icoding答案及解析

常用库介绍

`string.h`常用于字符串的处理

`string.h` 是 C 语言中常用的头文件，提供了处理字符串的函数和常量定义。下面是该头文件的主要用法和包含的一些常用函数：

1. 字符串操作函数：
    - `strlen(const char* str)`：计算字符串的长度（不包括结束符'\0'）。
    - `strcpy(char* dest, const char* src)`：将源字符串复制到目标字符串。
    - `strncpy(char* dest, const char* src, size_t n)`：将源字符串的前 n 个字符复制到目标字符串。
    - `strcat(char* dest, const char* src)`：将源字符串追加到目标字符串的末尾。
    - `strncat(char* dest, const char* src, size_t n)`：将源字符串的前 n 个字符追加到目标字符串的末尾。
    - `strcmp(const char* str1, const char* str2)`：比较两个字符串。
    - `strncmp(const char* str1, const char* str2, size_t n)`：比较两个字符串的前 n 个字符。
    - `strstr(const char* haystack, const char* needle)`：在字符串中搜索子串的第一次出现。
    - `strchr(const char* str, int c)`：在字符串中搜索指定字符的第一次出现。
    - `strrchr(const char* str, int c)`：在字符串中搜索指定字符的最后一次出现。
2. 字符串操作宏函数：
    - `memset(void* ptr, int value, size_t num)`：将内存块设置为指定的值。
    - `memcpy(void* dest, const void* src, size_t num)`：将源内存块复制到目标内存块。
    - `memcmp(const void* ptr1, const void* ptr2, size_t num)`：比较两个内存块的前 n 个字节。
    - `memmove(void* dest, const void* src, size_t num)`：将源内存块移动到目标内存块。
3. 字符操作函数：
    - `isalnum(int c)`：检查字符是否为字母或数字。
    - `isalpha(int c)`：检查字符是否为字母。
    - `isdigit(int c)`：检查字符是否为数字。
    - `islower(int c)`：检查字符是否为小写字母。
    - `isupper(int c)`：检查字符是否为大写字母。
    - `tolower(int c)`：将字符转换为小写字母。
    - `toupper(int c)`：将字符转换为大写字母。

除了上述函数和宏定义外，`string.h` 还定义了一些常用的字符常量，例如 '\0'（字符串结束符）。

需要注意的是，`string.h` 是 C 语言的头文件，对于 C++ 编程，可以使用 `<cstring>` 头文件代替，并将函数放在 `std` 命名空间中使用，如 `std::strlen`、`std::strcpy` 等。此外，C++ 还提供了更多的字符串处理函数和类，可以使用 `<string>` 头文件中的 `std::string` 类和相关函数来代替 C 语言中的字符串操作函数。

`ctype.h`字符分类函数和字符转换函数

是 C 语言中常用的头文件，主要定义了字符分类函数和字符转换函数，用于处理字符类型和大小写等操作。下面介绍一些常用函数：

1. 字符分类函数：
    - `isalnum(int c)`：判断字符是否为字母或数字。
    - `isalpha(int c)`：判断字符是否为字母。
    - `isdigit(int c)`：判断字符是否为数字。
    - `isxdigit(int c)`：判断字符是否为十六进制数字。
    - `islower(int c)`：判断字符是否为小写字母。
    - `isupper(int c)`：判断字符是否为大写字母。
    - `isgraph(int c)`：判断字符是否为可打印字符（除空格之外的字符）。
    - `isspace(int c)`：判断字符是否为空格字符（包括空格、制表符、换行符等）。
    - `iscntrl(int c)`：判断字符是否为控制字符。
    - `isprint(int c)`：判断字符是否为可打印字符（包括空格）。
    - `ispunct(int c)`：判断字符是否为标点符号。
2. 字符转换函数：
    - `tolower(int c)`：将大写字母转换成小写字母。
    - `toupper(int c)`：将小写字母转换成大写字母。

需要注意的是，这些函数的参数和返回值都是整数类型，因此需要将字符转换成整数再进行判断或转换。例如：

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

int main() {
    char c = 'a';
    if (isalpha(c)) {
        printf("%c is a letter\n", c);
    }
    printf("%c to upper case is %c\n", c, toupper(c));
    return 0;
}
```

这段代码中，我们使用了 `isalpha()` 函数判断字符 `c` 是否为字母，打印出相应的结果，同时将字符 `c` 转换成大写字母并输出。输出结果为：

```
a is a letter
a to upper case is A
```

总体上，`<ctype.h>` 头文件提供了对字符类型的判断和转换等常用功能，可以较方便地对字符进行处理。

`<stdlib.h>` 内存的操作及随机数生成等函数

是 C 语言中常用的头文件，主要包含了一些通用工具函数，涉及一些与内存的操作及随机数生成等函数。下面介绍一些常用的函数：

1. 内存管理函数：
    - `malloc(size_t size)`：动态分配内存，返回指向分配内存的指针。
    - `calloc(size_t num, size_t size)`：在内存中分配指定数量、指定大小的连续空间，并且每个字节被初始化为 0，返回指向第一个字节的指针。
    - `realloc(void* ptr, size_t size)`：重新分配先前分配的内存大小，并返回指向重新分配内存的指针。
    - `free(void* ptr)`：释放动态分配的内存。
2. 数值转换函数：
    - `atoi(const char* str)`：将字符串 str 转换为整数。
    - `atof(const char* str)`：将字符串 str 转换为浮点数。
    - `strtol(const char* str, char** endptr, int base)`：将字符串 str 转换为长整数，并将转换之后未转换的部分储存在 endptr 中。
    - `strtod(const char* str, char** endptr)`：将字符串 str 转换为双精度浮点数，并将转换之后未转换的部分储存在 endptr 中。
3. 随机数函数：
    - `rand(void)`：生成一个伪随机数。
    - `srand(unsigned int seed)`：设置随机数种子。

需要注意的是，这些函数都需要使用类型转换将返回值转换成需要的类型，例如使用 `strtol()` 函数将字符串转换为长整数：

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

int main() {
    char str[] = "12345";
    char* end = NULL;
    long num = strtol(str, &end, 10); // base 10
    if (*end == '\0') {
        printf("Number: %ld\n", num);
    }
    return 0;
}
```

这段代码中，我们将字符串 “12345” 转换成长整数，并使用 `printf()` 函数打印结果。需要注意的是，当转换完毕之后，我们需要检查剩余未转换的部分，以确保转换的正确性。输出结果为：

```
Number: 12345
```

