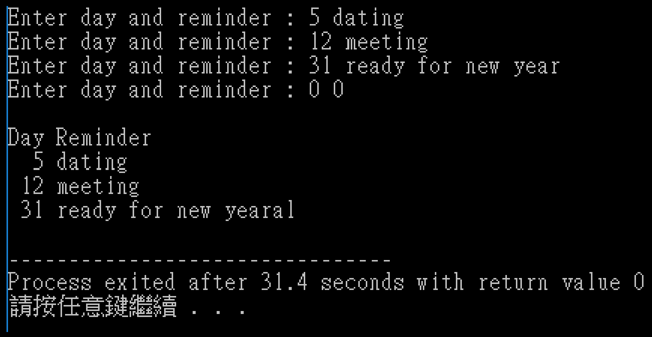
Solution



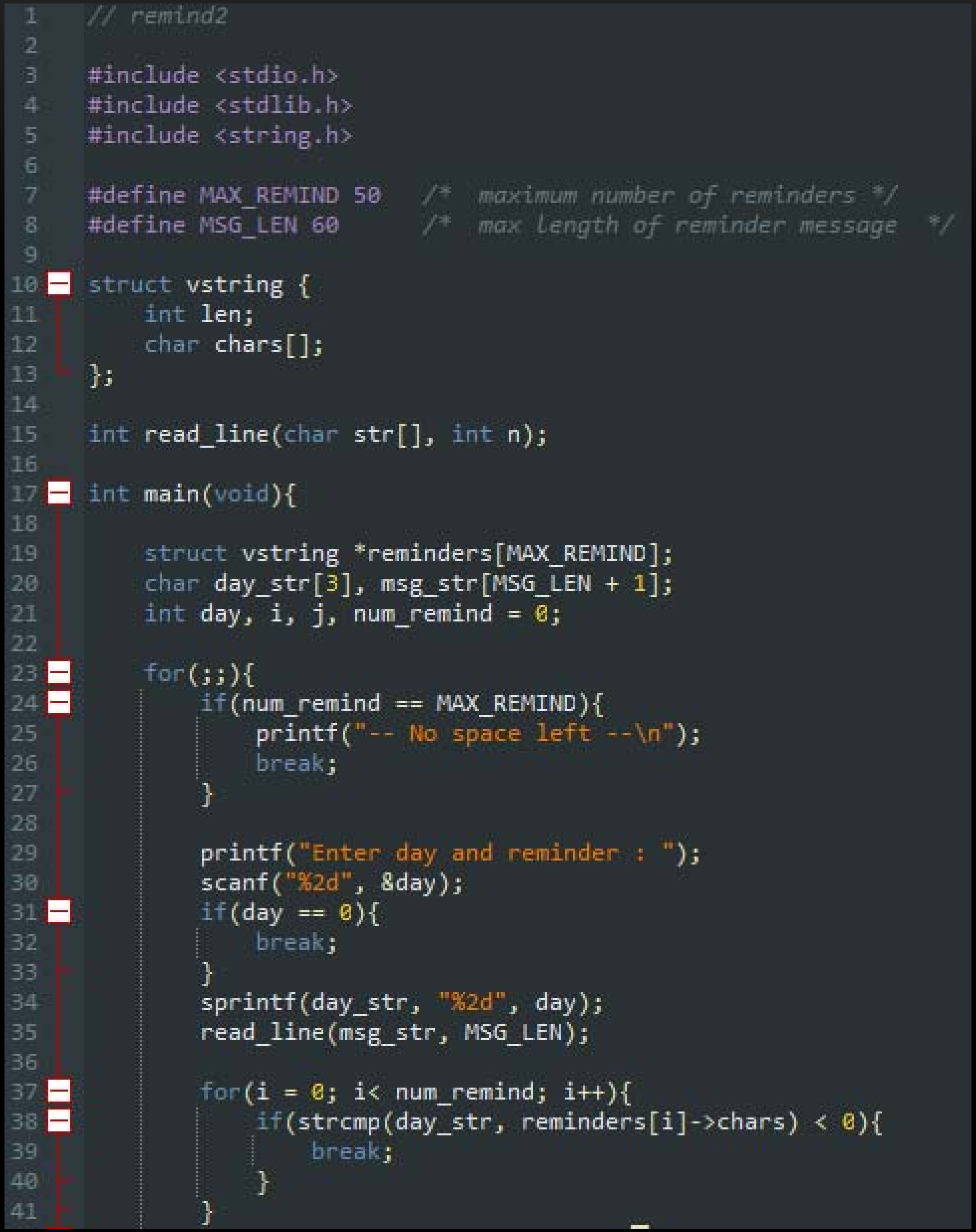
```c

        strcpy(words[num_words], word);
        num_words++;
    }

    qsort(words, num_words, sizeof(char *), compare_strings);

    printf("\nIn sorted order : ");
    for(i=0; i < num_words; i++){
        printf(" %s", words[i]);
    }
    printf("\n");

    return 0;
}

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

    int ch, i=0;

    while((ch = getchar()) != '   '){
        if(i < n){
            str[i++] = ch;
        }
    }

    str[i] = '  ';

    return i;
}

int compare_strings(const void *p, const void *q){
    return strcmp(*(char **)p, *(char **)q);
}
```

