

第一部分 基本程序练习

本节主要练习如何编写基本程序

熟悉程序结构

1. 抄写下列基本程序结构

```
#include <stdio.h>

int main(int argc, char * argv[])
{
    printf("Hello World\n");
    return 0;
}
```

2. 编写一个程序，以月/日/年（即 mm/dd/yy）的格式接受用户录入的日期信息，并以年月日（及 yyyymmdd）的格式将其显示出来：

```
Enter a date(mm/dd/yyyy):10/11/2010
You entered the date 20101011
```

```
#include <stdio.h>

int main(void)
{
    int year, month, day;

    printf("Enter a date(mm/dd/yyyy):");
    scanf("%d/%d/%d", &month, &day, &year);

    printf("You entered the date:%4d/%02d/%02d\n", year, month, day);
    return 0;
}
```

3. printf 函数基本使用，分别使用后续格式控制字 %d %x %f \t \n \b

- 1). 将整形值 7、100、27，以多种格式打印到屏幕 %d %x %f。
- 2). %f 练习，打印 17.2365
- 3). 打印字符串 %d、单引号、双引号
- 4). 使用 sizeof()函数测试打印各种数据类型在 32 平台计算机上所占字节长度

```
#include <stdio.h>
int main(void)
{
    printf("%d, 0x%x, %f\n", 100, 100, (double)100);
    printf("%%, '\n");
    printf("sizeof(char) = %d\n", sizeof(char));
    printf("sizeof(int) = %d\n", sizeof(int));

    return 0;
}
```

4. 计算题

1). 整形数值使用

计算任意自然数的平方值，并打印输出结果

2). 除法题

计算 25 除以 2 的值（要求%d 和 %f 分别显示结果）

3). 取模运算 % ，请说明为什么会有这样的输出结果。

```
if(15 % 4 == 15 / 4)
    printf("equal\n");
else
    printf("not equal\n");
```

4). 使用预定义 #define PI 3.1415926 使用 %f 打印。

计算 $r = 10$ 的圆的周长和面积并打印显示结果

```
#include <stdio.h>
#define PI 3.1415926

int main(void)
{
    unsigned int n;
    int r = 10;

    printf("enter a nature:");
    scanf("%u", &n);
    printf("%d * %d = %d\n", n, n, n * n);

    printf("int:25/2 = %d, double:25/2 = %f\n", 25/2, 25.0/2);
}
```

```
if(15 % 4 == 15 / 4)
    printf("equal!\n");
else
    printf("Not equal!\n");

printf("circle:%f, area:%f\n", 2 * PI * r, PI * r * r);
return 0;
}
```

5. 字符型变量练习

- 1). 声明一个字符型变量并赋初值，显示在屏幕上
- 2). 大写字母转小写字母，并显示在屏幕上
- 3). int i = 49;用 %c 和 %d 分别打印输出

```
#include <stdio.h>

int main(void)
{
    char c = 'A';
    int i = 49;

    printf("c = %c\n", c);

    if(c >= 'A' && c <= 'Z')
        printf("lower:%c\n", c - 32);

    printf("i = %d, %c\n", i, i);
    return 0;
}
```

6. 编写一个自己的大写转小写的函数 char A_to_a(char a); 并在主函数里调用测试验证。

- 1). 只限于字母转换
- 2). 如果输入的不是字母，增加判断处理使程序更加健壮

```
#include <stdio.h>

char A_to_a(char c)
{
    if(c >= 'a' && c <= 'z')
        return c;
```

```
    else if(c >= 'A' && c <= 'Z')
        return c + 32;
    else{
        printf("Can't conversion!\n");
        return c;
    }
}
int main(void)
{
    char c;

    printf("enter a character:");
    scanf("%c", &c);

    printf("A_to_a(%c):%c\n", c, A_to_a(c));
    return 0;
}
```

7. 统计一个自然数的二进制表示形式中有多少个 1。

```
#include <stdio.h>

int main(void)
{
    unsigned int n;
    int i, count = 0;

    printf("enter a number( >0):");
    scanf("%u", &n);

    i = 0;
    do{
        if(n % 2 == 1)
            count++;
    }while(n /= 2);

    printf("count = %d\n", count);
    return 0;
}
```

8. 找出下面程序的错误并修改，并把修改后函数的输出写出来。

```
#include <stdio.h>

void print_segment(void)
```

```
int main(void)
{
    printf("hello world\n");
    print_segment();
    return 0;
}

void print_segment(void)
{
    printf("*****\n");
}
```

9. 编写一个函数 `int is_leap_year(int year)`，判断参数 `year` 是不是闰年，如果 `year` 是闰年则返回 1，否则返回 0。如果某年份能被 4 整除，但不能被 100 整除，那么这一年就是闰年，此外，能被 400 整除的年份也是闰年。

```
#include <stdio.h>

int is_leapyear(int year)
{
    if((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))
        return 1;
    return 0;
}

int main(void)
{
    int year;

    printf("enter year:");
    scanf("%d", &year);

    if(is_leapyear(year) == 1)
        printf("%d is leap year!\n", year);
    else
        printf("%d is not leap year!\n", year);
    return 0;
}
```

10. 编写一个函数 `double myround(double x)`，输入一个小数，将它四舍五入。

例如 myround(-3.51)的值是-4.0 , myround(4.49)的值是 4.0。可以调用 math.h 中的库函数 ceil 和 floor 实现这个函数。

编译命令：gcc -Wall 10.c -o 10 -lm (-lm: 'l':为字符 L 的小写)

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

double myround(double f)
{
    if(f - floor(f) < 0.5)
        return floor(f);
    return ceil(f);
}

int main(void)
{
    double f;

    printf("enter double:");
    scanf("%lf", &f);

    printf("myround(%f) = %f\n", f, myround(f));
    return 0;
}
```

第二部分 基本程序控制练习

本节主要练习 if for while switch 等

熟悉程序结构

1. 抄写程序

```
#include <stdio.h>

int main(int argc, char * argv[])
{
    int a = 10;

    if(a) /*请将 a 替换成 1 测试一下结果*/
        printf("Ture\n");
    else
        printf("False\n");
    return 0;
}
```

1). 编写取绝对值函数 int abs(int x);并在主函数调用测试。

```
#include <stdio.h>

int myabs(int n)
{
    if(n >= 0)
        return n;
    else
        return -n;
}

int main(void)
{
    int n;

    printf("enter a numbe:r");
    scanf("%d", &n);

    printf("abs(%d) = %d\n", n, myabs(n));
    return 0;
}
```

2. 将 1-200 间不能被 3 整除的数输出。

```
#include <stdio.h>

int main(void)
{
    int n, count = 0;

    for(n = 1; n <= 200; n++){
        if(n % 3)
            printf("%3d%c", n, (++count)%10?' ':'\n');
    }
    printf("\n");
    return 0;
}
```

3. 数学表达式 $1000!$ 结果末尾有多少个 0 ?