总体上，`<stdlib.h>` 头文件提供了对内存和数值的处理等常用功能，可以较方便地对数据进行管理和转换。

使用Visual studio运行时的声明

```
#define _CRT_SECURE_NO_WARNINGS  1
```

实验1

计算税金

涉及知识

printf占位符问题

代码

```
#include<stdio.h>

int main()
{
    float a, b;
    printf("Enter an amount:");
    scanf("%f", &a);
    b = a * 1.05;
    printf("with tax added: $%.2f", b);
    return 0;
}
```

计算账单

涉及知识点

运算符

代码

```
#include<stdio.h>

int main()
{
    int a, b, c, d,e,f,g,h;
    printf("Enter a dollar amount :");
    scanf("%d", &a);
    b = a / 20;
    c = a % 20;
    d = c / 10;
    e = c % 10;
    f = e / 5;
    g = e % 5;
    printf("$20 bills:%d\n$10 bills : %d\n$5 bills : %d\n$1 bills :
%d",b,d,f,g);
}
```

还贷计算

解题思路

这个题比较奇怪

是先计算月利率（1+rate）乘本金后再减去本金

即先算息，后还钱

代码

```
#include<stdio.h>

int main()
{
    float loan,rate,payment, first,pay2,third,mrate;
    printf("Enter amout of loan:");
    scanf("%f", &loan);
    printf("Enter interest rate:");
    scanf("%f", &rate);
    printf("Enter monthly payment:");
    scanf("%f", &payment);
    mrate = (rate / 100) / 12 + 1;
    first = loan* mrate- payment;
    pay2 = first* mrate- payment;
    third = pay2* mrate- payment;
    printf("Balance remaining after first payment: $%.2f\n", first);
    printf("Balance remaining after second payment: $%.2f\n", pay2);
    printf("Balance remaining after third payment: $%.2f\n", third);
    return 0;
}
```

日期格式转化

解题思路

考虑录入整型变量与打印时，前面的0会丢失，因此要对mun，day讨论是否添0

代码

```
#include<stdio.h>

int main()
{
    int mun,day, year;
    printf("Enter a date (mm/dd/yyyy):");
    scanf("%d/%d/%d", &mun,&day,&year);
    if (day < 10 && mun < 10)
        printf("You entered the date %d0%d0%d", year, mun, day);
    else if (day >= 10 && mun < 10)
        printf("You entered the date %d0%d%d", year, mun, day);
    else if (day < 10 && mun >= 10)
        printf("You entered the date %d%d0%d", year, mun, day);
    else if (day >= 10 && mun >= 10)
        printf("You entered the date %d%d%d", year, mun, day);
    return 0;
}
```

分数加法

解题思路

通分

代码

```
#include <stdio.h>

int main()
{
    int a,b,c,d,b_d,a_d,b_c,pluss;
    printf("Enter two fractions separated by a plus sign: ");
    scanf ("%d/%d+%d/%d",&a,&b,&c,&d);
    b_d=b*d;
    a_d=a*d;
    b_c=b*c;
    pluss=a_d+b_c;
    printf("The sum is %d/%d",pluss,b_d);
    return 0;
}
```

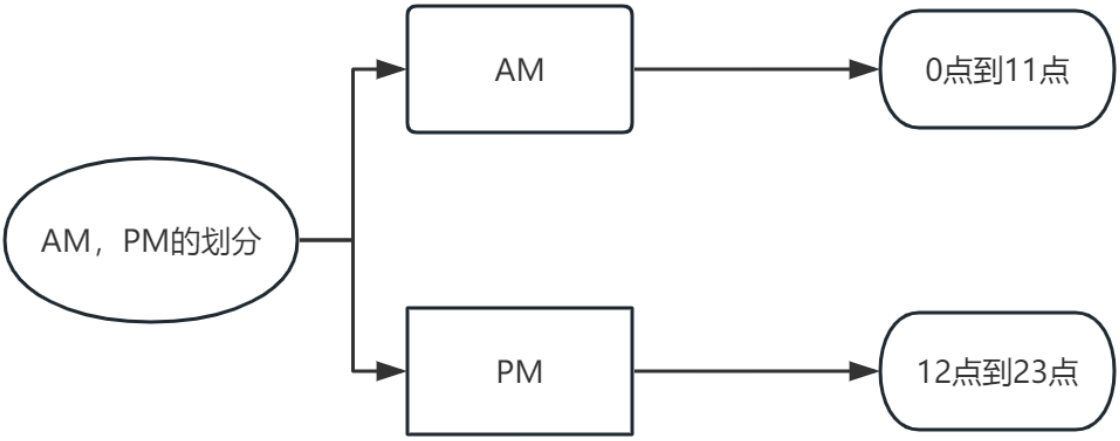
实验2

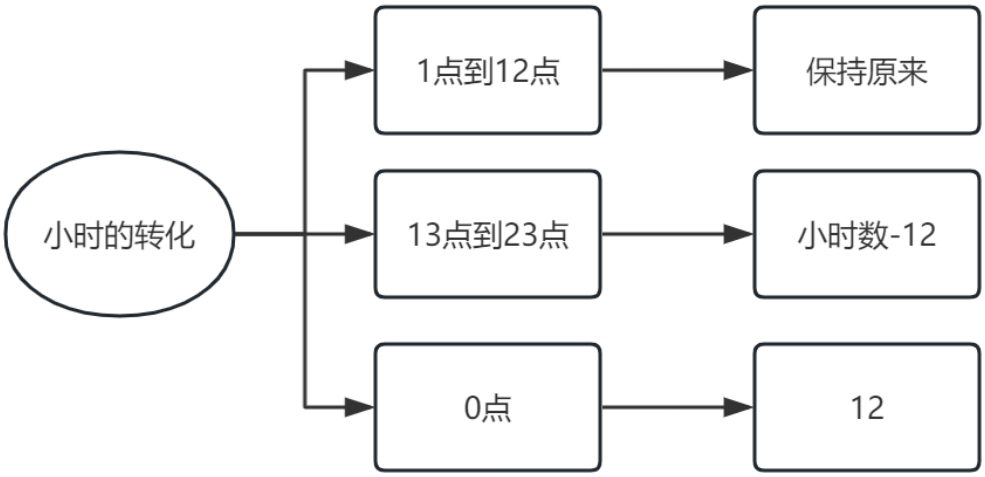
24小时制

看到这道题，我们想到的是

首先分成上下午PM，AM

下图的思路应该比较清晰





其次是对分钟补0的讨论

代码

```
#include <stdio.h>
/*先分AM，PM；*/
int main()
{
    int hour,sec,sec_pm;
    printf("Enter a 24-hour time: ");
    scanf ("%d:%d",&hour,&sec);//读入时间
    if(hour==0&&sec<10)
      printf("Equivalent 12-hour time: 12:0%d AM",sec);
    else if(hour==0&&sec>=10)
      printf("Equivalent 12-hour time: 12:%d AM",sec);
    else if(hour==12&&sec<10)
      printf("Equivalent 12-hour time: 12:0%d PM",sec);
    else if(hour==12&&sec>=10)
      printf("Equivalent 12-hour time: 12:%d PM",sec);
    else if(hour<=11&&sec<10)
      printf("Equivalent 12-hour time: %d:0%d AM",hour,sec);
    else if(hour<=11&&sec>=10)
      printf("Equivalent 12-hour time: %d:%d AM",hour,sec);
    else if(hour>12&&sec<10){
      sec_pm=hour-12;
      printf("Equivalent 12-hour time: %d:0%d PM",sec_pm,sec);
    }
```

```
    else if(hour>12&&sec>=10){
      sec_pm=hour-12;
      printf("Equivalent 12-hour time: %d:%d PM",sec_pm,sec);
    }
    return 0;
}
```

