計算機程式語言

助教：蔡詠聿、吳品頤

Chapter 17_project 5

Modify the justify program of Section 15.3 by rewriting the line.c file so that it stores the current line in a linked list. Each node in the list will store a single word. The line array will be replaced by a variable that points to the node containing the first word. This variable will store a null pointer whenever the line is empty.

Chapter 17_project 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

```c
//  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

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

Chapter 17_project 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

```
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
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  int compare_strings(const void *p, const void *q){
66      return strcmp(*(char **)p, *(char **)q);
67  }
```

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.

```
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

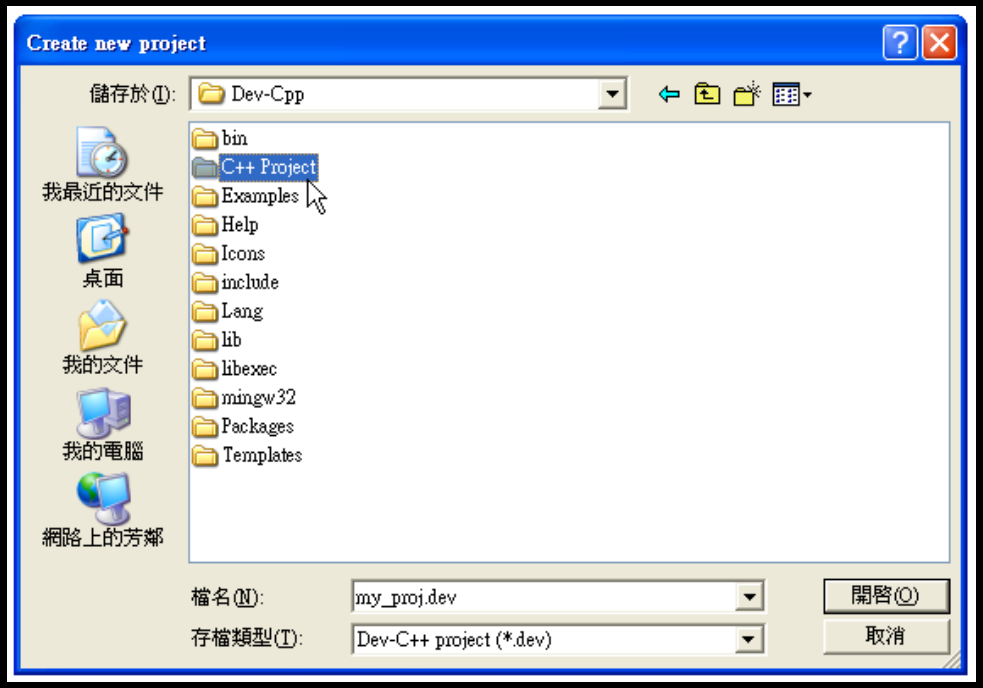
```
42          for(j = num_remind; j > i; j--){
43              reminders[j] = reminders[j-1];
44          }
45
46          reminders[i] = (vstring *)malloc(sizeof(struct vstring) + 2 + strlen(msg_str));
47          if(reminders[i] == NULL){
48              printf("-- No space left --\n");
49              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
69      int ch, i=0;
70
71      while((ch = getchar()) != '\n'){
72          if(i < n){
73              str[i++] = ch;
74          }
75      }
76
77      str[i] = '\0';
78
79      return i;
80  }
```