```
#include <stdio.h>

int main(void)
{
    int result = 0, i;

    for(i = 5; 1000 > i; i *= 5)
        result += 1000 / i;

    printf("result = %d\n", result);
    return 0;
}
```

4. 求表达式值： $\text{sum} = 1 + 3 + 5 + \dots + 99$

```
#include <stdio.h>

int main(void)
{
    int sum = 0, i;

    for(i = 1; i < 100; i += 2)
        sum += i;
    printf("sum = %d\n", sum);

    return 0;
}
```

5. 编程判断 3 人中谁的年龄最大，并打印最大者的年龄。

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



```
int max_age(int a, int b)
{
    return (a > b)?a:b;
}

int main(void)
{
    int x, y, z;

    scanf("%d %d %d", &x, &y, &z);

    printf("max:%d\n", max_age(x, max_age(y, z)));
    return 0;
}
```

6. 有 1、2、3、4 个数字，能组成多少个互不相同且无重复数字的三位数？都是多少？

提示：可填在百位、十位、个位的数字都是 1、2、3、4。组成所有的排列后再去掉不满足条件的排列。

```
#include <stdio.h>

int main(void)
{
    int i, j, k;

    for(i = 1; i < 5; i++)
        for(j = 1; j < 5; j++)
            for(k = 1; k < 5; k++)
                if(i != j && j != k && k != i)
                    printf("%d%d%d\n", i, j, k);

    return 0;
}
```

7. 题目：打印出如下图案（菱形）

```
*
***
*****
*****
*****
***
*
```

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

```
int main(void)
{
    int line, i, j;

    do{
        scanf("%d", &line);
    }while(line % 2 == 0);

    for(i = 0; i < line; i++){
        if(i < line / 2){
            for(j = 0; j < 2 * i + 1; j++)
                printf("*");
            putchar('\n');
        }else{
            for(j = (line - i) * 2 - 1; j > 0; j--)
                printf("*");
            putchar('\n');
        }
    }

    return 0;
}
```

8. 统计一下某字符串中某指定字符出现的次数

```
#include <stdio.h>
#define LEN 20

int main(void)
{
    char str[LEN], c;
    int i, count = 0;

    scanf("%s %c", str, &c);
    printf("%s\n", str);

    for(i = 0; str[i]; i++)
        if(str[i] == c)
            count++;
    printf("count = %d\n", count);

    return 0;
}
```

9. 替换字符串中的指定字符为大写，并打印结果

例如 I like this game, 替换 e 字符为 E

```
#include <stdio.h>
#define LEN 50

int main(void)
{
    char str[LEN], c;
    int i;

    printf("enter a string and a character:\n");
    fgets(str, LEN, stdin);
    c = getchar();

    for(i = 0; str[i] != '\0'; i++){
        if(str[i] == c)
            str[i] = str[i] - 32;
    }

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

10. 输入 3 个数，判断这 3 个数是否能构成三角形。

```
#include <stdio.h>

int main(void)
{
    int a, b, c;

    printf("enter three numbers:");
    scanf("%d %d %d", &a, &b, &c);

    if(a <= 0 || b <= 0 || c <= 0){
        printf("Invaild!\n");
        return 0;
    }

    if(a + b > c && a + c > b && b + c > a){
        printf("%d, %d, %d can found triangle!\n", a, b, c);
    }else{
        printf("%d, %d, %d can't found triangle!\n", a, b, c);
    }
}
```

```
}  
  
return 0;  
}
```

11. 寻找一维数组中最大、小值及其坐标位置并打印输出。

```
#include <stdio.h>  
#include <stdlib.h>  
#include <time.h>  
#define LEN 10  
  
int main(void)  
{  
    int num[LEN], i, max, max_i, min, min_i;  
  
    srand(time(NULL));  
    for(i = 0; i < LEN; i++)  
        printf("%d ", num[i] = rand() % 50);  
    printf("\n");  
  
    max = min = num[0];  
    max_i = min_i = 0;  
  
    for(i = 1; i < LEN; i++){  
        if(max < num[i]){  
            max = num[i];  
            max_i = i;  
        }  
        if(min > num[i]){  
            min = num[i];  
            min_i = i;  
        }  
    }  
  
    printf("max:num[%d] = %d, min:num[%d]= %d\n", max_i, max, min_i, min);  
    return 0;  
}
```

12. 指出下面的程序的问题，并说明为什么？（考点：块语句中的局部变量）

```
#include <stdio.h>  
  
int main(void)  
{  
    int a=9, b=1;
```

```
if(a > b)
{
    int temp = a;
    a = b;
    b = temp;
}
printf("a=%d,b=%d,temp=%d\n", a, b, temp);
return 0;
}
```

13. 实现一个函数，它根据参数进行加法、减法、乘法、除法、取模运算（建议采用 switch 语句）。

```
#include <stdio.h>

int main(void)
{
    int a, b;
    char c;

    printf("please input(num1 + num2):");
    scanf("%d %c %d", &a, &c, &b);

    switch(c){
        case '+': printf("%d %c %d = %d\n", a, c, b, a + b);
                    break;
        case '-': printf("%d %c %d = %d\n", a, c, b, a - b);
                    break;
        case '*': printf("%d %c %d = %d\n", a, c, b, a * b);
                    break;
        case '/': printf("%d %c %d = %d\n", a, c, b, a / b);
                    break;
        case '%': printf("%d %c %d = %d\n", a, c, b, a % b);
                    break;
        default: printf("Invalid!\n");
                    break;
    }
    return 0;
}
```

14. 把代码段

```
if (x > 0 && x < 10)
printf("x is valid.\n");
```

```
else  
printf("x is out of range.\n");
```

改写成下面这种形式：

```
if (____ || ____)  
printf("x is out of range.\n");  
else  
printf("x is valid.\n");
```

附加题：

15. 求 1-100 的素数

```
#include <stdio.h>  
  
int main(void)  
{  
    int n, i;  
  
    for(n = 1; n <= 100; n++){  
        for(i = 2; i < n; i++){  
            if(n % i == 0)  
                break;  
        }  
        if(i == n)  
            printf("%d is prime!\n", n);  
    }  
  
    return 0;  
}
```

16. 自己编写程序实现 void itoa(int x, char *p)(将数字 x，转换为字符串并保存到 p 中)

```
#include <stdio.h>  
  
void reverse(char str[], int len)  
{  
    int i, j;  
    char tmp;  
  
    for(i = 0, j = len - 1; i <= j; i++, j--){  
        tmp = str[i];  
        str[i] = str[j];  
        str[j] = tmp;  
    }  
}
```

```
char *itoa(int n, char str[])
{
    int i = 0, flag = 0;
    long long num = n;

    if(n < 0){
        flag = 1;
        num = (long long)n * -1;
    }
    do{
        str[i++] = num % 10 + '0';
    }while(num /= 10);

    if(flag)
        str[i++] = '-';
    str[i] = '\0';

    reverse(str, i);
    return str;
}

int main(void)
{
    int n;
    char str[50];

    while(1){
        scanf("%d", &n);
        printf("itoa:%s\n", itoa(n, str));
    }
}
```