风速等级与描述

代码

```
#include <stdio.h>

int main()
{
    int speed;
    printf("Enter a wind speed: ");
    scanf ("%d",&speed);
    if(speed<1)
      printf("Calm");
    else if(speed>=1&&speed<=3)
      printf("Light air");
    else if(speed>=4&&speed<=27)
      printf("Breeze");
    else if(speed>=28&&speed<=47)
      printf("Gale");
    else if(speed>=48&&speed<=63)
      printf("Storm");
    else if(speed>63)
      printf("Hurricane");
    return 0;
}
```

通用产品代码

通用产品代码规则：

一共有12位, 其中，前6位（013800）是商标识码，后5位（15173）是商品号，最后一位(5)是校验码。
校验码可以通过前11位计算出来，计算规则为：
校验码 = 9 - （3*奇数位和+偶数位和-1）%10

代码

```
#include <stdio.h>

int main()
{
    int i,j,k,res;
    int gro_1,gro_2;
    int
odd,oushu,head,group_1,group_2,last,amount_group_oushu,amount_group_odd,amount_a
ll;
    int lev;
    int shuzu[11];
```

```
    printf("Enter the first (single) digit: ");
    scanf("%d",&head);
    printf("Enter first group of five digits:");
    scanf("%d",&group_1);
    printf("Enter second group of five digits: ");
    scanf("%d",&group_2);
    printf("Enter the last (single) digit: ");
    scanf("%d",&last);
    shuzu[0]=head;
    for(i=5;group_1!=0;i--){
       shuzu[i]=group_1 %10;
       group_1 = group_1 /10;
    }
    for(j=10;group_2!=0;j--){
       shuzu[j]=group_2 %10;
       group_2 = group_2 /10;
    }
    amount_group_odd=shuzu[0]+shuzu[2]+shuzu[4]+shuzu[6]+shuzu[8]+shuzu[10];
    amount_group_oushu=shuzu[1]+shuzu[3]+shuzu[5]+shuzu[7]+shuzu[9];
    amount_all = amount_group_odd*3 + amount_group_oushu;
    lev = 10 - (amount_all%10);
    if (lev==last)
       printf("VALID");
    else
       printf("NOT VALID");
    return 0;
}
```

将百分制转化为等级制

代码

```
#include <stdio.h>

int main()
{
    int grade,leg;
    printf("Enter numerical grade:");
    scanf("%d",&grade);
    leg=grade/10;
    if (grade<0||grade>100){
       printf("Error, grade must be between 0 and 100.");
       return 0;
    }

    switch(leg){
       case 10:
       case 9: printf("Letter grade: A\n");
       break;
       case 8: printf("Letter grade: B\n");
       break;
       case 7: printf("Letter grade: C\n");
       break;
       case 6: printf("Letter grade: D\n");
       break;
```

```
        case 5:
        case 4:
        case 3:
        case 2:
        case 1:
        case 0: printf("Letter grade: F\n");
        break;
    }

    return 0;
}
```

最大公约数

记住辗转相除怎么写就可以了

代码

```
#include <stdio.h>

int gcd(int a, int b);

int gcd(int a, int b) {
    int r;

    while (b != 0) {
        r = a % b;
        a = b;
        b = r;
    }

    return a;
}

int main() {
    int num1, num2;

    printf("Enter two integers: ");
    scanf("%d %d", &num1, &num2);

    int result = gcd(num1, num2);

    printf("Greatest common divisor: %d\n", result);

    return 0;
}
```

股经纪人佣金

这是一个添加题

```
//原始代码
#include <stdio.h>

int main()
{
    float value,commission;
    printf("Enter value of trade:");
    scanf("%f",&value);
    if (value==0)
    printf("\t");
    else if (value < 2500.00f)
      commission = 30.00f + .017f * value;
    else if (value < 6250.00f)
      commission = 56.00f + .0066f * value;
    else if (value < 20000.00f)
      commission = 76.00f + .0034f * value;
    else if (value < 50000.00f)
      commission = 100.00f + .0022f * value;
    else if (value < 500000.00f)
      commission = 155.00f + .0011f * value;
    else if(value > 500000.00f)
      commission = 255.00f + .0009f * value;
    else if (0.00< commission < 39.00f)
      commission = 39.00f;
    printf("Commission: $%.2f",commission);
    return 0;
}
```

现在要在前面添加一个循环，使得除非读入0，不然就一直读入数据

解题思路

注意：相包含的情况不能放在一个if--else之下

```
#include <stdio.h>

int main(void)
{
    float commission, value;
    while (1)
    {
        printf("Enter value of trade: ");
        scanf("%f", &value);
        if (value == 0)
        {
            return 0;
        }

        if (value < 2500.00f)
            commission = 30.00f + .017f * value;
        else if (value < 6250.00f)
            commission = 56.00f + .0066f * value;
        else if (value < 20000.00f)
```

```
            commission = 76.00f + .0034f * value;
        else if (value < 50000.00f)
            commission = 100.00f + .0022f * value;
        else if (value < 500000.00f)
            commission = 155.00f + .0011f * value;
        else//注意改动
            commission = 255.00f + .0009f * value;
        if (commission < 39.00f)
            commission = 39.00f;

        printf("Commission: $%.2f\n\n", commission);
    }
}
```

偶数平方

参考函数

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

int main() {
    double num = 4.0;
    double result = pow(num, 2);

    printf("平方：%lf\n", result);

    return 0;
}
```

代码

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

int main()
{
   int i,j,squir,n,squ_2,t;
   printf("Enter a number:");
   scanf("%d",&n);
   squir=sqrt(n)/2;
   for (i=1;i<=squir;i++)
   {
      t=i*2;
      //squ_2=t*t;
      printf("%d\n",pow(t,2));
   }
   return 0;
}
```

