中南大學

CENTRAL SOUTH UNIVERSITY

計算機程序設計基礎（C）

實驗報告

學生姓名　許采婷

學　　號　8208230832

專業班級　計通 2308 班

指導教師　杨迎泽

學　　院　計算機學院

实验五 链表实验

1. 实验目的要求

（1）了解指针与数组的深刻联系，并学会利用指针解决问题。

（2）了解处理字符串的方法，C 语言库里面的处理字符串的函数。

（3）了解结构、联合、枚举等数据结构的定义。

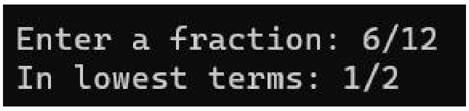
（4）掌握链表的生成，链表数据的输出、删除和插入及链表应用。

2. 实验内容

2.1 教材第 11 章编程题第 3 题：用指针变量编写简化一个分数的函数。

说明：本题用来解决函数无法传回传递的实际参数，指针变量。

（1）实验运行结果：



```
Enter a fraction: 6/12
In lowest terms: 1/2
```

（2）源代码：

```
    // 用指针变量编写简化一个分数的函数

    #include <stdio.h>

    int gcd(int a, int b){

        int remainder;

        while(b!=0){

            remainder = a%b;
            a = b;
            b = remainder;
        }

        return a;
    }

    void reduce(int numerator, int denominator, int 
*reduced_numerator, int *reduced_denominator){

        int common_divisor = gcd(numerator, denominator);
        *reduced_numerator = numerator/common_divisor;
        *reduced_denominator = denominator/common_divisor;
    }

    int main(){

        int numerator, denominator, reduced_numerator, 
reduced_denominator;
```

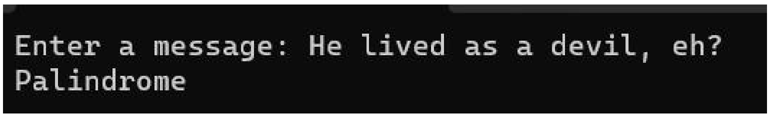
```
        printf("Enter a fraction: ");
        scanf("%d/%d", &numerator, &denominator);

        reduce(numerator, denominator, &reduced_numerator,
&reduced_denominator);

        printf("In lowest terms: %d/%d\n", reduced_numerator,
reduced_denominator);
        return 0;
    }
```

2.2　教材第 12 章编程题第 2 题：编写函数读入用户输入的一条消息，检测是否是回文，分别用数组、指针来解决。

说明：本题可以体会到数组与指针的特点，利用指针的便利。

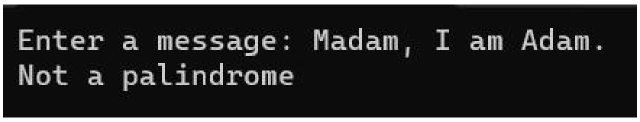
（1）实验运行结果：

① 數組



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

② 指針



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

（2）源代码：

① 數組

```
// 读入用户输入的一条消息，检测是否是回文，分别用数组、指针来解决。
//数组
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#define MAX_LENGTH 100

int main() {

    char message[MAX_LENGTH];

    printf("Enter a message: ");

    for(int i=0; i<MAX_LENGTH; i++){

        message[i] = getchar();

        if(message[i] == '\n')
            break;
    }
    for (int i=0, j=strlen(message)-1; i<j; i++, j--){

        while(!isalpha(message[i])){

            i++;
        }
        while(!isalpha(message[j])){

            j--;
        }

        if(tolower(message[i])!=tolower(message[j])){

            printf("Not a palindrome\n");
            return 0;
        }
    }

    printf("Palindrome\n");
    return 0;
}
```

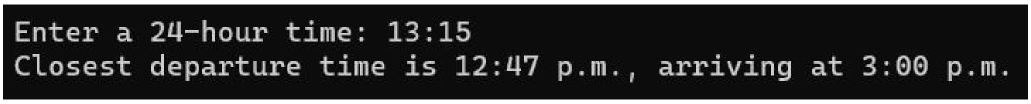
② 指針

```
// 读入用户输入的一条消息，检测是否是回文，分别用数组、指针来解决。
//指针
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#define MAX_LENGTH 100

int is_letter(char c) {

    return isalpha(c);
}

int is_palindrome(char *message) {

    char *p, *q;
    for (p=message, q=message+strlen(message)-1; p<q; p++, q--){

        while(!is_letter(*p) && p<q){

            p++;
        }
        while(!is_letter(*q) && p<q) {
            q--;
        }
        if(tolower(*p)!=tolower(*q)) {
            return 0;
        }
    }

    return 1;
}

int main(){

    char message[MAX_LENGTH];

    printf("Enter a message: ");
    fgets(message, MAX_LENGTH, stdin);

    message[strlen(message)-1] = '\0';

    if (is_palindrome(message)) {
        printf("Palindrome\n");
    }
    else {
        printf("Not a palindrome\n");
    }

    return 0;
}
```