第三部分 数组练习

本节主要练习各种数组的存储与控制等

熟练使用数组

1. 抄写程序结构

```
#include <stdio.h>
#define SIZE 10

int main(int argc, char * argv[])
{
    int a [SIZE] = {0}, i = 0;
    for(i = 0; i < SIZE; i++)
    {
        a[i] = i + 10;
        printf("a[%d] = %d\n", i , a[i]);
    }
    return 0;
}
```

2. 找出下列程序的错误

程序 1 :

```
int main(void)
{
    int a;
    float a[10];
    .....
}
```

程序 2 :

```
int a=3;
float f_array[a+2];

int main(void)
{
    int a=6;
    float f2_array[a+7];
    .....
}
```


3. 将第 1 题中的数组反向输出。

```
#include <stdio.h>
#define SIZE 10

int main(int argc, char *argv[])
{
    int num[SIZE] = {0}, i = 0;

    for(i = 0; i < SIZE; i++){
        num[i] = i + 10;
        printf("num[%d] = %d\n", i, num[i]);
    }

    printf("reverse...\n");
    for(i = SIZE - 1; i >= 0; i--){
        printf("num[%d] = %d\n", i, num[i]);
    }

    return 0;
}
```

4. 对第 1 题中的数组进行求和操作，并打印计算结果。

计算数组连减之差并打印计算结果。

```
#include <stdio.h>
#define SIZE 10

int main(int argc, char *argv[])
{
    int num[SIZE] = {0}, i = 0;
    int sum = 0, sub;

    for(i = 0; i < SIZE; i++){
        num[i] = i + 10;
        printf("num[%d] = %d\n", i, num[i]);
    }

    printf("sum...\n");
    for(i = 0; i < SIZE; i++){
        sum += num[i];
    }
    printf("sum = %d\n", sum);

    printf("sub...\n");
```

```
sub = num[0];
for(i = 1; i < SIZE; i++){
    sub -= num[i];
}
printf("sub = %d\n", sub);
return 0;
}
```

5. 将二维数组转置。(行变列、列变行)

```
A[5][5] = {
{1,6,2,3,7},
{2,7,3,2,2},
{3,2,5,2,6},
{3,2,6,4,2},
{4,3,7,4,8}};
```

```
#include <stdio.h>
void show(int num[][5], int row)
{
    int i, j;

    for(i = 0; i < row; i++){
        for(j = 0; j < 5; j++){
            printf("%2d ", num[i][j]);
        }
        printf("\n");
    }
}
int main(void)
{
    int num[5][5] = {{1, 6, 2, 3, 7},
                     {2, 7, 3, 2, 2},
                     {3, 2, 5, 2, 6},
                     {3, 2, 6, 4, 2},
                     {4, 3, 7, 4, 8}};

    int i, j, tmp;
    show(num, 5);

    for(i = 0; i < 4; i++){
        for(j = i + 1; j < 5; j++){
            tmp = num[i][j];
            num[i][j] = num[j][i];
            num[j][i] = tmp;
        }
    }
}
```

```
    }  
}  
  
printf("after...\n");  
show(num, 5);  
return 0;  
}
```

6. 编写整型值数组排序程序(升序) (分别使用冒泡排序、选择排序方法)

```
#include <stdio.h>  
#include <stdlib.h>  
#include <time.h>  
#define LEN 10  
  
void init(int num[], int len)  
{  
    int i;  
  
    srand(time(NULL));  
    for(i = 0; i < len; i++)  
        num[i] = rand() % 100;  
}  
  
void show(int num[], int len)  
{  
    int i;  
  
    for(i = 0; i < len; i++)  
        printf("%d ", num[i]);  
  
    printf("\n");  
}  
  
void bubble(int num[], int len)  
{  
    int i, j, tmp;  
  
    for(i = 0; i < len - 1; i++){  
        for(j = 0; j < len - i - 1; j++){  
            if(num[j] > num[j + 1]){  
                tmp = num[j];  
                num[j] = num[j + 1];  
                num[j + 1] = tmp;  
            }  
        }  
    }  
}
```

```
    }  
  }  
}  
  
void selection(int num[], int len)  
{  
    int i, j, pos, tmp;  
  
    for(i = 0; i < len - 1; i++){  
        pos = i;  
        for(j = i + 1; j < len; j++)  
            if(num[pos] > num[j])  
                pos = j;  
        if(pos != i){  
            tmp = num[pos];  
            num[pos] = num[i];  
            num[i] = tmp;  
        }  
    }  
}
```

```
int main(void)  
{  
    int num[LEN];  
  
    printf("bubble sort...\n");  
    init(num, LEN);  
    printf("before sort...\n");  
    show(num, LEN);  
    printf("after sort...\n");  
    bubble(num, LEN);  
    show(num, LEN);  
  
    printf("selection sort...\n");  
    init(num, LEN);  
    printf("before sort...\n");  
    show(num, LEN);  
    printf("after sort...\n");  
    selection(num, LEN);  
    show(num, LEN);  
  
    return 0;  
}
```

7. 通过折半查找法在数组中找到某个数所在的位置并打印，若该数不存在则打印相应信息。(折半查找的前提是待查找数列有序)

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

void init(int num[], int len)
{
    int i;

    srand(time(NULL));
    for(i = 0; i < len; i++)
        num[i] = rand() % 100;
}

void show(int num[], int len)
{
    int i;

    for(i = 0; i < len; i++)
        printf("%d ", num[i]);

    printf("\n");
}

void bubble(int num[], int len)
{
    int i, j, tmp;

    for(i = 0; i < len - 1; i++){
        for(j = 0; j < len - i - 1; j++){
            if(num[j] > num[j + 1]){
                tmp = num[j];
                num[j] = num[j + 1];
                num[j + 1] = tmp;
            }
        }
    }
}

int main(void)
```

```
{
    int num[LEN], start, end, mid, n;

    printf("bubble sort...\n");
    init(num, LEN);
    show(num, LEN);
    bubble(num, LEN);
    show(num, LEN);

    printf("input a number which you want to search\n");
    scanf("%d", &n);

    start = 0, end = LEN - 1;
    while(start <= end){
        mid = (start + end)/2;
        if(num[mid] == n){
            printf("find %d at %d!\n", n, mid);
            break;
        }
        if(num[mid] > n){
            end = mid - 1;
        }else{
            start = mid + 1;
        }
    }

    if(start > end)
        printf("Can't find %d!\n", n);

    return 0;
}
```

8. 给定某年某月某日，将其转换成这一年的第几天并输出。

```
#include <stdio.h>

int daysofmonth[2][13] = {{0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31},
                          {0, 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31}};

int main(void)
{
    int year, month, day, i, leapyear;

    printf("please input year month day:");
    scanf("%d %d %d", &year, &month, &day);
```

```
printf("%02d/%02d is ", month, day);

leapyear = (year % 4 == 0 && year % 100) || (year % 400 == 0);

for(i = 1; i < month; i++)
    day += daysofmonth[leapyear][i];
printf("%d of %4d!\n", day, year);
return 0;
}
```

9. 编写程序检查某一个整数中是否有重复的数字，如检查 2822 中存在重复数字 2。