日历

解题思路

对于循环的考察：使用什么循环(循环嵌套？循环套if？)，打印时要打印多少（i初始值，<or<=），

首先要打印空格，要知道输入的是开始日期，而空格是开始日期之前的，要减一；

之后要打印数字和换行符，难在判断何时打印换行符，我们不难得出，当一行满7时就要打印换行了，因此就是

`空格数+数字数=7` 而空格数是starting_day-1,数字数是循环j，就很容易写出了。

代码

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int total_day, starting_day;
    printf("Enter number of days in month :");
    scanf("%d", &total_day);
    printf("Enter starting day of the week(1 = Sun, 7 = Sat) :");
    scanf("%d", &starting_day);
    printf("日\t一\t二\t三\t四\t五\t六\n");
    for (int i = 1; i < starting_day; i++) {
        printf("\t");
    }
    for (int j = 1; j <= total_day; j++) {
        printf("%3d", j);
        if ((j+ starting_day-1)%7==0)
            printf("\n");

    }
    return 0;
}
```

实验3

翻译

涉及知识点

字符类型数组

函数用法介绍

代码

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

void translatePhoneNumber(char *phoneNumber) {
    char translatedNumber[50] = ""; // 存储翻译后的电话号码

    for(int i = 0; phoneNumber[i] != '\0'; i++) {
        if(isalpha(phoneNumber[i])) { // 判断是否是字母
```

```
            switch(toupper(phoneNumber[i])) { // 将字母转换为大写
                case 'A': case 'B': case 'C':
                    translatedNumber[i] = '2';
                    break;
                case 'D': case 'E': case 'F':
                    translatedNumber[i] = '3';
                    break;
                case 'G': case 'H': case 'I':
                    translatedNumber[i] = '4';
                    break;
                case 'J': case 'K': case 'L':
                    translatedNumber[i] = '5';
                    break;
                case 'M': case 'N': case 'O':
                    translatedNumber[i] = '6';
                    break;
                case 'P': case 'Q': case 'R': case 'S':
                    translatedNumber[i] = '7';
                    break;
                case 'T': case 'U': case 'V':
                    translatedNumber[i] = '8';
                    break;
                case 'W': case 'X': case 'Y': case 'Z':
                    translatedNumber[i] = '9';
                    break;
                default:
                    translatedNumber[i] = phoneNumber[i];
            }
        } else {
            translatedNumber[i] = phoneNumber[i];
        }
    }

    printf("Translated phone number: %s\n", translatedNumber);
}

int main() {
    char phoneNumber[50];

    printf("Enter phone number: ");
    fgets(phoneNumber, sizeof(phoneNumber), stdin);

    // 将换行符替换为空字符
    for(int i = 0; phoneNumber[i] != '\0'; i++) {
        if(phoneNumber[i] == '\n') {
            phoneNumber[i] = '\0';
            break;
        }
    }

    translatePhoneNumber(phoneNumber);

    return 0;
}
```

表达式求值

解题思路

读入时：利用了算式一个字符一个数字的特性，可以先读入一个数字，再读入一个字符，直到读到'\n'

计算时：利用了算式从左到右无优先级，计算的结果可以一直存储在第一个变量里面

代码

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

int main()
{
    float num ,num_2;//记录下操作数
    char ope;//记录下操作符

    printf("Enter an expression:");
    scanf("%f",&num);
    while((ope=getchar())!='\n')
    {
        scanf("%f",&num_2);
        switch(ope)
        {
            case '+':
                num+=num_2;
            break;
            case '-':
                num-=num_2;
            break;
            case '*':
                num*=num_2;
            break;
            case '/':
                num/=num_2;
            break;
            defealt:
                printf("输入不合法");
        }
    }
    printf("Value of expression: %.1f",num);
    return 0;
}
```

出现次数

涉及知识点

数组存储数据 数组遍历 除余求每一位数字

代码

```
#include <stdio.h>

int main() {
    int digit_seen[10] = { 0 };
    int digit;
    long n;

    printf("Enter a number: ");
    scanf("%ld", &n);
    if (n==0)
       printf("Digit: 0 1 2 3 4 5 6 7 8 9\n Occurrences: 1 0 0 0 0 0 0 0 0 0");
    else{
        while (n > 0) {
           digit = n % 10;
           digit_seen[digit] += 1;
           n /= 10;
        }
    printf("Digit:    ");
    for ( digit = 0; digit < 10; digit++){
        printf("%3d",digit);
    }

    printf("\nOccurrences:\t");
    for (int digit = 0; digit < 10; digit++) {
        printf("%3d", digit_seen[digit]);
    }
    }
    return 0;
}
```

随机步法

涉及知识点

函数介绍：

`rand()` 和 `srand()` 是 C/C++ 中用于生成伪随机数的函数。下面是它们的配合用法：

1. `srand(unsigned int seed)`：`srand()` 函数用于设置伪随机数的种子。种子决定了随机数序列的起始点。通常情况下，我们可以使用当前时间作为种子，以获得一个不同的伪随机数序列。例如：

```
srand(time(NULL)); // 使用当前时间作为种子
```

注意：`srand()` 函数只需要在程序中调用一次即可。

1. `rand()`：`rand()` 函数用于生成一个伪随机数。调用此函数可返回一个介于 0 和 `RAND_MAX`（通常是 32767）之间的整数。例如：

```
int randomNumber = rand(); // 生成一个伪随机数
```

可以使用 `srand()` 和 `rand()` 结合使用来生成一系列不同的伪随机数。例如，生成一个介于 1 和 100 之间的随机数：

