

Assignment Chapter 7

张理凯 202300101051

2.

// 求方程 $ax^2+bx+c=0$ 的根，用三个函数分别求当： $b^2-4ac>0$ ， $b^2-4ac=0$ ， $b^2-4ac<0$ 时的根并输出结果。从主函数输入a,b,c的值。

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

void positive(double a, double b, double c) {
    double discriminant = b * b - 4 * a * c;
    double root1 = (-b + sqrt(discriminant)) / (2 * a);
    double root2 = (-b - sqrt(discriminant)) / (2 * a);
    printf("两个不同的实根: %.2f 和 %.2f\n", root1, root2);
}

void zero(double a, double b, double c) {
    double root = -b / (2 * a);
    printf("一个实根: %.2f\n", root);
}

void negative(double a, double b, double c) {
    double discriminant = b * b - 4 * a * c;
    double realPart = -b / (2 * a);
    double imaginaryPart = sqrt(-discriminant) / (2 * a);
    printf("两个虚根: %.2f + %.2fi 和 %.2f - %.2fi\n", realPart,
    imaginaryPart, realPart, imaginaryPart);
}

int main() {
    double a, b, c;
    printf("请输入a, b, c的值: ");
    scanf("%lf %lf %lf", &a, &b, &c);

    double discriminant = b * b - 4 * a * c;

    if (discriminant > 0) {
        positive(a, b, c);
    } else if (discriminant == 0) {
        zero(a, b, c);
    } else {
        negative(a, b, c);
    }

    return 0;
}
```

```
(usr) (base) albert@alberts-fedora:~/agrLaguEx/Algorithmic-Language-Experiment/cpt7$ ./2
请输入a, b, c的值: 4 1 1
两个虚根: -0.12 + 0.48i 和 -0.12 - 0.48i
(usr) (base) albert@alberts-fedora:~/agrLaguEx/Algorithmic-Language-Experiment/cpt7$ ./2
请输入a, b, c的值: 1 4 1
两个不同的实根: -0.27 和 -3.73
(usr) (base) albert@alberts-fedora:~/agrLaguEx/Algorithmic-Language-Experiment/cpt7$ ./2
请输入a, b, c的值: 1 2 1
一个实根: -1.00
```

4.

// 写一个函数，使给定的一个3*3的二维整型数组转置，即行列互换。

```
#include <stdio.h>

void transpose(int matrix[3][3]) {
    int temp;
    for (int i = 0; i < 3; i++) {
        for (int j = i + 1; j < 3; j++) {
            temp = matrix[i][j];
            matrix[i][j] = matrix[j][i];
            matrix[j][i] = temp;
        }
    }
}

void printMatrix(int matrix[3][3]) {
    for (int i = 0; i < 3; i++) {
        for (int j = 0; j < 3; j++) {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }
}

int main() {
    int matrix[3][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };

    printf("原始矩阵:\n");
    printMatrix(matrix);

    transpose(matrix);
```

```
    printf("转置后的矩阵:\n");  
    printMatrix(matrix);  
  
    return 0;  
}
```

原始矩阵:

1 2 3

4 5 6

7 8 9

转置后的矩阵:

1 4 7

2 5 8

3 6 9

5.

// 写一个函数，使输入的一个字符串按反序存放，在主函数中输入输出反序前后的字符串。

```
#include <stdio.h>  
#include <string.h>  
  
void reverseString(char str[]) {  
    int n = strlen(str);  
    for (int i = 0; i < n / 2; i++) {  
        char temp = str[i];  
        str[i] = str[n - i - 1];  
        str[n - i - 1] = temp;  
    }  
}  
  
int main() {  
    char str[100];  
  
    printf("请输入一个字符串: ");  
    fgets(str, sizeof(str), stdin);  
  
    str[strcspn(str, "\n")] = '\0';  
  
    printf("反序前的字符串: %s\n", str);  
  
    reverseString(str);  
  
    printf("反序后的字符串: %s\n", str);  
  
    return 0;  
}
```

请输入一个字符串: 1 5 3 7
反序前的字符串: 1 5 3 7
反序后的字符串: 7 3 5 1

9.

// 编写一个函数，由实参传来一个字符串，统计此字符串中字母、数字、空格和其他字符的个数，并在主函数中输入字符串以及输出上述的结果。

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

void countCharacters(const char str[], int *letters, int *digits, int
*spaces, int *others) {
    *letters = *digits = *spaces = *others = 0;
    for (int i = 0; str[i] != '\0'; i++) {
        if (isalpha(str[i])) {
            (*letters)++;
        } else if (isdigit(str[i])) {
            (*digits)++;
        } else if (isspace(str[i])) {
            (*spaces)++;
        } else {
            (*others)++;
        }
    }
}

int main() {
    char str[100];
    int letters, digits, spaces, others;

    printf("请输入一个字符串: ");
    fgets(str, sizeof(str), stdin);

    str[strcspn(str, "\n")] = '\0';

    countCharacters(str, &letters, &digits, &spaces, &others);

    printf("字母个数: %d\n", letters);
    printf("数字个数: %d\n", digits);
    printf("空格个数: %d\n", spaces);
    printf("其他字符个数: %d\n", others);

    return 0;
}
```

请输入一个字符串: 1a2b!#ss好
字母个数: 4
数字个数: 2
空格个数: 0
其他字符个数: 5

17.

// 用递归法将一个整数n转换成字符串。例如，输入483，应输出字符串"483"。n的位数不确定，可以是任意位数的整数。

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

void intToStr(int n, char str[], int *index) {
    if (n == 0) {
        return;
    }

    int digit = n % 10;
    intToStr(n / 10, str, index);
    str[*index++] = digit + '0';
    str[*index] = '\0';
}

int main() {
    int n;
    char str[100];
    int index = 0;

    printf("请输入一个整数: ");
    scanf("%d", &n);

    if (n < 0) {
        str[index++] = '-';
        n = -n;
    }

    if (n == 0) {
        str[index++] = '0';
        str[index] = '\0';
    } else {
        intToStr(n, str, &index);
    }

    printf("转换后的字符串: %s\n", str);

    return 0;
}
```

请输入一个整数： 483
转换后的字符串： 483