```
#include <stdio.h>

int main(void)
{
    unsigned int n, digit[10] = {0};

    printf("enter a nature:");
    scanf("%u", &n);

    do{
        if(digit[n % 10] == 1)
            break;
        digit[n % 10] = 1;
    }while(n /= 10);

    if(n > 0)
        printf("Repeated!\n");
    else
        printf("Not repeated!\n");

    return 0;
}
```

10. 通过数组实现对栈的操作，如：push，pop，判空

```
#include <stdio.h>

int stack[100];
int top = 0;

void push(int n)
{
    stack[top++] = n;
}
```

```
int pop(void)
{
    return stack[--top];
}

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

int is_full(void)
{
    return top == 100;
}

int main(void)
{
    int n = 3;
    printf("push 3 times...\n");
    while(!is_full() && n > 0){
        printf("push %d\n", n);
        push(n--);
    }

    printf("pop...\n");
    while(!is_empty())
        printf("pop:%d\n", pop());

    return 0;
}
```


第四部分 字符型数据练习(含字符串) (1)

本节主要练习各种字符型数据练习等，熟练使用字符变量、字符串以及简单字符函数。

1. 抄写程序结构

```
#include <stdio.h>
int main(int argc, char * argv[])
{
    char str[] = "I like programming!";

    int i = 0 ;

    while(str[i] != '\0')    /* 思考一下为啥是 \0 呢？及其作用*/
    {
        printf("%c", str[i]) ;

        i++ ;
    }
    printf("\n");
    return 0;
}
```

1). 怎么能将 str 数组里的内容反向输出？

2). 对于 str 数组里的所有小写字符改成大写输出、大写改小写输出

```
#include <stdio.h>

int main(void)
{
    char str[] = "I like programming!";
    int i = 0;

    while(str[i] != '\0'){
        printf("%c", str[i]);
        i++;
    }
    printf("\n");

    printf("reverse...\n");
    i--;
    while(i >= 0){
```

```
        printf("%c", str[i]);
        i--;
    }
    printf("\n");

    printf("lower and upper exchange...\n");
    i++;
    while(str[i] != '\0'){
        if(str[i] >= 'a' && str[i] <= 'z'){
            str[i] -= 32;
        } else if(str[i] >= 'A' && str[i] <= 'Z'){
            str[i] += 32;
        }
        i++;
    }
    printf("str:%s\n", str);

    return 0;
}
```

2. 判断输入的字符串是否为“回文”（顺读和倒读都一样）字符串

```
#include <stdio.h>
#include <string.h>
#define LEN 50

int main(void)
{
    char str[LEN];
    int i, j, len;

    printf("enter a string:\n");
    scanf("%s", str);
    len = strlen(str);

    for(i = 0, j = len - 1; i <= j; i++, j--){
        if(str[i] != str[j])
            break;
    }
    if(i <= j){
        printf("\"%s\" is not palindrome!\n", str);
    } else {
        printf("\"%s\" is palindrome!\n", str);
    }
}
```

```
return 0;  
}
```

3. 用循环结构将一个字符数组里的值复制到另外一个字符数组里(main 函数里实现)。

1). 练习使用 strcpy 函数

2). 将上题设计思想整理成 my_strcpy 函数，并在 main 函数里调用、测试。

3). 部分复制，例如从第 3 个字符开始复制

4). 编写 my_strncpy 函数

```
#include <stdio.h>  
#define LEN 50  
  
char *my_strcpy(char *dest, char *src)  
{  
    char *dcp;  
  
    for(dcp = dest; (*dcp = *src); dcp++, src++)  
        ;  
  
    return dest;  
}  
  
char *my_strncpy(char *dest, char *src, int n)  
{  
    char *dcp;  
  
    for(dcp = dest; n > 0 && (*dcp = *src); n--, dcp++, src++)  
        ;  
  
    while(n > 0){  
        *dcp = '\0';  
        dcp++;  
        n--;  
    }  
    return dest;  
}  
  
int main(void)  
{  
    char dest[LEN], src[LEN];
```

```
int n, choice;

printf("enter the source string:\n");
scanf("%s", src);

printf("1.strcpy\n2.strcpy from 2\n3.strncpy\nYour choice:");
scanf("%d", &choice);

switch(choice){
case 1: printf("strcpy(dest, \"%s\"): %s\n", src, my_strcpy(dest, src));
        break;
case 2: printf("copy from 2: strcpy(dest, \"%s\"): %s\n", src + 2, my_strcpy(dest, src + 2));
        break;
case 3: printf("enter the number(the 3rd parameter of strncpy):");
        scanf("%d", &n);
        printf("strncpy(dest, \"%s\"): %s\n", src, my_strncpy(dest, src, n));
        break;
default: printf("Invalid choice!\n");
        break;
}

return 0;
}
```

4. 编写统计某字符串的字符长度：int my_strlen(char str[]);

```
#include <stdio.h>
#define LEN 50
int my_strlen(char *str)
{
    int i;
    for(i = 0; str[i] != '\0'; i++)
        ;
    return i;
}

int main(void)
{
    char str[LEN];

    printf("enter a string:");
    scanf("%s", str);

    printf("strlen(\"%s\") = %d\n", str, my_strlen(str));
    return 0;
}
```

```
}
```

5. 有两个字符串 (有效字符不超过 9 个) str1, str2 , 将 str1、str2 的对应字符按从小到大的顺序放到一个数组的对应位置上 ,

如 : char str1 = "hello"; char str2 = "akaedu";

完成组合后 , 新生成的字符串为 : "ahekaedllou"

```
#include <stdio.h>

int main(void)
{
    char str1[10] = "hello", str2[10] = "akaedu", str[20];
    int i = 0, j = 0, k = 0;

    while(str1[i] && str2[j]){
        if(str1[i] < str2[j])
            str[k++] = str1[i++];
        else
            str[k++] = str2[j++];
    }

    while(str1[i])
        str[k++] = str1[i++];

    while(str2[j])
        str[k++] = str2[j++];

    str[k] = '\0';
    printf("str:%s\n", str);
    return 0;
}
```

6. 将两个字符串连接起来 , 并打印结果。

1). 学习使用 strcat 函数

2). 整理成自己的 my_strcat 函数, 并在 main 函数里调用、测试

3). strncat 函数设计

```
#include <stdio.h>
#define LEN 50
```

```
char *my_strcat(char *dest, char *src)
{
    char *dcp = dest;

    while(*dcp)
        dcp++;

    for(; (*dcp = *src); dcp++, src++)
        ;

    return dest;
}

char *my_strncat(char *dest, char *src, int n)
{
    char *dcp = dest;

    while(*dcp)
        dcp++;

    for(; n > 0 && (*dcp = *src); n--, dcp++, src++)
        ;