```
srand(time(NULL)); // 设置种子
int randomNumber = rand() % 100 + 1; // 生成一个介于 1 和 100 之间的随机数
```

需要注意的是，使用 `rand()` 函数生成的随机数序列是伪随机的，它们是通过算法计算出来的，并不是真正的随机数序列。同时，在多线程环境下使用 `srand()` 和 `rand()` 可能会导致问题，因为它们使用一个全局的随机数生成器。如果需要更高质量的随机数，可以考虑使用 C++11 引入的 `<random>` 头文件中的随机数函数。

解题思路

1. 随机方向生成：

   ```
   #include <time.h>
   #include <stdlib.h>
   srand(time(NULL));
   int diraction=rand()%4;
   ```

2. 设置两个数组 一个字符型用于储存图表，另一个是数组，用于设置检查机制；

   还有一个字符型变量用于输出A~Z

3. 对于每一个case都判断后再进行操作

代码1(大佬提供)：

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#define N 10
int main()
{

    char A[N][N], c = 'A';
    int check[N + 2][N + 2];
    for (int i = 0; i <= 11; i++) {
        for (int j = 0; j <= 11; j++) {
            check[i][j] = 0;
        }
    }

    for (int i = 0; i < 12; i++) check[0][i] = 1;
    for (int i = 0; i < 12; i++) check[11][i] = 1;
    for (int i = 1; i < 11; i++) check[i][0] = 1;
    for (int i = 1; i < 11; i++) check[i][11] = 1;
    check[1][1] = 1;

    for (int i = 0; i <= 9; i++) {
        for (int j = 0; j <= 9; j++) {
            A[i][j] = '.';
        }
    }

    int i = 0, j = 0, k = 0;
    A[0][0] = 'A';
```

```
    c++;
    srand((unsigned)time(NULL));
    while (c <= 90)
    {
        int n = rand() % 4 + 1;
        switch (n)
        {
        case 1:
            i++;
            if (i<10 && j<10 && i>-1 && j>-1 && !check[i + 1][j + 1])
            {

                A[i][j] = c;
                check[i + 1][j + 1] = 1;
                c++;
            }
            else i--;
            break;
        case 2:
            i--;
            if (i<10 && j<10 && i>-1 && j>-1 && !check[i + 1][j + 1])
            {
                A[i][j] = c;
                check[i + 1][j + 1] = 1;
                c++;
            }
            else i++;
            break;
        case 3:
            j++;
            if (i<10 && j<10 && i>-1 && j>-1 && !check[i + 1][j + 1])
            {

                A[i][j] = c;
                check[i + 1][j + 1] = 1;
                c++;
            }
            else j--;
            break;
        case 4:
            j--;
            if (i<10 && j<10 && i>-1 && j>-1 && !check[i + 1][j + 1])
            {
                A[i][j] = c;
                check[i + 1][j + 1] = 1;
                c++;
            }
            else j++;
            break;
        }
            if (check[i + 1 + 1][j + 1] && check[i - 1 + 1][j + 1] && check[i +
1][j + 1 + 1] && check[i + 1][j - 1 + 1])
            break;
    }
    //打印最终表格
    for (int i = 0; i <= 9; i++)
       {
        for (int j = 0; j <= 9; j++)
```

```
        {
            printf("%c  ", A[i][j]);
        }
        printf("\n");
       }
    return 0;
}
```

代码2(单数组解决):

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#define N 10

int main()
{

    char A[N][N], c = 'A';
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
        {
            A[i][j] = '.';
        }
    }

    int i = 0, j = 0, k = 0;
    A[0][0] = 'A';
    c++;
    srand((unsigned)time(NULL));
    while (c <= 90)
    {
        int n = rand() % 4 + 1;
        switch (n)
        {
        case 1:
            i++;
            if (i < N && j<N && i>-1 && j>-1 && A[i][j]=='.')
            {
                A[i][j] = c;
                c++;
            }
            else i--;
            break;
        case 2:
            i--;
            if (i < N && j<N && i>-1 && j>-1 && A[i][j]=='.')
            {
                A[i][j] = c;
                c++;
            }
            else i++;
            break;
        case 3:
            j++;
            if (i < N && j<N && i>-1 && j>-1 && A[i][j]=='.')
```

```
            {
                A[i][j] = c;
                c++;
            }
            else j--;
            break;
        case 4:
            j--;
            if (i < N && j<N && i>-1 && j>-1 && A[i][j]=='.')
            {
                A[i][j] = c;
                c++;
            }
            else j++;
            break;
        }
        if (A[i + 1][j]!='.' && A[i][j + 1]!='.' && A[i - 1][j]!='.' && A[i][j -
1]!='.')
        {
            break;
        }
    }

    //打印最终表格
    for (int i = 0; i < N; i++)
    {
        for (int j = 0; j < N; j++)
        {
            printf("%c  ", A[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

加密

涉及知识点

函数介绍：整体读入字符串的fgets()

```
#include <stdio.h>

int main() {
    char str[100];

    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);

    printf("You entered: %s", str);

    return 0;
}
```

解题思路

录入：使用fgets函数从键盘输入中读取字符串

翻译：然后在函数kaisa中利用ascll码表，分两类对大小写分别加密

代码

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

void kaisa(char* org_massage, int num)
{
    int i;
    for (i = 0; org_massage[i] != '\0'; i++)
    {

        char c = org_massage[i];

        if (c >= 'A' && c <= 'Z')
        {
            c = 'A' + (c - 'A' + num) % 26;
        }
        else if (c >= 'a' && c <= 'z')
        {
            c = 'a' + (c - 'a' + num) % 26;
        }
        org_massage[i] = c;
    }
}
int main()
{
    char str[50];
    int num;
    printf("Enter message to be encrypted :");
    fgets(str, sizeof(str), stdin);
    printf("Enter shift amount(1 - 25) :");
    scanf("%d", &num);

    kaisa(str, num);

    printf("Encrypted message : %s", str);
    return 0;
}
```

实验4

栈

暂时没有学，期末不考，寒假再研究一下

代码

```
#include <stdbool.h>
#include <stdio.h>
#include <stdlib.h>

#define STACK_SIZE 100

char contents[STACK_SIZE];
int top = 0;
void stack_overflow(void)
{
    printf("Stack overflow\n");
    exit(EXIT_FAILURE);
}

void stack_underflow(void)
{
    printf("Stack underflow\n");
    exit(EXIT_FAILURE);
}


void make_empty(void)
{
    top = 0;
}

bool is_empty(void)
{
    return top == 0;
}

bool is_full(void)
{
    return top == STACK_SIZE;
}

void push(char ch)
{
    if (is_full())
        stack_overflow();
    else
        contents[top++] = ch;
}

char pop(void)
{
    if (is_empty())
        stack_underflow();
    else
        return contents[--top];

    return '\0'; /* prevents compiler warning due to stack_underflow() call */
}

char reverse(char c) {
```

```
    if (c == '}') {
        return '{';
    }
    if (c == ']') {
        return '[';
    }
    return '(';
}

int main(void)
{
    int flag = 0;
    printf("Enter parentheses and/or braces: ");
    char braces[100] = { '/0' };
    scanf("%s", braces);
    for (int i = 0; i < 100; i++) {
        if (braces[i] == 0) {
            break;
        }
        if (braces[i] == '(' || braces[i] == '[' || braces[i] == '{') {
            push(braces[i]);
        }
        else {
            if (!is_empty() && reverse(braces[i]) == contents[top - 1]) {
                pop();
            }
            else {
                flag = 0;
            }
        }

    }
    if (top == 0) {
        flag = 1;
    }
    if (flag == 0) {
        printf("Parentheses/braces are NOT nested properly");
    }
    else
    {
        printf("Parentheses/braces are nested properly");
    }
    return 0;
}
```

逆序

涉及知识点

在需要从标准输入流中读取单个字符时，通常使用 `getchar()` 函数。而在需要从文件中读取一行字符串时，通常使用 `fgets()` 函数

解题思路

读入一个数组之后 在change函数中通过两个指针来两头交换

代码