Chapter 17_problem 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.

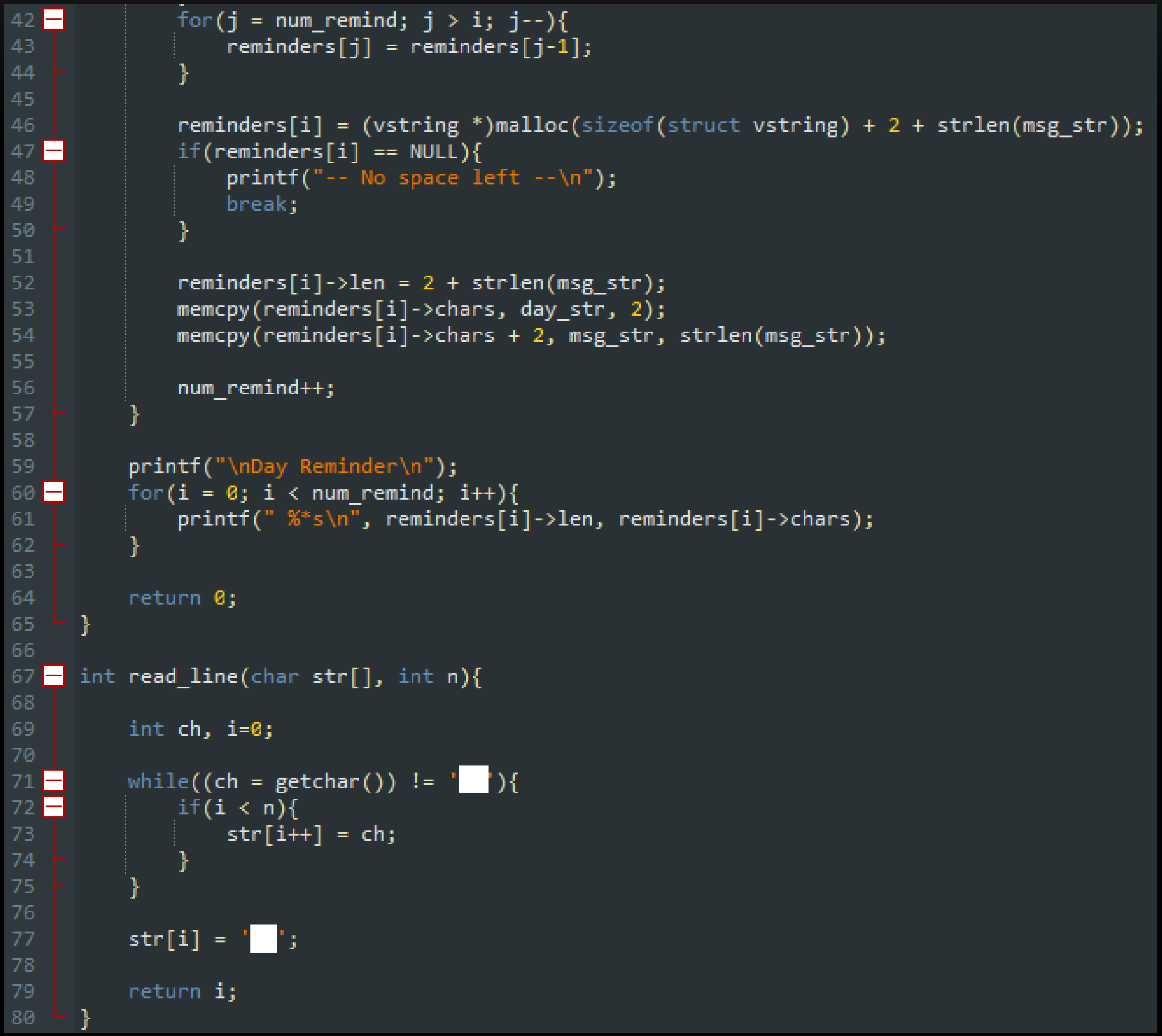

```
Enter day and reminder : 5 dating
Enter day and reminder : 12 meeting
Enter day and reminder : 31 ready for new year
Enter day and reminder : 0 0

Day Reminder
  5 dating
 12 meeting
 31 ready for new yearal

--------------------------------
Process exited after 31.4 seconds with return value 0
請按任意鍵繼續 . . .
```