    *dcp = '\0';

    return dest;
}

int main(void)
{
    char dest[LEN], src[LEN];
    int n;

    printf("enter the dest and source string:\n");
    scanf("%s %s", dest, src);

    printf("strcat(dest, \"%s\"): %s\n", src, my_strcat(dest, src));

    printf("enter a number(the 3rd parameter of strncat)");
    scanf("%d", &n);
    printf("strncat(dest, \"%s\", %d): %s\n", src, n, my_strncat(dest, src, n));

    return 0;
}
```

7. 练习使用 strcmp 并自己编程实现 strcmp

```
#include <stdio.h>
#define LEN 30
int my_strcmp(char *str1, char *str2)
{
    for(; *str1 == *str2; str1++, str2++)
        if(*str1 == '\0')
            return 0;
    return (*str1 > *str2)?1:-1;
}

int main(void)
{
    char str1[LEN], str2[LEN];

    printf("enter two strings:\n");
    scanf("%s %s", str1, str2);

    printf("my_strcmp(\"%s\", \"%s\") = %d\n", str1, str2, my_strcmp(str1, str2));
    return 0;
}
```

第四部分 字符串库函数练习 (2)

1、strchr、strrchr 函数的测试及实现

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

char *my_strchr(char *s, char c)
{
    char *p;

    printf("c = %d\n", c);
    for(p = s; *p != c; p++){
        if(*p == '\0')
            return NULL;
    }
    return p;
}

char *my_strrchr(char *s, char c)
{
    char *p;

    for(p = NULL; ; s++){
        if(*s == c)
            p = s;
        if(*s == '\0')
            return p;
    }
}

int main(int argc, char *argv[])
{
    printf("%s, %s\n", strchr(argv[1], *argv[2]), my_strchr(argv[1], *argv[2]));
    printf("%s, %s\n", strrchr(argv[1], *argv[2]), my_strrchr(argv[1], *argv[2]));

    return 0;
}
```

2、strstr 函数测试及实现

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

char *my_strstr(char *s, char *needle)
```



```
{
    char *p;
    int len1 = strlen(s), len2 = strlen(needle);

    for(p = s; p <= s + len1 - len2; p++){
        if(strncmp(p, needle, len2) == 0)
            return p;
    }
    return NULL;
}

char *my_strstr2(char *s, char *needle)
{
    char *ps;

    if(*needle == '\0')
        return s;

    for(ps = s; *ps; ps++){
        if(*ps == *needle){
            printf("ps:%s, needle:%s\n", ps, needle);
            char *pneedle = needle + 1, *psc = ps + 1;
            for(; *pneedle; pneedle++){
                if(*pneedle == *psc){
                    psc++;
                }else
                    break;
            }
            if(*pneedle == '\0')
                return ps;
        }
    }
    return NULL;
}

int main(int argc, char *argv[])
{
    printf("%s, %s\n", strstr(argv[1], argv[2]), my_strstr(argv[1], argv[2]));
    printf("%s, %s\n", strstr(argv[1], argv[2]), my_strstr2(argv[1], argv[2]));

    return 0;
}
```

3、strcasecmp 函数的测试及实现

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
int my_strcasecmp(char *s1, char *s2)
{
    for(; tolower(*s1) == tolower(*s2); s1++, s2++)
        if(*s1 == '\0')
            return 0;
    return tolower(*s1) - tolower(*s2);
}
int main(int argc, char *argv[])
{
    printf("%d, %d\n", strcmp(argv[1], argv[2]), my_strcasecmp(argv[1], argv[2]));

    return 0;
}
```

- 4、strspn 函数的测试及实现：strspn 函数范例 strspn(“abc”, “bcd”), 搜索字符串“abc”中是否有不存在于字符集“bcd”中出现的字符。如‘a’没有出现在”bcd”中，则函数返回字符串“abc”中‘a’的下标：0，如果第一个参数代表的字符串中的所有字符都出现在第二个参数代表的字符串中则函数返回'\0'字符的下标：3。

```
#include <stdio.h>
#include <string.h>
int my_strspn(char *s, char *set)
{
    char *sc, *setc;

    for(sc = s; *sc; sc++){
        for(setc = set; ; setc++){
            if(*setc == *sc)
                break;
            if(*setc == '\0')
                return sc - s;
        }
    }

    return sc - s;
}
int main(int argc, char *argv[])
{
    printf("%d, %d\n", strspn(argv[1], argv[2]), my_strspn(argv[1], argv[2]));
}
```

```
return 0;  
}
```

- 5、strcspn 函数的测试及实现：strcspn 函数范例 strcspn(“abc”, “bcd”),搜索字符串“abc”中出现在”bcd”字符集中的字符返回其下标，则函数应该返回“abc”中‘b’的下标：1；若第一个参数中所有字符都没有在第二个参数中出现则返回‘\0’的下标。

```
int my_strcspn(char *s, char *set)  
{  
    char *sc, *setc;  
  
    for(sc = s; *sc; sc++){  
        for(setc = set; *setc; setc++){  
            if(*setc == *sc)  
                return sc - s;  
        }  
    }  
  
    return sc - s;  
}  
  
int main(int argc, char *argv[])  
{  
    printf("%d, %d\n", strcspn(argv[1], argv[2]), my_strcspn(argv[1], argv[2]));  
    return 0;  
}
```

- 6、输入字符串，同时输入帧头和帧尾（可多个字符），将该字符串中合法的帧识别出来。

示例 帧头 1a2b 帧尾 3c4d

输入字符串：001a2bssddef3c4d4asf1a2bddaace343c4d1a2bttt1a2bddd 3c4d3c4d

```
#include <stdio.h>  
#include <string.h>  
#define LEN 100  
  
int main(void)  
{  
    char str[LEN], *start, *end, *p;  
  
    fgets(str, LEN, stdin);  
    start = end = str;
```

```
do{
    start = strstr(start, "1a2b");
    end = strstr(end, "3c4d");
    if(!start || !end)
        break;
    if(end < start){
        end = start + 1;
        continue;
    }
    for(p = start + 4; p < end; p++)
        printf("%c", *p);
    printf("\n");
    start = end = end + 1;
}while(start < str + strlen(str));

return 0;
}
```

7、按字符串形式输入 IP 地址，如“192.168.0.120”，判断该地址是哪类地址

知识点：依据 IP 地址的第一个‘.’之前的数判定此 IP 地址是哪一类地址：

若 0 -> A 类地址 10 -> B 类地址 110 -> C 类地址；

若是 A 类地址，那么第一个点前面的是网络地址，余下是主机地址；B 类地址，那么第二个点前面的是网络地址，余下是主机地址；若是 c 类地址，那么第三个点前面的是网络地址，余下是主机地址。如网络地址相同则属于同一个网段地址。

根据以上判断“192.168.0.120”是 C 类地址

```
#include <stdio.h>

int main(void)
{
    char ip[16], *p;
    int n = 0;

    while(1){
        scanf("%s", ip);
        n = 0;
        for(p = ip; *p != '.'; p++)
            n = n * 10 + (*p - '0');
        if(n < 128)
```

```
        printf("Type A!\n");  
    else if(n < 192)  
        printf("Type B!\n");  
    else  
        printf("Type C!\n");  
  
    }  
    return 0;  
}
```

第五部分 数组、字符串、指针综合练习 (1)

