程式設計

教課教授：謝東儒

助教：李哲源、傅繹嘉

Chapter 12_problem 2

(a) Write a program that reads a message, then checks whether it‘s a palindrome (the letters in the message are the same from left to right as from right to left)

Ignore all characters that aren’t letters. Use integer variables to keep track of positions in the array.

(b) Revise the program to use pointers instead of integers to keep track of positions in the array.

Regardless of uppercase and lowercase

```
Enter a message : Madam, I am Adam.
Not a palindrome

--------------------------------
Process exited after 25.32 seconds with return value 0
請按任意鍵繼續 . . .

Enter a message : He lived as a devil, eh?
Palindrome

--------------------------------
Process exited after 71.75 seconds with return value 0
請按任意鍵繼續 . . .
```

Solution – (a)

```c
//  palindrome_int

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

#define MAX_MSG_LEN 80


int main(void){

    char msg[MAX_MSG_LEN], ch;
    int i, n=0;

    printf("Enter a message : ");
    while(n < MAX_MSG_LEN){
        if((ch = getchar()) == '\n'){
            break;
        }
        if(isalpha(ch)){
            msg[n++] = toupper(ch);
        }
    }

    for(i = 0; i < n/2; i++){
        if(msg[i] != msg[n-i-1]){
            break;
        }
    }

    if(i == n){
        printf("Palindrome\n");
    }else{
        printf("Not a palindrome\n");
    }


    return 0;
}
```

Solution – (b)

```c
// palindrome_ptr

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

#define MAX_MSG_LEN 80


int main(void){

    char msg[MAX_MSG_LEN], ch, *p, *q = &msg[0];

    printf("Enter a message : ");
    while(q < &msg[MAX_MSG_LEN]){
        if((ch = getchar()) == '\n'){
            break;
        }
        if(isalpha(ch)){
            *q++ = toupper(ch);
        }
    }

    for(p = &msg[0], q--; p < q; p++, q--){
        if(*p != *q){
            break;
        }
    }

    if(p >= q){
        printf("Palindrome\n");
    }else{
        printf("Not a palindrome\n");
    }

    return 0;
}
```

Chapter 12_problem 5

Modify Programming Project 14 from Chapter 8 (reverse a sentence) so that it uses a pointer instead of an integer to keep track of the current position in the array that contains the sentence.

```
Enter a sentence : you can cage a swallow can't you?
Reversal of sentence :  you can't swallow a cage can you?

--------------------------------
Process exited after 37.81 seconds with return value 0
請按任意鍵繼續 . . .
```

Solution

```c
// reverse_sentence

#include <stdio.h>

#define MAX_SENTENCE_LEN 80

int main(void){

    char ch, sentence[MAX_SENTENCE_LEN + 1] = {' '}, terminator = '.',
        *start, *finish = sentence + 1, *p = sentence;

    printf("Enter a sentence : ");
    while(finish <= sentence + MAX_SENTENCE_LEN){
        ch = getchar();
        if(ch ==  || ch ==  || ch == ){
            terminator = ch;
            break;
        }
        ++ = ch;
    }

    printf("Reversal of sentence : ");
    for(start = (finish-1); start >= sentence; start--){
        if(*start == ' '){
            for(p = start; p < finish; p++){
                putchar();
            }
            finish = start;
        }
    }
    printf("%c\n", terminator);

    return 0;
}
```

Chapter 12_problem 7

Modify the maxmin.c program of Section 11.4 so that the max_min function uses a pointer instead of an integer to keep track of the current position in the array.

```
Enter 10 numbers : 1 5 9 8 7 4 6 3 2 10
Largest : 10
Smallest : 1

--------------------------------
Process exited after 251.1 seconds with return value 0
請按任意鍵繼續 . . .
```

Solution

```c
// max_ min

#include <stdio.h>

#define N 10

void max_min(int a[], int n, int *max, int *min);

int main(void){

    int b[N], i, big, small;

    printf("Enter %d numbers : ", N);
    for(i = 0; i < N; i++){
        scanf("%d", &b[i]);
    }

    max_min(b, N, [redacted], [redacted]);

    printf("Largest : %d\n", big);
    printf("Smallest : %d\n", small);

    return 0;
}

void max_min(int a[], int n, int *max, int *min){

    int *p;

    *max = *min = *a;
    for(p = a; p < a + n; p++){
        if(*p > *max){
            [redacted] = *p;
        }else if(*p < *min){
            [redacted] = *p;
        }
    }
}
```