```
#include <stdio.h>
#include <stdlib.h>
#define N 100

void change(char* left, char* right) {
    char tmp;
    while (left < right) {
        tmp = *left;
        *left = *right;
        *right = tmp;
        left++;
        right--;
    }
}
int main() {
    printf("Enter a message:");
    char chr[N] = { '\0' };
    int i;
    fgets(chr,sizeof(chr),stdin);
    int n = strlen(chr);
    change(chr, chr + n - 1);
    printf("Reversal is: %s", chr);
    return 0;
}
```

最大最小单词简略版

涉及知识点

1. 冒泡排序

```
void change(int *pa,int *pb )
{
    int tmp = *pa;
    *pa = *pb;
    *pb = tmp;
}
void bobo(int *arr,int length)
{
    for (int i = 0; i < length; i++)
    {
        for (int j = 0; j < length-i-1; j++) //遍历一次需要把一个最小的数字搬运到开头
        {
            if (arr[j] > arr[j + 1])
                change(&arr[j], &arr[j + 1]);
        }
    }
}
```

```c
int main()
{
    int arr[10];
    printf("请输入10个整数:");
    for (int i = 0; i < 10; i++) {
        scanf("%d", &arr[i]);
    }
    int sz = sizeof(arr) / sizeof(arr[0]);
    bobo(arr, sz);
    printf("数字从小到大排列: ");
    for (int i = 0; i < N; i++)
    {
        printf("%d\t", arr[i]);
    }
    return 0;
}
```

2. strcpy函数

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

int main() {
    char source[] = "Hello, World!"; // 源字符串
    char destination[20]; // 目标字符数组，要足够大以容纳源字符串

    strcpy(destination, source); // 将源字符串复制到目标字符数组

    printf("复制后的字符串： %s\n", destination);
    return 0;
}
```

3. strcmp函数

```c
int strcmp(const char* str1, const char* str2);
```

参数说明：

- `str1`：要比较的第一个字符串。
- `str2`：要比较的第二个字符串。

返回值：

- 若 `str1` 比 `str2` 先，或者 `str1` 与 `str2` 相等，则返回一个负整数。
- 若 `str2` 比 `str1` 先，则返回一个正整数。
- 若 `str1` 和 `str2` 完全相同，则返回0。

使用 `strcmp` 函数的示例代码如下：

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

int main() {
    char str1[] = "hello";
    char str2[] = "world";
    int result = strcmp(str1, str2); // 比较两个字符串
```

```
    if (result < 0) {
        printf("str1 小于 str2\n");
    }
    else if (result > 0) {
        printf("str1 大于 str2\n");
    }
    else {
        printf("str1 等于 str2\n");
    }
    return 0;
}
```

解题思路

首先，使用for循环把单词一个一个录入二维数组的每一层

之后，利用函数strcpy strcmp实现单词的转换和比较大小

代码

```
#include <stdio.h>
#include<string.h>
#define WIDTH 100
#define LENGTH 20
int main()
{
    char a[WIDTH][LENGTH + 1], str[100] = { "\0" };
    int i, j, k, m;
    for (i = 0; j != 4; ++i)
    {
        printf("Enter word:");
        scanf("%s", &a[i]);
        j = strlen(a[i]);
    }
    for (j = 0; j < i; ++j)   //冒泡排序法
    {
        for (k = 0; k < i - j; ++k)
        {
            if (strcmp(a[k], a[k + 1]) > 0)
            {
                strcpy(str, a[k]);
                for (m = 0; m < 21; ++m)
                    a[k][m] = '\0';
                strcpy(a[k], a[k + 1]);
                for (m = 0; m < 21; ++m)
                    a[k + 1][m] = '\0';
                strcpy(a[k + 1], str);
            }
        }
    }
    printf("Smallest word:%s", a[1]);
    printf("Largest word:%s", a[i]);
    return 0;
}
```

实验5

main函数改造

涉及知识点

解题思路

代码

```
#include <stdio.h>
#include <stdlib.h>
#include "lab51.h" // 请不要删除本行头文件，否则检查不通过
int main()
{
    char code;
    int num_parts = 0;
    struct part inventory[MAX_PARTS];
    for (;;) {
        printf("Enter operation code: ");
        scanf(" %c", &code);
        while (getchar() != '\n')   /* skips to end of line */
            ;
        switch (code) {
        case 'i': insert(num_parts, inventory);
            break;
        case 's': search(num_parts, inventory);
            break;
        case 'u': update(num_parts, inventory);
            break;
        case 'p': print(num_parts, inventory);
            break;
        case 'q': return 0;
        default:  printf("Illegal code\n");
        }
        printf("\n");
    }
}
```

insert函数改造

涉及知识点

解题思路

代码

```
#include <stdio.h>
#include <stdlib.h>
#include "lab51.h" // 请不要删除本行头文件，否则检查不通过
void insert(struct part inv[], int* np)
{
    int part_number;

    if (*np == MAX_PARTS) {
        printf("Database is full; can't add more parts.\n");
```

```
        return;
    }

    printf("Enter part number: ");
    scanf("%d", &part_number);
    if (find_part(part_number, inv, *np) >= 0) {
        printf("Part already exists.\n");
        return;
    }

    inv[*np].number = part_number;
    printf("Enter part name: ");
    read_line(inv[*np].name, NAME_LEN);
    printf("Enter quantity on hand: ");
    scanf("%d", &inv[*np].on_hand);
    (*np)++;
}
```

search函数改造

涉及知识点

解题思路

代码

```
#include <stdio.h>
#include <stdlib.h>
#include "lab51.h" // 请不要删除本行头文件，否则检查不通过

void search(const struct part inv[], int np)
{
    int i,number;

    printf("Enter part number: ");
    scanf("%d",&number);
    i = find_part(number,inv,np);
    if (i >= 0) {
        printf("Part name: %s\n",inv[i].name);
        printf("Quantity on hand: %d\n",inv[i].on_hand);
    } else
      printf("Part not found.\n");
}
```

update函数改造

涉及知识点

解题思路

代码

```
#include <stdio.h>
#include <stdlib.h>
#include "lab51.h" // 请不要删除本行头文件，否则检查不通过

void update(struct part inv[], int np)
{
    int i,number,change;

    printf("Enter part number: ");
    scanf("%d",&number);
    i = find_part(number,inv,np);
    if (i >= 0) {
        printf("Enter change in quantity on hand: ");
        scanf("%d",&change);
        inv[i].on_hand += change;
    } else
      printf("Part not found.\n");
}
```

print函数改造

涉及知识点

解题思路

代码