1. 计算 100-999 的数中有哪些是水仙花数。水仙花数：是指一个三位正整数，其各位数字的立方之和等于该正整数。如： $407 = 4 \times 4 \times 4 + 0 \times 0 \times 0 + 7 \times 7 \times 7$ ，故 407 是水仙花数。

```
#include <stdio.h>

int main(void)
{
    int n, units = 0, tens = 0, hundreds = 0;

    for(n = 100; n < 1000; n++){
        units = (n % 10) * (n % 10) * (n % 10);
        tens = (n / 10 % 10) * (n / 10 % 10) * (n / 10 % 10);
        hundreds = (n / 100) * (n / 100) * (n / 100);
        if(n == units + tens + hundreds)
            printf("%d is narcissitic!\n", n);
    }

    return 0;
}
```

2. 将两个字符串输入一个二维字符数组中：char str[2][20]，寻找这两个字符串中的最大字符。

```
#include <stdio.h>

int main(void)
{
    char str[2][20], max;
    int i, j;

    for(i = 0; i < 2; i++)
        scanf("%s", str[i]);
    max = str[0][0];

    for(i = 0; i < 2; i++)
        for(j = 0; str[i][j]; j++)
            if(max < str[i][j])
                max = str[i][j];

    printf("max = %c\n", max);
    return 0;
}
```

3. 将 I like this game! 字符串依据空格拆成 4 个单词并存入 str[4][6] 数组, 并输出结果(%s)。

```
#include <stdio.h>

void sep(char *s, char str[][6])
{
    int col = 0, row = 0;
    char *ps = s;

    while(*ps){
        if(*ps != ' '){
            str[col][row++] = *ps;
            ps++;
        } else{
            str[col++][row] = '\0';
            row = 0;
        }
    }
    str[col][row] = '\0';
}

int main(int argc, char *argv[])
{
    char str[4][6];
    int i;

    sep(argv[1], str);
    for(i = 0; i < 4; i++)
        printf("%s\n", str[i]);

    return 0;
}
```

4. char *p = "hello world"; 能否把 h 小写改成大写 H? 有办法么? **不能改写, 没有办法**

char str[] = "hello world"; 能否把 h 小写改成大写 H? **可以改写**

5. 使用交换字节 unsigned short swap_byte(unsigned short i);

Input: 0x1234

Output: 0x3412

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

```
void swap_byte(short *np)
{
    char c;

    c = *(char *)np;
    *(char *)np = *((char *)np + 1);
    *((char *)np + 1) = c;;
}

int main(void)
{
    short sn;

    scanf("%hx", &sn);

    printf("%hx:", sn);
    swap_byte(&sn);
    printf("%hx\n", sn);
    return 0;
}
```

6. char *a[5] ; 怎么赋值和使用？自我设计程序练习实现，排序输出数组内容自我设计程序实现
- 字符串的拷贝

```
#include <stdio.h>
#include <string.h>
void selection(char *strp[], int len)
{
    int i, j, pos;
    char *p;

    for(i = 0; i < len - 1; i++){
        pos = i;
        for(j = i + 1; j < len; j++)
            if(strcmp(strp[pos], strp[j]) > 0)
                pos = j;
        if(pos != i){
            p = strp[pos];
            strp[pos] = strp[i];
            strp[i] = p;
        }
    }
}

int main(void)
```



```
{
    char *strp[5] = {"Tianjin", "Wuhan", "Beijing", "Shanghai", "Changchun"};
    int i;

    selection(strp, 5);
    for(i = 0; i < 5; i++)
        printf("%s\n", strp[i]);

    return 0;
}
```

7. 用数组实现约瑟夫环：约瑟夫环 (Josephus) 问题是由古罗马的史学家约瑟夫 (Josephus) 提出的，他参加并记录了公元 66-70 年犹太人反抗罗马的起义。约瑟夫作为一个将军，设法守住了裘达伯特城达 47 天之久，在城市沦陷之后，他和 40 名将士在附近的一个洞穴中避难。在那里，将士们群情激愤并表示：要投降毋宁死。于是，约瑟夫建议每个人轮流杀死他旁边的人，而这个顺序是由抽签决定的。约瑟夫有预谋地抓到了最后一签并且做为洞穴中两个幸存者之一生存下来。
- 约瑟夫环问题的具体描述是：设有编号为 1, 2, …, n 的 $n(n>0)$ 个人围成一个圈，从第 1 个人开始报数，报到 m 时停止报数，报 m 的人出圈，再从他的下一个人起重新报数，报到 m 时停止报数，报 m 的出圈，……，如此下去，直到只剩下一人为止。当任意给定 n 和 m 后，求 n 个人出圈的次序。

```
#include <stdio.h>
#define PERSON 5
#define COUNT 3
void init(int person[], int len)
{
    int i;

    for(i = 0; i < len; i++)
        person[i] = i + 1;
}

int main(void)
{
    int person[PERSON], step = 1, remain = PERSON, i;

    init(person, PERSON);

    i = 0;
    while(remain > 0){
```

```
    if(person[i] != -1 && step != COUNT){  
        step++;  
    }else if(person[i] != -1 && step == COUNT){  
        printf("%d out!\n", person[i]);  
        person[i] = -1;  
        step = 1;  
        remain--;  
    }  
    i++;  
    if(i == PERSON)  
        i = 0;  
}  
return 0;  
}
```

第五部分 数组、字符串、指针综合练习 (2)

1. 随意输入 5 个整数并存入数组，完成对这 5 个整数的升序排列，然后在输入一个数，插入到此有序数列中来，要求插入后数组仍有序。请对以下三种情况完成程序测试

1) 在最前插入数据 2) 在中间插入数据 3) 在最后插入数据

```
#include <stdio.h>
#define LEN 5

void init(int num[], int len)
{
    int i;

    printf("enter %d numbers:\n", len);
    for(i = 0; i < len; i++)
        scanf("%d", &num[i]);
}

void show(int num[], int len)
{
    int i;

    for(i = 0; i < len; i++)
        printf("%d ", num[i]);
    printf("\n");
}

void selection(int num[], int len)
{
    int i, j, pos, tmp;

    for(i = 0; i < len - 1; i++){
        pos = i;
        for(j = i + 1; j < len; j++)
            if(num[pos] > num[j])
                pos = j;

        if(pos != i){
            tmp = num[pos];
            num[pos] = num[i];
            num[i] = tmp;
        }
    }
}
```

```
    }  
}  
  
int main(void)  
{  
    int num[LEN + 1], n, i, j;  
  
    init(num, LEN);  
    selection(num, LEN);  
    show(num, LEN);  
  
    scanf("%d", &n);  
    printf("n = %d\n", n);  
  
    for(i = 0; i < LEN; i++){  
        if(num[i] > n){  
            for(j = LEN - 1; j >= i; j--)  
                num[j + 1] = num[j];  
            break;  
        }  
    }  
    num[i] = n;  
  
    show(num, LEN + 1);  
    return 0;  
}
```

