計算機程式語言

教課教授 : 謝東儒

助教 :蔡詠聿、吳品頤

Chapter 12_project 1

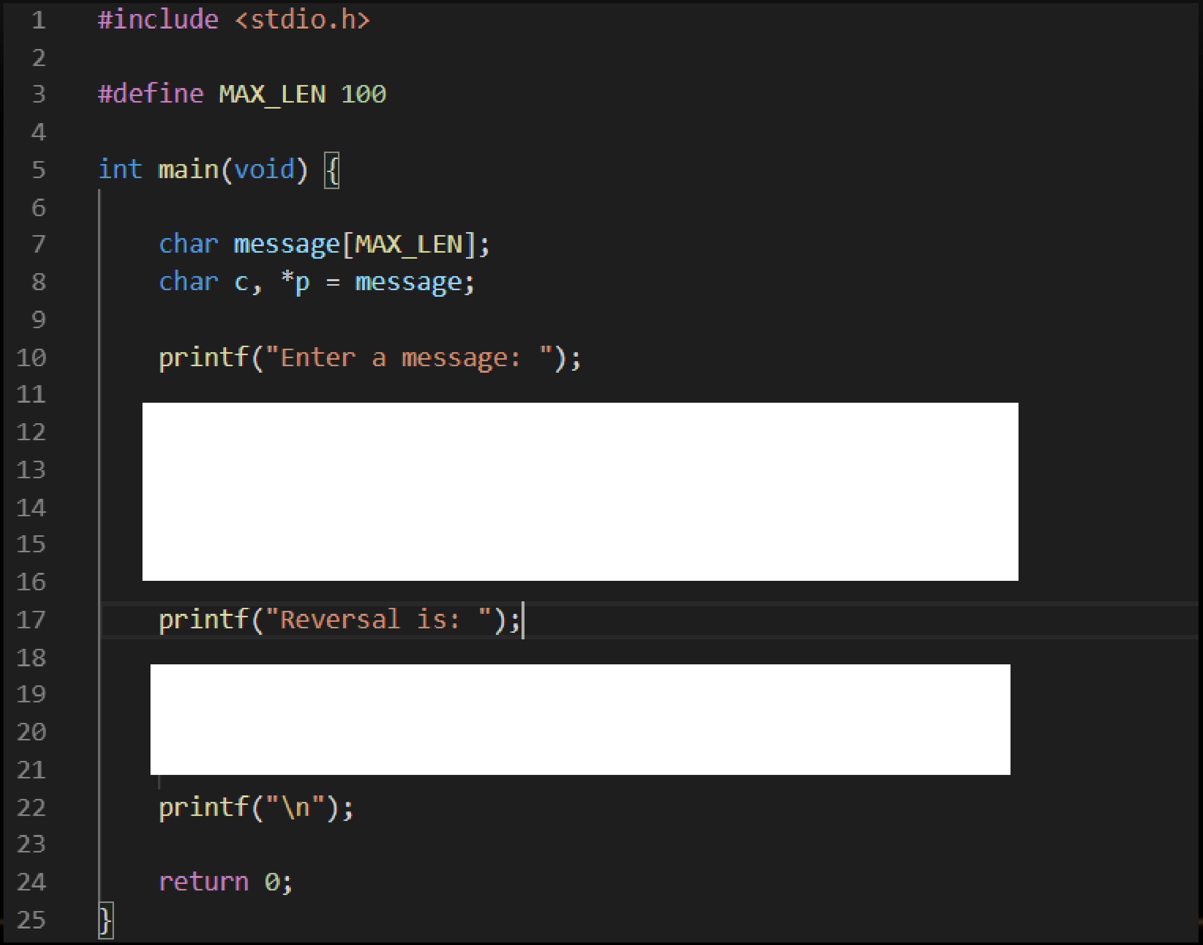
Write a program that reads a message, then prints the reversal of the message:

```
Enter a message: Don’t get mad, get even.
Reversal is: .neve teg ,dam teg t’noD
```

Hint:Read the message one character at a time(using getchar) and store the characters in an array. Stop reading when the array is full or the character read is ‘\n’.