2.3 教材第 16 章编程题第 5 题：用结构变量解决搜索离用户输入时间最近的航班信息，并输出。

说明：本题考查结构与数组的结合运用，有助于理解结构。首先，定义一个存储起飞、到达时间的结构，再定义一个该类型的数组，循环搜索即可。

（1）实验运行结果：



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

（2）源代码：

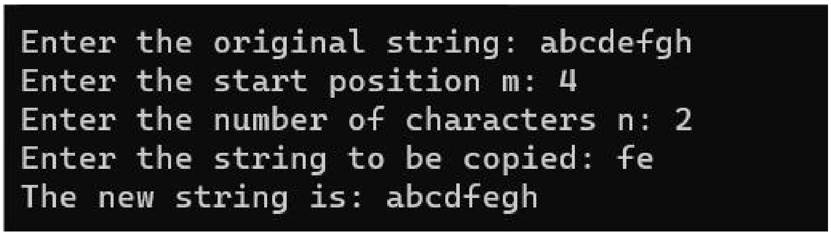
```
// 搜索离用户输入时间最近的航班信息，并输出

#include <stdio.h>

struct FlightTime {
    int departure;
    int arriving;
} flight[] = {
    {0, 0}, {480, 616}, {583, 712},
    {679, 811}, {767, 900}, {840, 968},
    {945, 1075}, {1140, 1280}, {1305, 1438}
};

int main() {

    int hour, minute;
    int input_minutes;

    printf("Enter a 24-hour time: ");
    scanf("%d:%d", &hour, &minute);
```

```
        input_minutes = hour*60 + minute;

        for(int i=0; i<8; i++){

            if(input_minutes>flight[i].departure &&
               input_minutes<flight[i + 1].departure)
                break;
        }

        if(input_minutes-flight[i].departure < flight[i + 1].departure-
input_minutes){

            printf("Closest departure time is %d:%02d ",
                   (flight[i].departure%720)/60==0? 12 :
(flight[i].departure % 720)/60,
                   flight[i].departure%60);

            printf("%s", flight[i].departure<720? "a.m." : "p.m.");

            printf(", arriving at %d:%02d ",
                   (flight[i].arriving%720)/60==0? 12 :
(flight[i].arriving%720)/60,
                   flight[i].arriving%60);

            printf("%s\n", flight[i].arriving<720? "a.m." : "p.m.");
        }
        else{

            printf("Closest departure time is %d:%02d ",
                (flight[i+1].departure%720)/60==0? 12 :
(flight[i+1].departure%720)/60,
                flight[i+1].departure%60);

            printf("%s", flight[i+1].departure<720? "a.m." : "p.m.");

            printf(", arriving at %d:%02d ",
                (flight[i+1].arriving%720)/60==0? 12 :
(flight[i+1].arriving%720)/60,
                flight[i+1].arriving%60);

            printf("%s\n", flight[i+1].arriving<720? "a.m." : "p.m.");
        }

        return 0;
    }
```

2.4 利用指针编写程序，实现从键盘键入字符串，将此字符串从 m 个字符开始的 n 个字符复制成另一个字符串。

说明：本题用指针指向字符串，进行跟踪和复制操作。

（1）实验运行结果：



```
Enter the original string: abcdefgh
Enter the start position m: 4
Enter the number of characters n: 2
Enter the string to be copied: fe
The new string is: abcdfegh
```

（2）源代码：

```
// 键入字符串，将此字符串从 m 个字符开始的 n 个字符复制成另一个字符串

#include <stdio.h>
#include <string.h>
#define MAX_LENGTH 100

int main(){

    char str1[MAX_LENGTH], str2[MAX_LENGTH];
    int m, n;
    char *p1, *p2;

    printf("Enter the original string: ");
    fgets(str1, MAX_LENGTH, stdin);

    printf("Enter the start position m: ");
    scanf("%d", &m);

    printf("Enter the number of characters n: ");
    scanf("%d", &n);

    printf("Enter the string to be copied: ");
    scanf("%s", str2);

    if(m<0 || m>=strlen(str1) || n<=0 || n>strlen(str1)-m){

        printf("Invalid input\n");
        return 0;
    }

    p1 = str1 + m;
    p2 = str2;

    while(n-- && *p2!='\0'){

        *p1++ = *p2++;
    }

    printf("The new string is: %s\n", str1);
```

```
        return 0;
    }
```