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

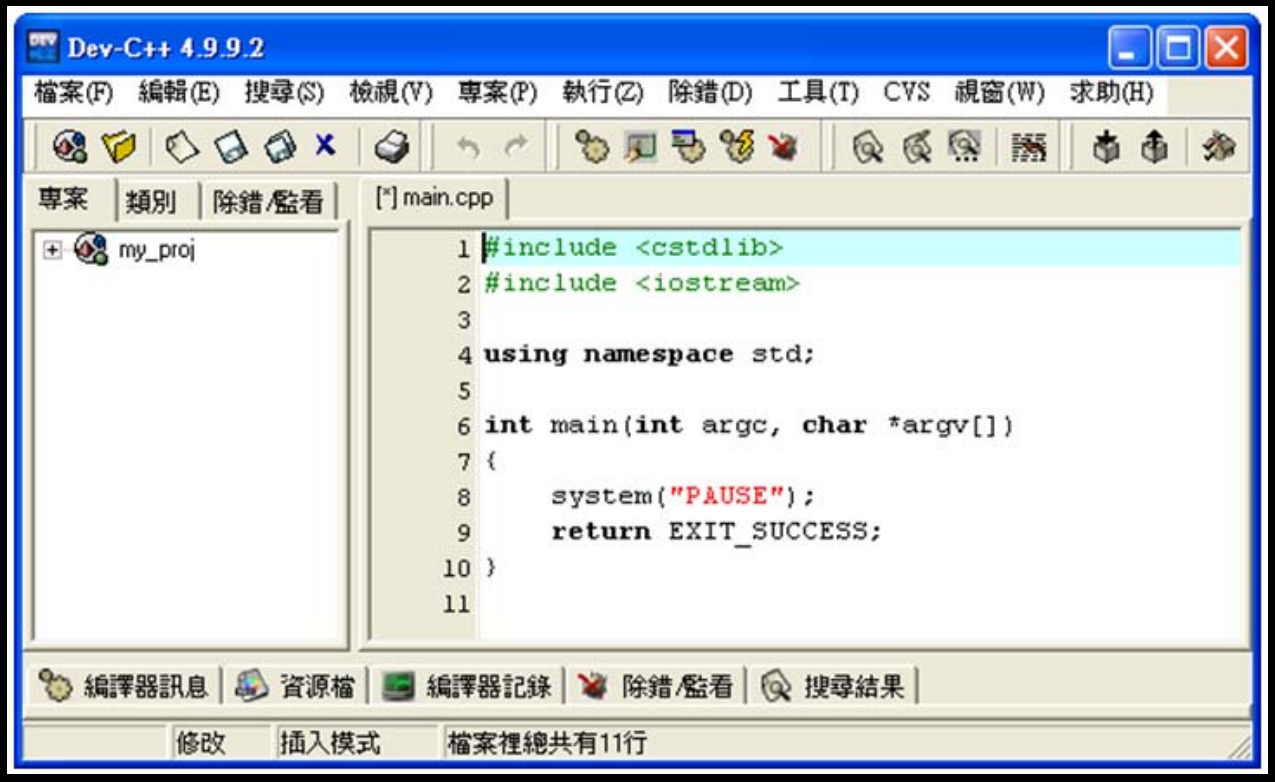
各別編譯的實作

- 步驟2 選擇所要存放的資料夾，如下圖所示



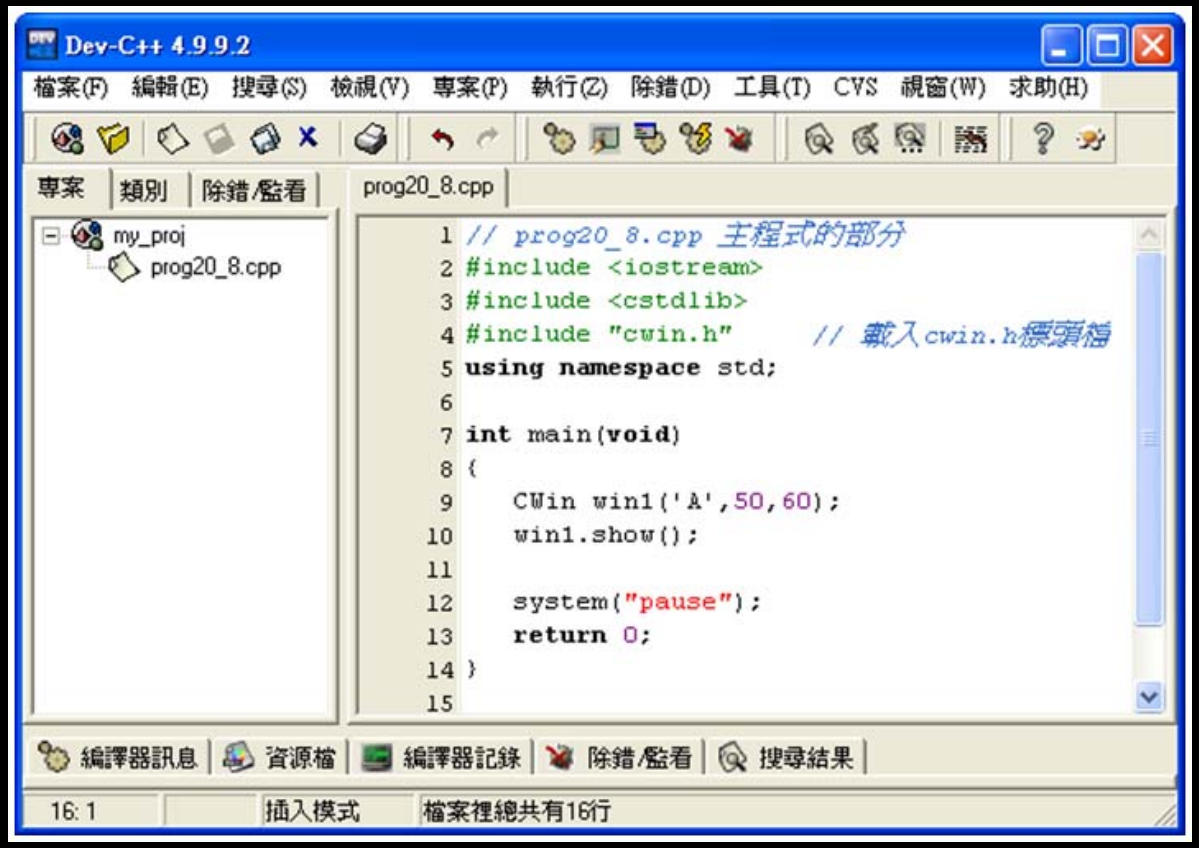
各別編譯的實作

- 步驟3 按下「儲存」鈕後，進入Dev C++的專案開發環境