Solution



```c
// remind2

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_REMIND 50   /*  maximum number of reminders */
#define MSG_LEN 60      /*  max length of reminder message  */

struct vstring {
    int len;
    char chars[];
};

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

int main(void){

    struct vstring *reminders[MAX_REMIND];
    char day_str[3], msg_str[MSG_LEN + 1];
    int day, i, j, num_remind = 0;

    for(;;){
        if(num_remind == MAX_REMIND){
            printf("-- No space left --\n");
            break;
        }

        printf("Enter day and reminder : ");
        scanf("%2d", &day);
        if(day == 0){
            break;
        }
        sprintf(day_str, "%2d", day);
        read_line(msg_str, MSG_LEN);

        for(i = 0; i< num_remind; i++){
            if(strcmp(day_str, reminders[i]->chars) < 0){
                break;
            }
        }
```

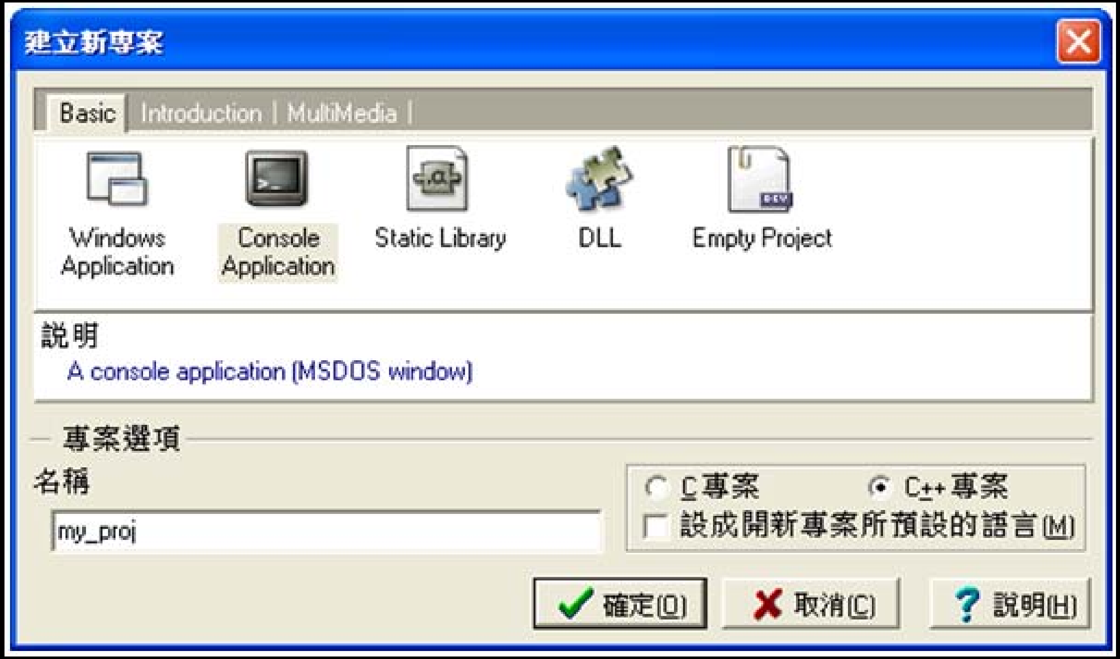
Solution



```c
        for(j = num_remind; j > i; j--){
            reminders[j] = reminders[j-1];
        }

        reminders[i] = (vstring *)malloc(sizeof(struct vstring) + 2 + strlen(msg_str));
        if(reminders[i] == NULL){
            printf("-- No space left --\n");
            break;
        }

        reminders[i]->len = 2 + strlen(msg_str);
        memcpy(reminders[i]->chars, day_str, 2);
        memcpy(reminders[i]->chars + 2, msg_str, strlen(msg_str));

        num_remind++;
    }

    printf("\nDay Reminder\n");
    for(i = 0; i < num_remind; i++){
        printf(" %*s\n", reminders[i]->len, reminders[i]->chars);
    }

    return 0;
}

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

    int ch, i=0;

    while((ch = getchar()) != '  '){
        if(i < n){
            str[i++] = ch;
        }
    }

    str[i] = '  ';

    return i;
}
```