```
#include <stdio.h>
#include <stdlib.h>
#include "lab51.h" // 请不要删除本行头文件，否则检查不通过

void print(const struct part inv[], int np)
{
    int i;

    printf("Part Number   Part Name          "
           "Quantity on Hand\n");
    for (i = 0; i< np; i++)
    printf("%7d        %-25s%11d\n",inv[i].number,
            inv[i].name,inv[i].on_hand);
}
```

实验6

链表：01-软件界面控制

解题思路

打印一个列表

代码

```
#include "lab52.h" // 请不要删除本行头文件，否则检查不通过
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    GoodsList* goodsList;
    init_list(&goodsList);
    GoodsInfo item;
    char temp_id[MAX_ID_LEN];
    while (1) {
        int choice;
        printf("超市商品管理系统\n");
        printf("********************************************\n");
        printf("1.显示所有商品的信息:\n");
        printf("2.修改某个商品的信息:\n");
        printf("3.插入某个商品的信息:\n");
        printf("4.删除某个商品的信息:\n");
        printf("5.查找某个商品的信息:\n");
        printf("6.商品存盘并退出系统:\n");
        printf("7.对商品价格进行排序:\n");
        printf("8.(慎用)删除所有内容:\n");
        printf("其他.不存盘并退出系统:\n");
        printf("********************************************\n");
        printf("输入您的选择: ");

        scanf("%d", &choice);
        switch (choice) {
        case 1:
            output_all_items(goodsList);
        case 2:
            item = read_goods_info();
            printf("输入要修改记录的 ID: ");
            read_line(temp_id, MAX_ID_LEN);
            change_item(goodsList, temp_id, item);
            break;
        case 3:
            item = read_goods_info();
            int pos;
            printf("输入数字表明你要插入的商品位置: 0.商品列表尾部 1.商品列表头部 i.商品列表中间第i号位置\n");
            scanf("%d", &pos);
            insert_item(goodsList, item, pos);
            break;
        case 4:
            printf("输入要删除记录的 ID: ");
            read_line(temp_id, MAX_ID_LEN);
            delete_item(goodsList, temp_id);
            break;
```

```
        case 5:
            printf("输入要删除记录的 ID：");
            read_line(temp_id, MAX_ID_LEN);
            goodsList = search_item(goodsList, temp_id);
            break;
        case 6:
            save_to_file(goodsList);
            printf("您已经存盘并退出超市商品管理系统!\n");
            return 0;
        case 7:
            bubble_sort(goodsList);
            break;
        case 8:
            destory_list_and_file(&goodsList);
            printf("您已经删除商品文件内容以及链表内容!\n");
            break;
        default:
            destory_list(&goodsList);
            printf("您已经退出超市商品管理系统!\n");
            return 0;
        }
    }
}
```

链表：02-初始化

代码

```
#include "lab52.h" // 请不要删除本行头文件，否则检查不通过
#include <stdio.h>
#include <stdlib.h>

extern int CurrentCnt; // 请不要删除本行的全局变量声明，否则检查不通过

void init_list(GoodsList** L)
{
    FILE* fp;
    GoodsInfo goodsInfo;
    GoodsList *p, *r;

    (*L) = (GoodsList*)malloc(sizeof(GoodsList));
    r = (*L);
    if ((fp = fopen(GOODS_FILE_NAME, "r")) == NULL) {
        if ((fp = fopen(GOODS_FILE_NAME, "w")) == NULL)
            printf("提示：不能创建商品文件\n");
    } else {
        while (!feof(fp)) {
            fscanf(fp, "%s", goodsInfo.goods_id);
            fscanf(fp, "\t%s", goodsInfo.goods_name);
            fscanf(fp, "\t%d", &goodsInfo.goods_price);
            fscanf(fp, "\t%s", goodsInfo.goods_discount);
            fscanf(fp, "\t%d", &goodsInfo.goods_amount);
            fscanf(fp, "\t%d\n", &goodsInfo.goods_remain);
            p = (GoodsList*)malloc(sizeof(GoodsList));
            p->data = goodsInfo;
```

```
            r->next = p;
            r = p;
            CurrentCnt++;
        }
    }
    fclose(fp);
    r->next = NULL;
    printf("商品的链表文件已建立，有%d个商品记录\n", CurrentCnt);
}
```

链表：03-插入

代码

```
#include "lab52.h" // 请不要删除本行头文件，否则检查不通过
#include <stdio.h>
#include <stdlib.h>

extern int CurrentCnt; // 请不要删除本行的全局变量声明，否则检查不通过

bool insert_item(GoodsList* L, GoodsInfo goodsInfo, int choice)
{
    GoodsList* temp;
    GoodsList *pre = L, *p = L->next;
    int i;
    if (CurrentCnt >= 100) {
        printf("信息库已满，要插入请先删除一定量的商品数据!\n");
        return false;
    }
    switch (choice) {
    case 0:
        //尾插法插入新商品
        while (p != NULL) {
            pre = p;
            p = p->next;
        }
        temp = (GoodsList*)malloc(sizeof(GoodsList));
        temp->data = goodsInfo;
        pre->next = temp;
        temp->next = NULL;
        printf("Tips:添加商品%s成功\n", goodsInfo.goods_name);
        CurrentCnt++;
        return true;
    case 1:
        //头插法插入新商品
        temp = (GoodsList*)malloc(sizeof(GoodsList));
        temp->data = goodsInfo;
        temp->next = L->next;
        L->next = temp;
        printf("Tips:添加商品%s成功\n", goodsInfo.goods_name);
        CurrentCnt++;
        return true;
    default:
        //中间i号位置插入新商品，例如：输入3，应该在第二个节点后插入新节点
        // CurrentCnt 改为 CurrentCnt+1，因为当 CurrentCnt 为2时，链表中有两个记录，
```

```
        // 此时输入3，即 choise为 3，表示在第二条记录后插入数据，新记录成为第3条数据
        if (choice <= CurrentCnt + 1 && choice > 0) {
            for (i = 1; i < choice; i++) {
                pre = p;
                p = p->next;
            }
            temp = (GoodsList*)malloc(sizeof(GoodsList));
            temp->data = goodsInfo;
            pre->next = temp;
            temp->next = p;
            printf("Tips:添加商品%s成功\n", goodsInfo.goods_name);
            CurrentCnt++;
            return true;
        } else {
            printf("输入的位置超出当前商品列表范围\n");
            return false;
        }
    }
}
```

链表：04-删除节点

代码