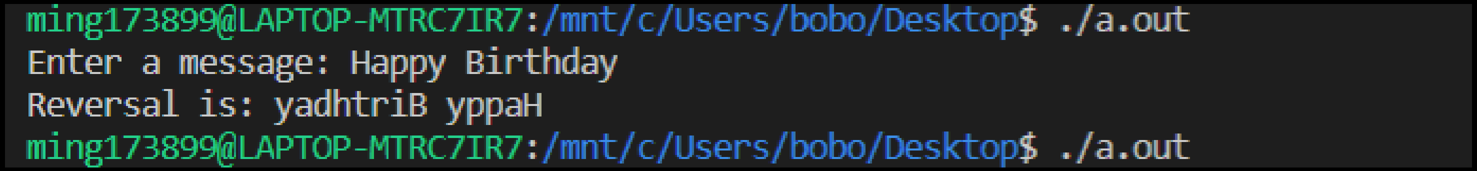
Revise the program to use a pointer instead of an integer to keep track of the current position in the array.

Solution



```c
#include <stdio.h>

#define MAX_LEN 100

int main(void) {

    char message[MAX_LEN];
    char c, *p = message;

    printf("Enter a message: ");

    [redacted]

    printf("Reversal is: ");

    [redacted]

    printf("\n");

    return 0;
}
```

Example



```
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
Enter a message: Happy Birthday
Reversal is: yadhtriB yppaH
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
```

Chapter 12_project 2

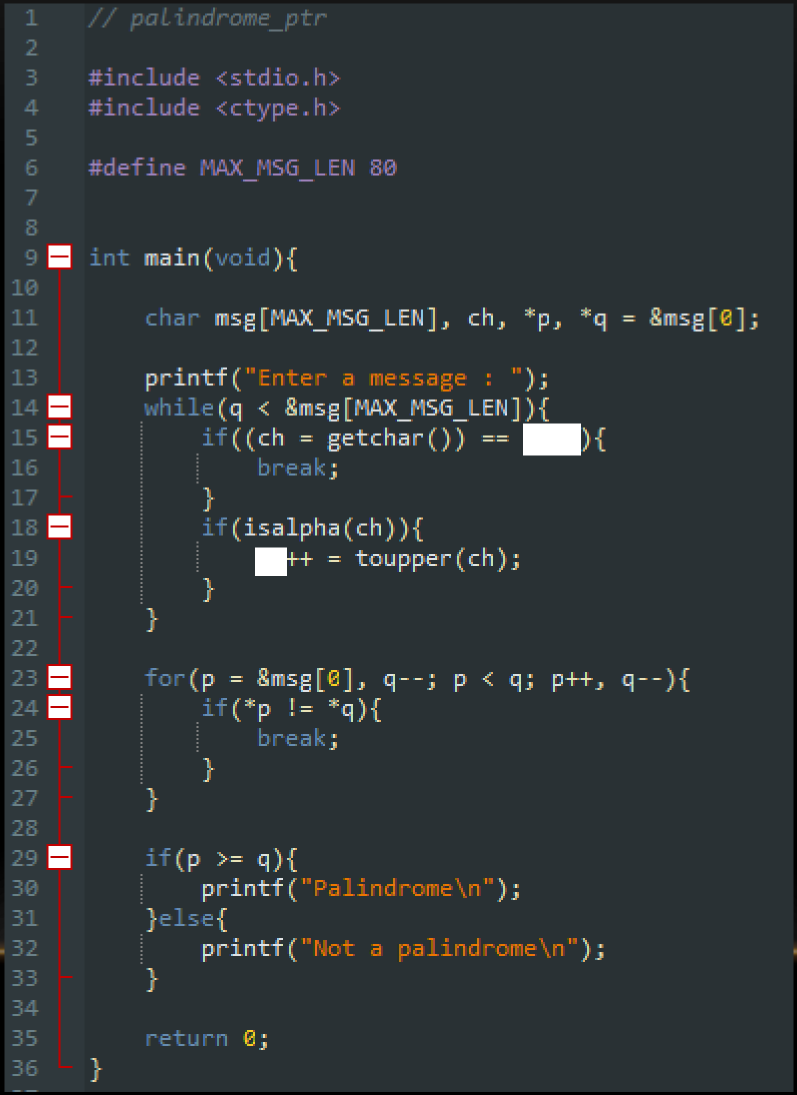
Write a program that reads a message, then checks whether it‘s a palindrome (the letters in the message are the same from left to right as from right to left)