各別編譯的實作

- 下面的步驟介紹如何於Dev C++裡分別建立主程式、函數模組，以及標頭檔
- 步驟1 首先建立一個全新的專案



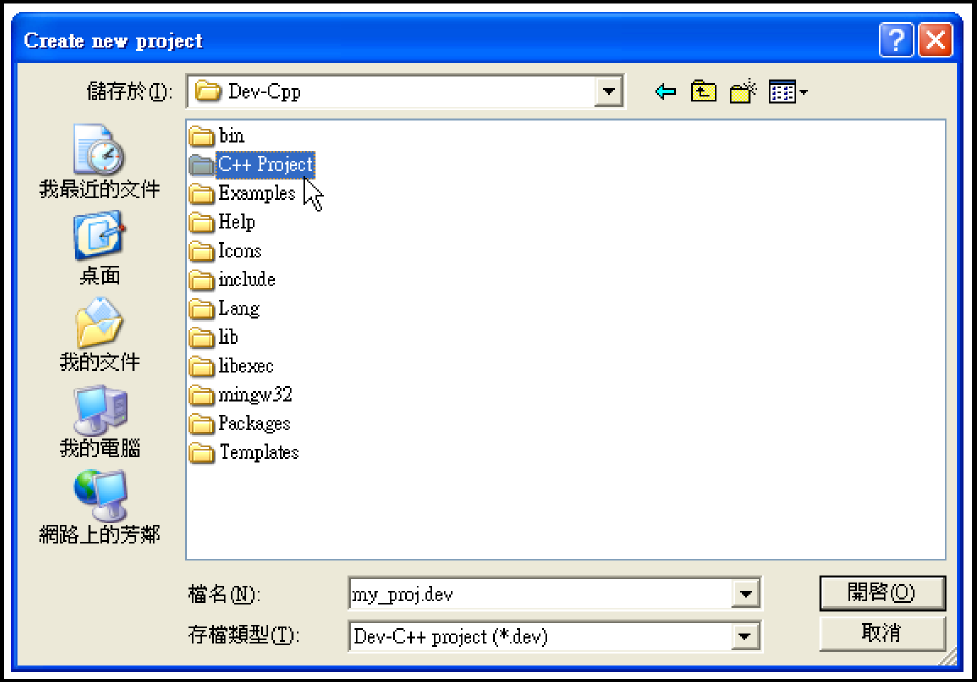
說明
A console application (MSDOS window)