```
#include <stdio.h>
#include "lab52.h" // 请不要删除本行头文件，否则检查不通过
#include <stdio.h>
#include <stdlib.h>

extern int CurrentCnt; // 请不要删除本行的全局变量声明，否则检查不通过

bool delete_item(GoodsList* L, char* goods_id)
{
    GoodsList *pre = L, *p = L->next;
    while (p != NULL && (strcmp(p->data.goods_id, goods_id))) {
        pre = p;
        p = p->next;
    }
    if (p == NULL) {
        return false;
    } else {
        pre->next = p->next;
        free(p);
        CurrentCnt--;
        return true;
    }
}
```

链表：05-查找

解题思路

代码

```
#include "lab52.h" // 请不要删除本行头文件，否则检查不通过
#include <stdio.h>
#include <stdlib.h>

extern int CurrentCnt; // 请不要删除本行的全局变量声明，否则检查不通过

GoodsList* search_item(GoodsList* L, char* goods_id)
{
    GoodsList* p = L->next;
    if (strcmp(goods_id, "-1") == 0)
        return NULL;
    while (p != NULL && (strcmp(p->data.goods_id, goods_id))) {
        p = p->next;
    }
    return p;
}
```

链表：06-修改

解题思路

代码

```
#include "lab52.h" // 请不要删除本行头文件，否则检查不通过
#include <stdio.h>
#include <stdlib.h>

extern int CurrentCnt; // 请不要删除本行的全局变量声明，否则检查不通过

GoodsList* search_item(GoodsList* L, char* goods_id)
{
    GoodsList* p = L->next;
    if (strcmp(goods_id, "-1") == 0)
        return NULL;
    while (p != NULL && (strcmp(p->data.goods_id, goods_id))) {
        p = p->next;
    }
    return p;
}
```

链表：07-显示单个

解题思路

代码

```
#include "lab52.h" // 请不要删除本行头文件，否则检查不通过
#include <stdio.h>
#include <stdlib.h>

extern int CurrentCnt; // 请不要删除本行的全局变量声明，否则检查不通过

void output_one_item(GoodsList* p)
{
    if (p == NULL)
        return;
    else{
        printf("%s\n%s\n%3d\n%s\n%d\n%d\n",
        p->data.goods_id, p->data.goods_name, p->data.goods_price, p-
>data.goods_discount,
        p->data.goods_amount, p->data.goods_remain);
    }
}
```

链表：08-显示所有

解题思路

代码

```
#include "lab52.h" // 请不要删除本行头文件，否则检查不通过
#include <stdio.h>
#include <stdlib.h>

extern int CurrentCnt; // 请不要删除本行的全局变量声明，否则检查不通过

void output_all_items(GoodsList* L)
{
    // GoodsList *pre = L, *p = L->next;
    // int i; //commented by bzj
    GoodsList* p = L->next;
    //printf("当前有%d个商品\n", CurrentCnt);
    while (p != NULL) {
        output_one_item(p);
        p = p->next;
    }
}
```

链表：09-释放链表

解题思路

代码

```
#include <stdio.h>
#include <stdlib.h>
#include "lab52.h" // 请不要删除本行头文件，否则检查不通过

extern int CurrentCnt; // 请不要删除本行的全局变量声明，否则检查不通过

void destory_list_and_file(GoodsList **L){
    destory_list(*L);
    *L=NULL;
    CurrentCnt =0;
    remove("goodinfo.txt");
}
```

链表：10-释放链表并删除文件

解题思路

代码

```
#include <stdio.h>
#include <stdlib.h>
#include "lab52.h" // 请不要删除本行头文件，否则检查不通过

extern int CurrentCnt; // 请不要删除本行的全局变量声明，否则检查不通过

void destory_list_and_file(GoodsList **L){
    destory_list(*L);
    *L=NULL;
    CurrentCnt =0;
    remove("goodinfo.txt");
}
```

链表：11-保存文件

解题思路

代码

```
#include "lab52.h" // 请不要删除本行头文件，否则检查不通过
#include <stdio.h>
#include <stdlib.h>

extern int CurrentCnt; // 请不要删除本行的全局变量声明，否则检查不通过

int save_to_file(GoodsList* L)
```

```
{
    if (L == NULL)
        return 0;
    GoodsList* p = L->next;
    FILE* fp;
    if ((fp = fopen("goodsinfo.txt", "w")) == NULL) {
        printf("提示：不能打开商品文件\n");
        return 0;
    }
    int save_count = 0;
    while (p != NULL) {
        fprintf(fp, "%s\t", p->data.goods_id);
        fprintf(fp, "%s\t", p->data.goods_name);
        fprintf(fp, "%d\t", p->data.goods_price);
        fprintf(fp, "%s\t", p->data.goods_discount);
        fprintf(fp, "%d\t", p->data.goods_amount);
        fprintf(fp, "%d\n", p->data.goods_remain);
        p = p->next;
        save_count++;
    }
    fclose(fp);
    return save_count;
}
```

链表：12-排序

解题思路

代码

```
#include "lab52.h" // 请不要删除本行头文件，否则检查不通过
#include <stdio.h>
#include <stdlib.h>

extern int CurrentCnt; // 请不要删除本行的全局变量声明，否则检查不通过

void bubble_sort(GoodsList* L)
{
    GoodsList* p;
    GoodsInfo temp;
    int n = CurrentCnt;
    int i, j;

    if (L == NULL || L->next == NULL)
        printf("当前链表中没有商品\n");
    for (j = 1; j < n; ++j) {
        p = L->next;
        for (i = 0; i < n - j; ++i) {
            if (p->data.goods_price > p->next->data.goods_price) {
                temp = p->data;
                p->data = p->next->data;
                p->next->data = temp;
            }
            p = p->next;
        }
```

```
    }
}
```

链表：13-读商品信息

解题思路

代码

```
#include "lab52.h" // 请不要删除本行头文件，否则检查不通过
#include <stdio.h>
#include <stdlib.h>

extern int CurrentCnt; // 请不要删除本行的全局变量声明，否则检查不通过

GoodsInfo read_goods_info()
{
    GoodsInfo goodsInfo;
    printf("输入你要插入的商品信息：\n");
    printf("商品ID：");
    read_line(goodsInfo.goods_id, MAX_ID_LEN);
    printf("商品名称：");
    read_line(goodsInfo.goods_name, MAX_NAME_LEN);
    printf("商品价格：");
    scanf("%d", &goodsInfo.goods_price);
    printf("商品折扣：");
    read_line(goodsInfo.goods_discount, MAX_DISCOUNT_LEN);
    printf("商品数量：");
    scanf("%d", &goodsInfo.goods_amount);
    printf("商品剩余：");
    scanf("%d", &goodsInfo.goods_remain);
    return goodsInfo;
}
```