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

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

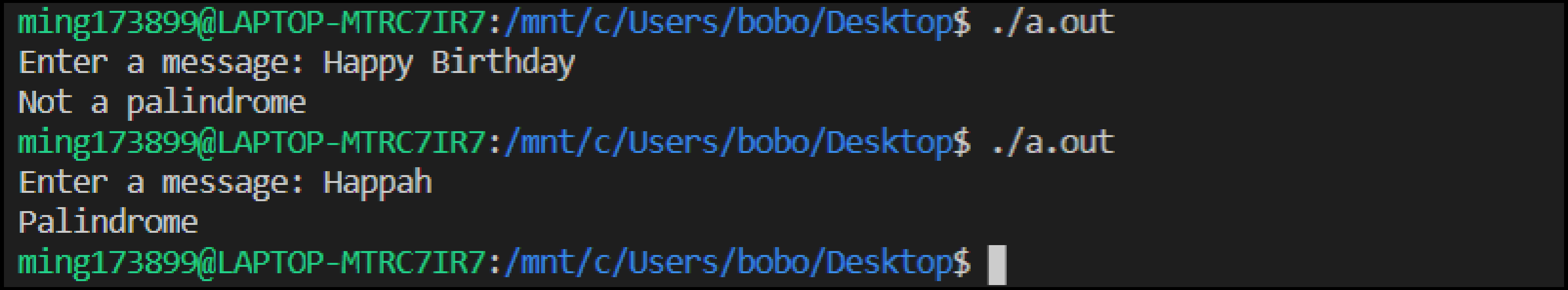
Regardless of uppercase and lowercase

Solution



```
// 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()) ==     ){
            break;
        }
        if(isalpha(ch)){
              ++ = 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;
}
```

Example

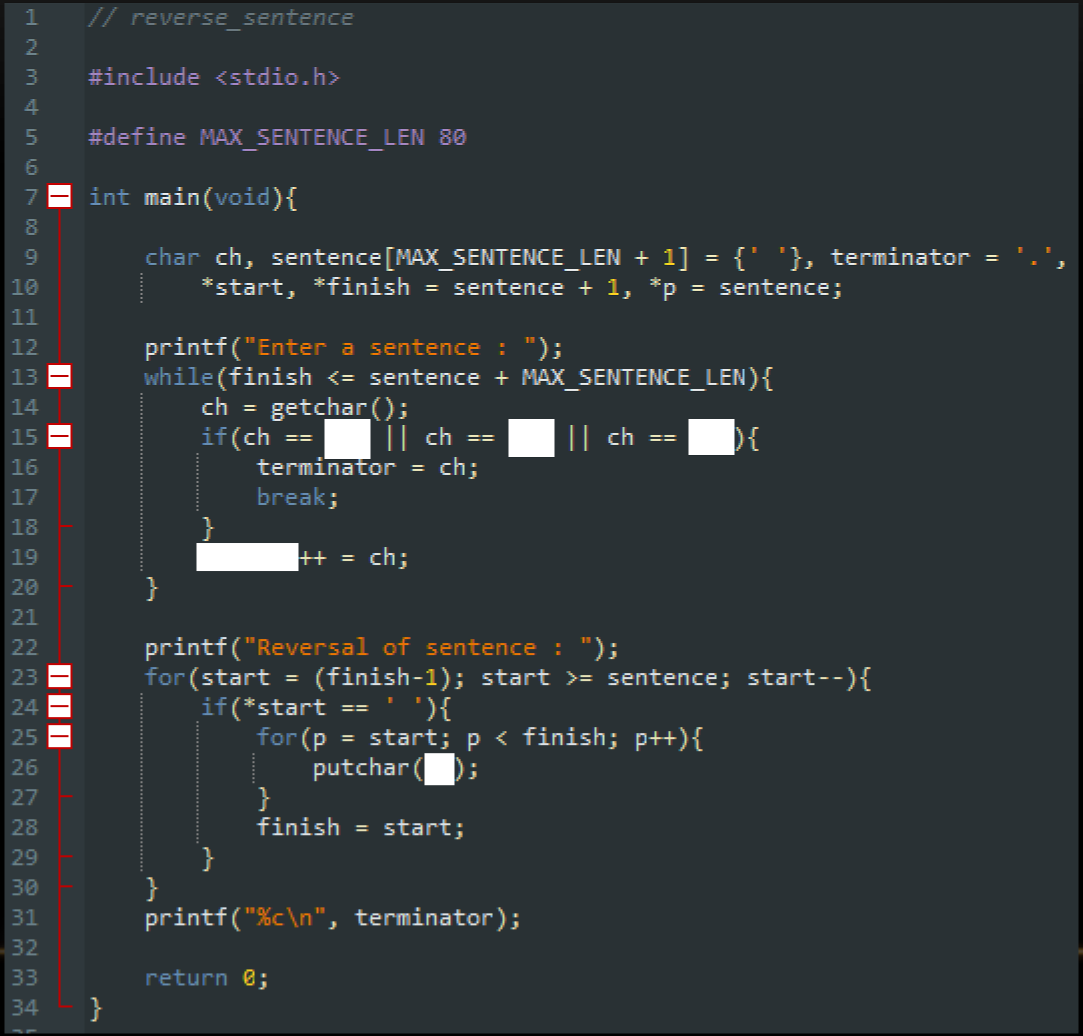


```
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
Enter a message: Happy Birthday
Not a palindrome
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
Enter a message: Happah
Palindrome
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$
```