專案選項
名稱: my_proj

C專案 [✓ C++專案]
設成開新專案所預設的語言(M) ☐

各別編譯的實作

- 步驟2 選擇所要存放的資料夾，如下圖所示

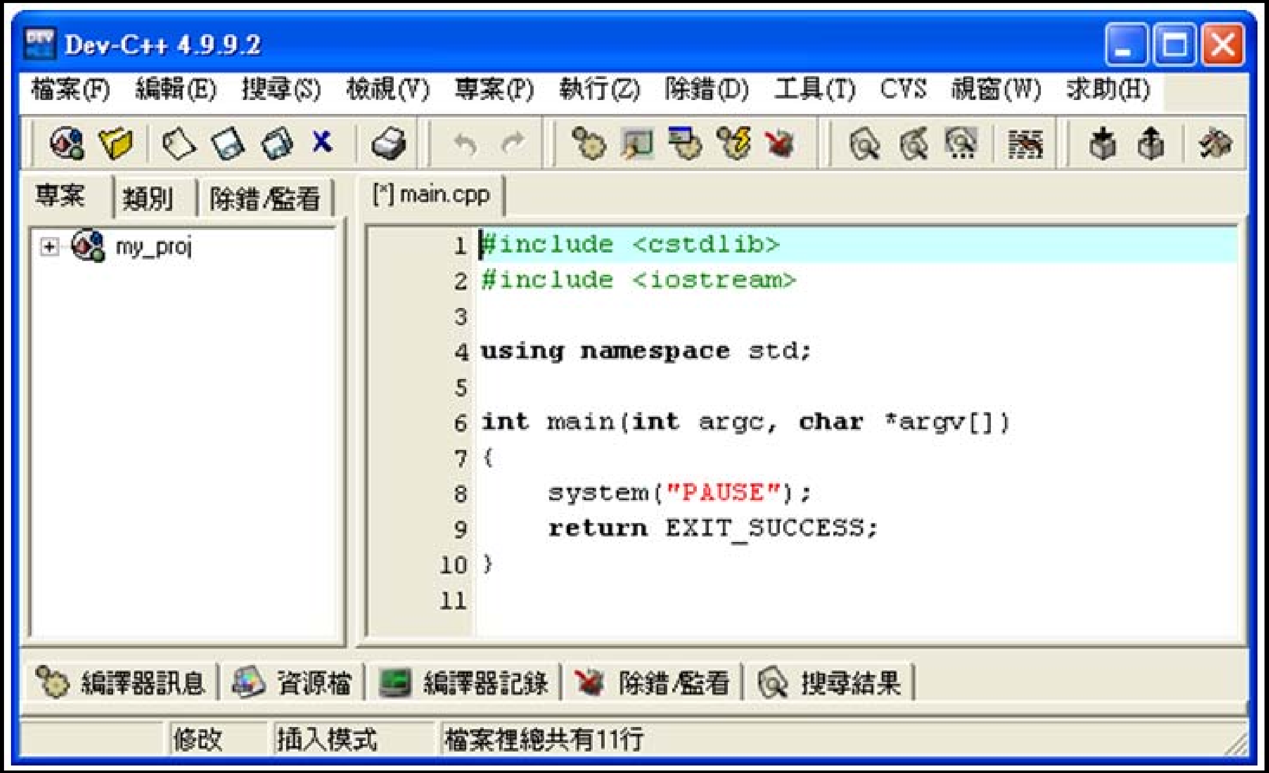


bin
C++ Project
Examples
Help
Icons
include
Lang
lib
libexec
mingw32
Packages
Templates

檔名(N): my_proj.dev
存檔類型(T): Dev-C++ project (*.dev)

各別編譯的實作

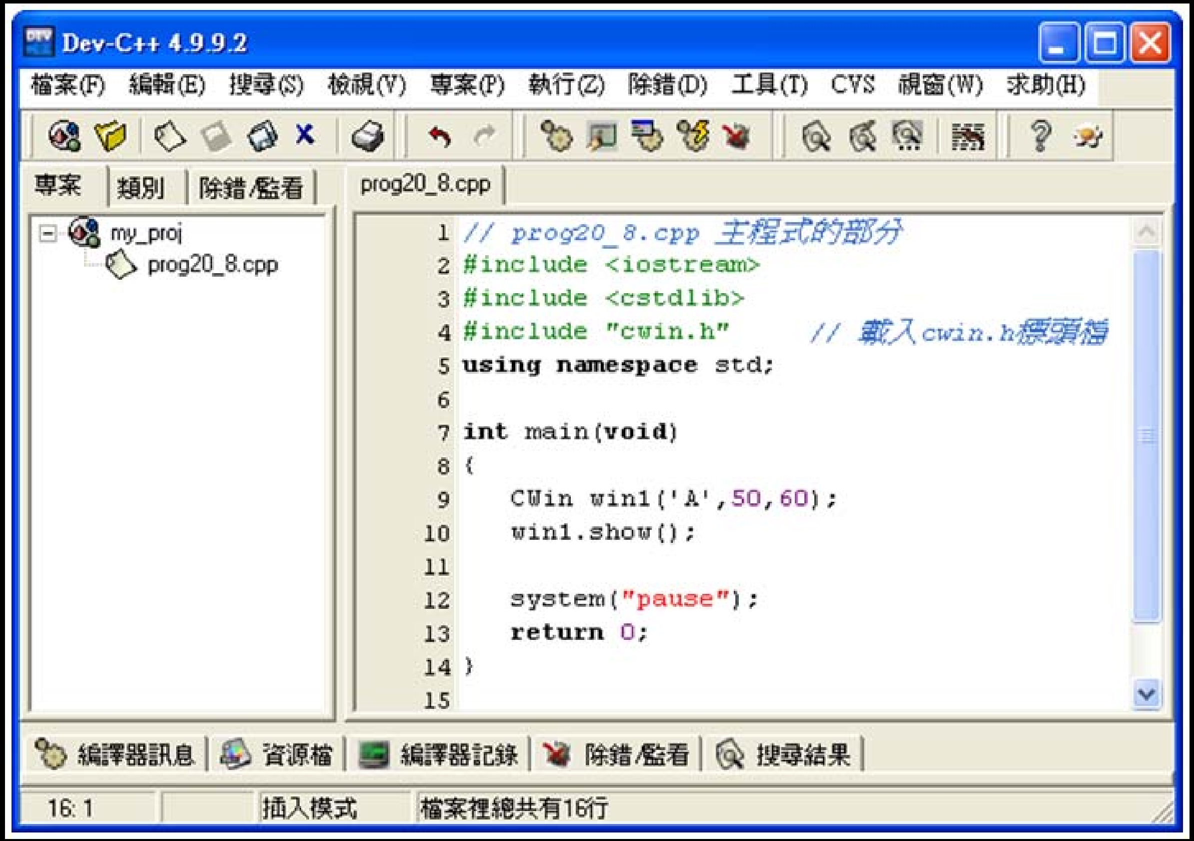
- 步驟3 按下「儲存」鈕後，進入Dev C++的專案開發環境



```
#include <cstdlib>
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    system("PAUSE");
    return EXIT_SUCCESS;
}