2、编写函数调用随机数生成函数产生 0 到 19 之间的数，在数组中存入 15 个互不重复的整数。

```
#include <stdio.h>  
#include <stdlib.h>  
#include <time.h>  
  
int main(void)  
{  
    int num[15], i, j, n;  
  
    srand(time(NULL));  
    for(i = 0; i < 15; ){  
        num[i] = rand() % 20;  
        for(j = 0; j < i; j++){  
            if(num[i] == num[j])  
                break;  
        }  
    }
```

```
        if(j == i)
            i++;
    }

    for(i = 0; i < 15; i++)
        printf("%d ", num[i]);
    printf("\n");

    return 0;
}
```

3、编写函数把任意十进制正整数转换为二进制，主函数调用该函数完成整数的二进制形式输出，要求不得按逆序输出。

```
#include <stdio.h>

int main(void)
{
    unsigned int n;
    int binary[32], i;

    scanf("%x", &n);

    printf("n = 0x%x\n", n);
    i = 0;
    do{
        binary[i++] = n % 2;
    }while(n /= 2);
    printf("i = %d\n", i);

    for(--i; i >= 0; i--)
        printf("%d", binary[i]);
    printf("\n");

    return 0;
}
```

4、编写程序，该程序读入一行字符，根据命令行中的参数进行不同的输出。若程序执行时输入以下命令 `./a.out -2`，则程序对所输入的一行字符输出最后 2 个字符(正序)，若有以下命令：`./a.out +6`，则程序对所读入的一行字符输出开头的 6 个字符。(若字符串长度不足指定的数目则输出字符串所有内容)，若命令行中没有参数，则隐含规定输出最后 10 个字符。为简单起见，命令行的

参数指明的长度为个位数。

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define LEN 100

int main(int argc, char *argv[])
{
    char str[LEN];
    int len, n;

    if(!argv[1])
        n = 10;
    else
        n = atoi(argv[1]);

    fgets(str, LEN, stdin);
    len = strlen(str);
    if(str[len - 1] == '\n'){
        str[len - 1] = '\0';
        len--;
    }

    if(n >= len || (-n) >= len){
        printf("%s\n", str);
    }else
        if(n < 0 && -n < len)
            printf("%s\n", str - n);
    else if(n < len){
        str[n] = '\0';
        printf("%s\n", str);
    }

    return 0;
}
```

5、给定程序中，函数 fun 的功能是：将形参 s 所指字符串中的所有字母字符顺序前移，其他字符后移，处理后新字符串的首地址作为函数值返回。例如，s 所指字符串为:asd123fgh543df,处理后新字符串为:asdfghdf354312

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

```
#include <ctype.h>
#include <string.h>
#define LEN 100

int main(void)
{
    char str[LEN], tmp;
    int i, j, dig, len;

    scanf("%s", str);
    len = strlen(str);

    for(i = 0; str[i]; i++){
        if(str[i] >= '0' && str[i] <= '9'){
            dig = i;
            for(j = i + 1; j < len; j++){
                if(isalpha(str[j])){
                    tmp = str[j];
                    str[j] = str[i];
                    str[i] = tmp;
                    break;
                }
            }
            if(j == len)
                break;
        }
    }
    printf("%s\n", str);
    return 0;
}
```

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#define LEN 50
int main(void)
{
    char str[LEN], tmp;
    int i, j, len, pos;
    scanf("%s", str);
    len = strlen(str);
    for(i = pos = len - 1; i >= 0; i--){
        if(str[i] >= '0' && str[i] <= '9'){
```

```
        tmp = str[i];
        for(j = i + 1; j <= pos; j++)
            str[j - 1] = str[j];
        str[pos--] = tmp;
    }
}

printf("str:%s\n", str);
return 0;
}
```

6、给定程序中，函数 fun 的功能是：对形参 s 所指字符串中下标为奇数的字符按 ASCII 码大小递增排序，并将排序后下标为奇数的字符取出，存入形参 p 所指字符数组中，形成一个新串。

```
#include <stdio.h>
#include <string.h>
#define LEN 50
void func(char *src, char *dest)
{
    int i, j, pos, len = strlen(src);
    char tmp;
    for(i = 1, j = 0; src + i < src + len; i += 2, j++)
        dest[j] = src[i];
    dest[j] = '\0';

    printf("dest:%s\n", dest);
    len = strlen(dest);
    for(i = 0; i < len - 1; i++){
        pos = i;
        for(j = i + 1; j < len; j++)
            if(dest[pos] > dest[j])
                pos = j;
        if(pos != i){
            tmp = dest[pos];
            dest[pos] = dest[i];
            dest[i] = tmp;
        }
    }
    printf("dest:%s\n", dest);

    for(i = 1, j = 0; i < len; i += 2, j++)
        dest[j] = dest[i];
    dest[j] = '\0';
}
```



```
printf("dest:%s\n", dest);
}
int main(int argc, char *argv[])
{
    char dest[LEN];
    func(argv[1], dest);
    return 0;
}
```

7、给定程序中，函数 fun 的功能是：找出形参 s 所指字符串中出现频率最高的字母（不区分大小写），并统计出其出现的次数。例如，形参 s 所指的字符串为：abcAbsmaxless，程序执行后的输出结果为：letter 'a' : 3 times letter 's' : 3 times

```
#include <stdio.h>
#include <ctype.h>
int main(int argc, char *argv[])
{
    int letter[26] = {0};
    int i, j, pos;
    char *strp = argv[1], c, max_c, max;

    if(strp == NULL){
        printf("argv[1]:string!\n");
        return 0;
    }
    while(*strp){
        c = toupper(*strp);
        letter[c - 65]++;
        strp++;
    }
    for(i = 0; i < 25; i++){
        pos = i;
        max_c = i + 97;
        for(j = i + 1; j < 26; j++){
            if(letter[pos] < letter[j])
                pos = j;
        }
        if(pos != i){
            c = letter[pos];
            letter[pos] = letter[i];
            letter[i] = c;
            max_c = pos + 97;
        }
    }
    if(i == 0){
```

```
        printf("max: '%c': %d times!\n", max_c, letter[i]);  
        max = letter[i];  
    } else if (letter[i] == max)  
        printf("max: '%c': %d times!\n", max_c, max);  
    }  
    return 0;  
}
```

第六部分 共用体、结构体练习

结构体共用体练习：熟悉结构体、共用体结构和用法

- 1、有一个 unsigned int 型整数，现要分别将其前两个字节和后两个字节作为两个 unsigned short 型整数输出，试编写函数 partition 实现上述要求：void partition(unsigned int n)

```
#include <stdio.h>

union un{
    unsigned int n;
    unsigned short s[2];
};

void partition(unsigned int n)
{
    union un u;
    u.n = n;

    printf("lower:0x%hx, higher:0x%hx\n", u.s[0], u.s[1]);
}

int main(void)
{
    unsigned int n;

    printf("enter an unsigned int(hexadecimal):");
    scanf("%x", &n);

    partition(n);
    return 0;
}
```

2. 分别用指针强制类型转换、共用体知识，判断计算机是大端还是小端系统

大小端系统介绍：采用 Little-endian 模式的系统对操作数的存放方式是从低字节到高字节，而 Big-endian 模式对操作数的存放方式是从高字节到低字节。