Chapter 12_project 5

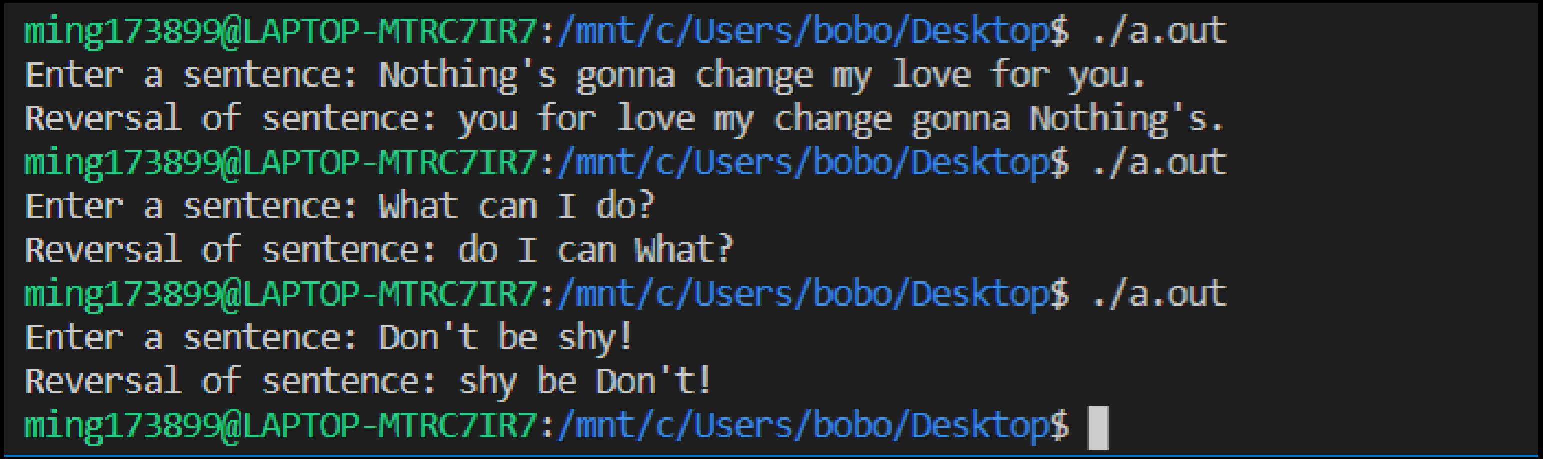
Modify Programming Project 14 from Chapter 8 (reverse a sentence) so that it uses a pointer instead of an integer to keep track of the current position in the array that contains the sentence.

