



# 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 楊迎澤

學 院 計算機學院

## 实验三: 数组实验

### 1. 实验目的要求

- (1) 掌握整型、浮点型，以及其他一些基本数据类型。
- (2) 掌握数组的定义和使用，掌握常见使用数组的算法，如排序算法和查找算法等。
- (3) 学会将数据类型和数组熟练运用到之前的一些循环、选择结构中。

### 2. 实验内容

2.1 教材第 7 章编程题第 12 题：编程计算从左至右求值的表达式的值。

说明：本题考查对运算符和操作数的判断。

(1) 实验运行结果：

```
Enter an expression: 1+2.5*3
Value of expression: 10.5
```

(2) 源代码：

```
// 表达式由左而右求值

#include<stdio.h>

int main(){

    double A, B;
    double sum = 0;
    char sign; // 运算符符号

    printf("Enter an expression: ");
    scanf("%lf", &A);

    sum = A;

    while((sign=getchar())!='\n'){ // 若不是回车，读取一组符号及数字

        scanf("%lf", &B);

        if(sign=='+')    sum = sum + B;
        else if(sign=='-')    sum = sum - B;
        else if(sign=='*')    sum = sum * B;
        else if(sign=='/')    sum = sum / B;
    }

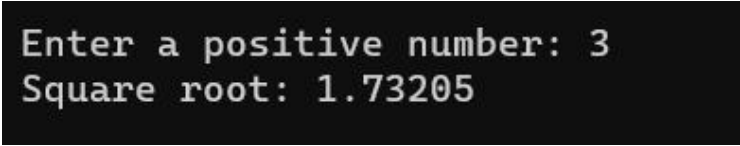
    printf("Value of expression: %.1f\n", sum);

    return 0;
}
```

2.2 教材第 7 章编程题第 14 题：编程序用牛顿算法计算浮点数平方根。

说明：本题考查对循环结构的掌握，对表达式的合理运用，对基本类型的理解。

(1) 实验运行结果：



```
Enter a positive number: 3
Square root: 1.73205
```

(2) 源代码：

```
// 牛頓方法計算平方根

#include<stdio.h>
#include<math.h>

int main(){
    double x, y, tmp_y;

    printf("Enter a positive number: ");
    scanf("%lf", &x);

    y = x/2;
    tmp_y = y;

    y = (x/y + y)/2;

    while(fabs(y-tmp_y)>=0.00001){
        tmp_y = y;
        y = (x/y + y)/2;
    }

    printf("Square root: %.5f\n", y);

    return 0;
}
```

### 2.3 教材第 8 章编程题第 14 题：编写程序颠倒句子中单词的顺序。

说明：本题考查对字符数组的处理，对一些循环选择结构的理解。

(1) 实验运行结果：

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

(2) 源代码：

```
// 顛倒句子單詞順序

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

int main(){

    char sentence[1000000] = {0};
    char sign; // 最後的標點符號
    int flag; // 單詞的最後一個字母
    int len = 0;

    printf("Enter a sentence: "); // 輸入句子
    while((sentence[len]=getchar())!='\n')    len++;

    sign = sentence[len-1]; // 存標點符號
    flag = len-2; // 句子最後一個單詞的最後一個字母

    printf("Reversal of sentence: "); // 開始顛倒
    while(len>=0){
        if(sentence[len]!=' '){ // 還沒讀完一個單詞

            len--;
            continue;
        }
        else{

            for(int j=len+1; j<=flag; j++){ // 單詞第一個字母到最後一個

                printf("%c", sentence[j]);

            }

            printf(" ");
            len--;
            flag = len; // 標記新的單詞的最後一個字母
        }
    }

    for(int j=0; j<=flag; j++) // 第一組單詞還未輸出
        printf("%c", sentence[j]);

    printf("%c\n", sign); // 輸出標點符號

    return 0;
}
```

## 2.4 教材第 8 章编程题第 15 题：编写程序实现凯撒密码加密。

说明：本题考查对字符数组的掌握，对字符处理的理解。

(1) 实验运行结果：

```
Enter message to be encrypted: Go ahead, make my day.
Enter shift amount (1-25): 3
Encrypted message: Jr dkhdg, pdnh pb gdb.
```

(2) 源代码：

```
// 凱薩密碼

#include<stdio.h>

int main(){

    char word[100000] = {0};
    int shift = 0; // 移動距離
    int len = 0;

    printf("Enter message to be encrypted: "); // 輸入明文
    while((word[len]=getchar())!='\n')
        len++;

    printf("Enter shift amount (1-25): ");
    scanf("%d", &shift);

    printf("Encrypted message: ");
    for(int i=0; i<len; i++){

        word[i] = word[i] + shift;

        if(word[i]>='A' && word[i]<='Z'){ // 大寫

            if(word[i]>'Z')
                word[i] = word[i] - 26;
        }
        else if(word[i]>='a' && word[i]<='z'){ // 小寫

            if(word[i]>'z')
                word[i] = word[i] - 26;
        }

        printf("%c", word[i]);
    }

    printf("\n");

    return 0;
}
```