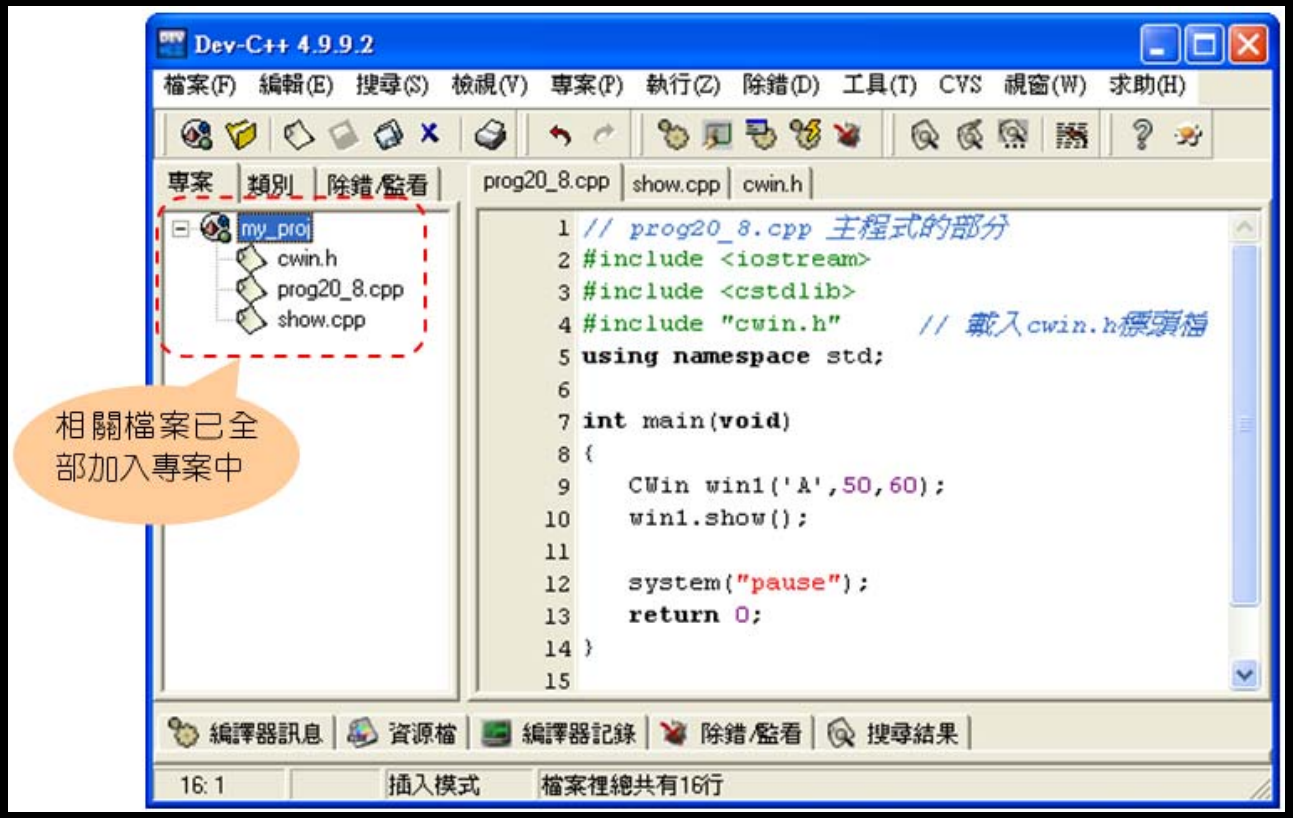
各別編譯的實作

- 步驟4 輸入主程式prog20_8.cpp的內容



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：



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