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;
}
```

Example

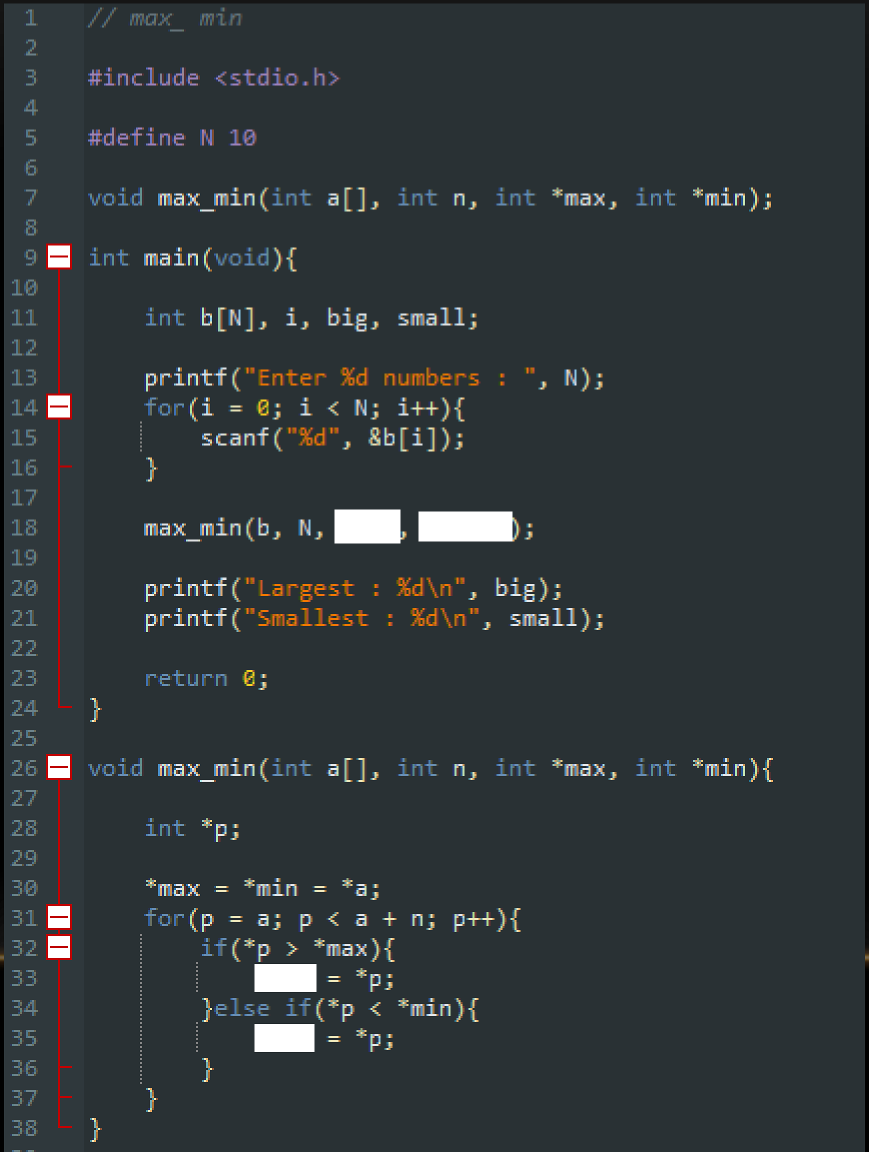


```
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
Enter a sentence: Nothing's gonna change my love for you.
Reversal of sentence: you for love my change gonna Nothing's.
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
Enter a sentence: What can I do?
Reversal of sentence: do I can What?
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
Enter a sentence: Don't be shy!
Reversal of sentence: shy be Don't!
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$
```

Chapter 12_project 7

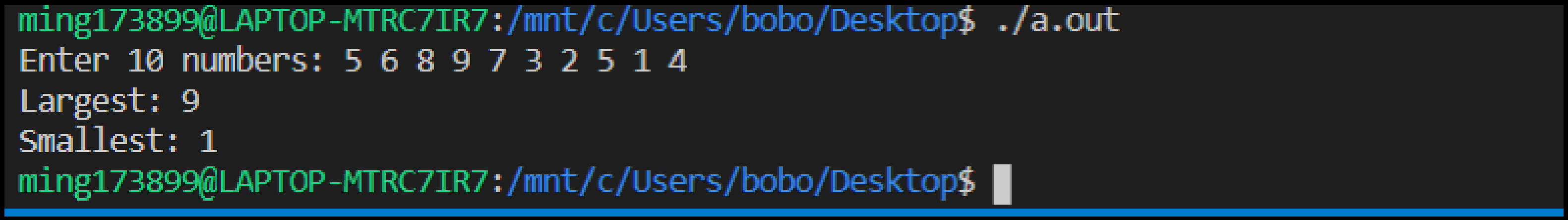
Modify the maxmin.c program of Section 11.4 so that the max_min function uses a pointer instead of an integer to keep track of the current position in the array.

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, ____, ____);

    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){
            ____ = *p;
        }else if(*p < *min){
            ____ = *p;
        }
    }
}
```

Example



```
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$ ./a.out
Enter 10 numbers: 5 6 8 9 7 3 2 5 1 4
Largest: 9
Smallest: 1
ming173899@LAPTOP-MTRC7IR7:/mnt/c/Users/bobo/Desktop$
```