```

各別編譯的實作

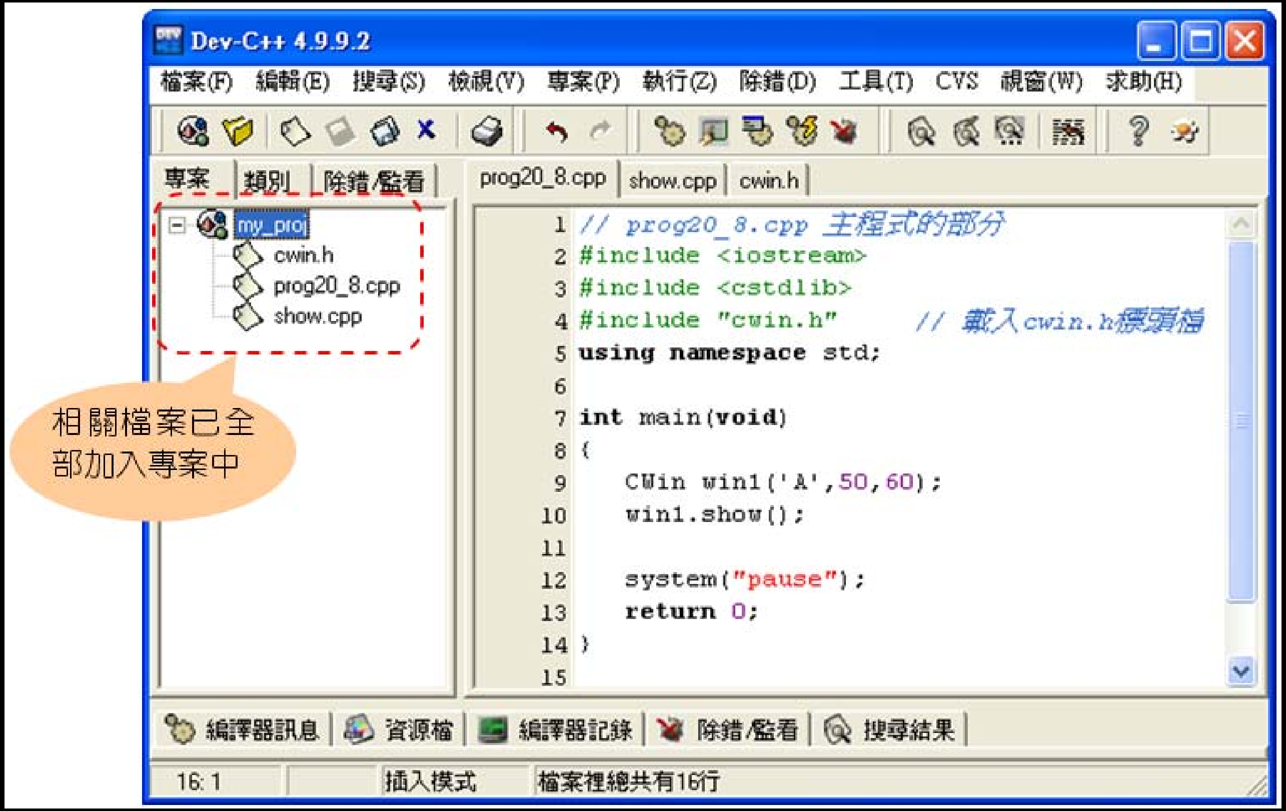
- 步驟4 輸入主程式prog20_8.cpp的內容


Dev-C++ 4.9.9.2
檔案(F) 編輯(E) 搜尋(S) 檢視(V) 專案(P) 執行(Z) 除錯(D) 工具(T) CVS 視窗(W) 求助(H)
專案 類別 除錯/監看
prog20_8.cpp
my_proj
prog20_8.cpp
編譯器訊息 資源檔 編譯器記錄 除錯/監看 搜尋結果
16: 1
插入模式
檔案裡總共有16行

```cpp
// prog20_8.cpp 主程式的部分
#include <iostream>
#include <cstdlib>
#include "cwin.h"       // 載入cwin.h標頭檔
using namespace std;

int main(void)
{
   CWin win1('A',50,60);
   win1.show();

   system("pause");
   return 0;
}

```

各別編譯的實作

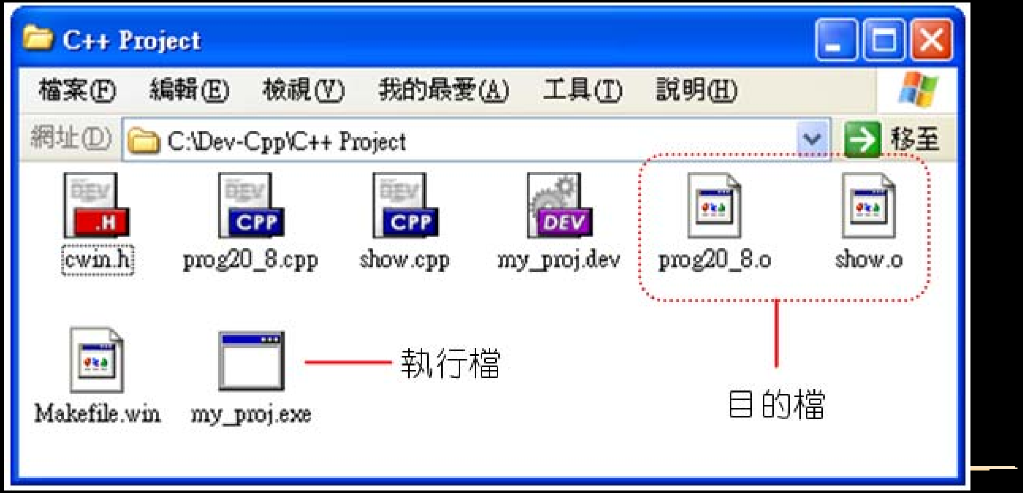
- 步驟5 重複步驟4，將show.cpp與cwin.h加入my_proj中，最後應該會得到如下的視窗：


Dev-C++ 4.9.9.2
檔案(F) 編輯(E) 搜尋(S) 檢視(V) 專案(P) 執行(Z) 除錯(D) 工具(T) CVS 視窗(W) 求助(H)
專案 類別 除錯/監看
prog20_8.cpp show.cpp cwin.h
my_proj
cwin.h
prog20_8.cpp
show.cpp
相關檔案已全部加入專案中
編譯器訊息 資源檔 編譯器記錄 除錯/監看 搜尋結果
16: 1 插入模式 檔案裡總共有16行

```
// prog20_8.cpp 主程式的部分
#include <iostream>
#include <cstdlib>
#include "cwin.h"       // 載入cwin.h標頭檔
using namespace std;

int main(void)
{
   CWin win1('A',50,60);
   win1.show();

   system("pause");
   return 0;
}
```

各別編譯的實作

- 步驟6 按下F9鍵，將程式一起編譯。程式執行的結果如下所示：
- 編譯後的目的檔與執行檔如下圖所示


C++ Project
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
網址(D) C:\Dev-Cpp\C++ Project
移至
cwin.h
prog20_8.cpp
show.cpp
my_proj.dev
prog20_8.o
show.o
Makefile.win
my_proj.exe
執行檔
目的檔