如，16bit 的数 0x1234 在 Little-endian 模式内存中的存放方式（设从地址 0x4000 开始）为：

内存地址	0x4000	0x4001
存放内容	0x34	0x12

在 Big-endian 模式内存中的存放方式则为：

内存地址	0x4000	0x4001
存放内容	0x12	0x34

```
#include <stdio.h>

union un{
    char c;
    short s;
};

int main(void)
{
    union un u;
    short s = 0x5678;

    u.s = 0x1234;
    if(u.c == 0x34)
        printf("little endian!\n");
    else
        printf("big endian!\n");

    if(*(char *)&s == 0x78)
        printf("little endian!\n");
    else
        printf("big endian!\n");

    return 0;
}
```

3. 创建结构体 struct person 并编写程序，从终端上输入 5 个人的年龄、性别和姓名，然后输出。

```
struct man{
    char name[20];
    int age;
    char gender[7];
};
```

```
#include <stdio.h>
#define LEN 5
```

```
typedef struct person{
    char name[20];
    int age;
    char gender[7];
}person_t;

int main(void)
{
    person_t person[LEN];
    int i;

    printf("enter 5 persons' info...\n");
    for(i = 0; i < LEN; i++)
        scanf("%s %d %s", person[i].name, &person[i].age, person[i].gender);

    printf("display...\n");
    for(i = 0; i < LEN; i++)
        printf("%s %d %s\n", person[i].name, person[i].age, person[i].gender);
    return 0;
}
```

4. 使用结构体知识，实现在屏幕上模拟一个数字时钟

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

struct clock{
    int hour;
    int min;
    int sec;
};

int main(void)
{
    struct clock time = {0, 0, 50};

    while(1){
        printf("%02d:%02d:%02d\r", time.hour, time.min, time.sec);
        fflush(stdout);
        sleep(1);
        time.sec++;
        if(time.sec == 60){
            time.sec = 0;
            time.min++;
        }
    }
}
```

```
        if(time.min == 60){
            time.min = 0;
            time.hour++;
        }
        if(time.hour == 24)
            time.hour = 0;
    }
    return 0;
}
```

5. 用结构体数组实现 10 个学生的成绩统计。功能包括

- 1). 输入学生学号、姓名、成绩
- 2). 按成绩由高到低进行排序并打印结果并且前 70%的学生打印出标志”pass”，后 30%的学员打印出标志”fail”

结构体为：struct student{

```
    unsigned short id;
    char name[10];
    int score;
```

};

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

```
#define LEN 5
```

```
typedef struct student{
    unsigned short id;
    char name[10];
    int score;
}std_t;
```

```
int main(void)
```

```
{
```

```
    std_t std[LEN], tmp;
    int i, j, pos;
```

```
    for(i = 0; i < LEN; i++){
        scanf("%hu %s %d", &std[i].id, std[i].name, &std[i].score);
```

```
    for(i = 0; i < LEN - 1; i++){
        pos = i;
        for(j = i + 1; j < LEN; j++){
            if(std[pos].score < std[j].score)
```

```
        pos = j;
    }
    if(pos != i){
        tmp = std[pos];
        std[pos] = std[i];
        std[i] = tmp;
    }
}

for(i = 0; i < LEN; i++)
    printf("%hu %s %d %s\n", std[i].id, std[i].name, std[i].score, (i < ((LEN * 70)/100))?"pass":"fail");
return 0;
}
```

6、struct stud{

```
    unsigned short id;
    char name[10];
    int s[4];
    double ave;
};
```

编写：

1)、函数 readrec 把 10 个学生的学号、姓名、四项成绩以及平均分放在一个结构体数组中，学生的学号、姓名和四项成绩由键盘输入，然后计算出平均分放在结构体对应域中。

2)、函数 writerec 输出 10 名学生的记录。

3)、main 函数调用 readrec 函数和 writerec 函数，实现全部程序功能（不允许使用全局变量，函数之间的数据全部使用参数传递）

```
#include <stdio.h>
#define LEN 5

typedef struct student{
    unsigned short id;
    char name[10];
    int s[4];
    double ave;
}std_t;

void readrec(std_t std[], int len)
{
```

```
int i, j;

for(i = 0; i < LEN; i++){
    scanf("%hu %s", &std[i].id, std[i].name);
    for(j = 0; j < 4; j++)
        scanf("%d", &std[i].s[j]);
    }
}

void writerec(std_t std[], int len)
{
    int i, j;

    printf("ID\t%10s\tSC1\tSC2\tSC3\tSC4\tAVERAGE\n", "NAME");
    printf("-----\n");
    for(i = 0; i < LEN; i++){
        printf("%2hu\t%10s\t", std[i].id, std[i].name);
        std[i].ave = 0;
        for(j = 0; j < 4; j++){
            std[i].ave += std[i].s[j];
            printf("%2d\t", std[i].s[j]);
        }
        std[i].ave /= 4;
        printf("%6.2f\n", std[i].ave);
    }
}

int main(void)
{
    std_t std[LEN];

    readrec(std, LEN);
    writerec(std, LEN);

    return 0;
}
```


第七部分 链表、文件练习

1. 抄写程序

```
#include <stdio.h>
typedef struct {
    char name[5];
    int age;
    double record;
} stu_t;
```

定义一个指向 stu 结构体的指针，通过指针赋值输出结构体成员结果。

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

typedef struct{
    char name[10];
    int age;
    double record;
}std_t;

int main(void)
{
    std_t *stdp, std = {"name", 20, 78.5};

    stdp = &std;
    printf("name:%s, age:%d, record:%f\n", stdp->name, stdp->age, stdp->record);

    stdp = malloc(sizeof(std_t));
    printf("enter name age record:\n");
    scanf("%s %d %lf", stdp->name, &stdp->age, &stdp->record);
    printf("name:%s, age:%d, record:%f\n", stdp->name, stdp->age, stdp->record);

    return 0;
}
```

2. 创建一个有 5 个结点的单链表

```
struct student{
    unsigned id;
    char name[20];
    struct student *next;
};
```

```
#include <stdio.h>
#include <stdlib.h>
#define N 5

typedef struct node{
    unsigned id;
    char name[20];
    struct node *next;
}node_t;

int main(void)
{
    node_t *head, *cur, *tail, *del;
    int i;

    for(i = 0; i < N; i++){
        cur = malloc(sizeof(node_t));
        if(cur == NULL){
            printf("malloc fail!\n");
            return 0;
        }
        cur->id = i + 1;
        printf("enter %d name:\n", i + 1);
        scanf("%s", cur->name);
        cur->next = NULL;

        if(i == 0)
            head = cur;
        else
            tail->next = cur;
        tail = cur;
    }

    cur = head;
    while(cur != NULL){
        printf("%d %s\n", cur->id, cur->name);
        del = cur;
        cur = cur->next;
        free(del);
    }
    return 0;
}
```

3. 根据上题创建的链表，删除指定 name 的节点。

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define N 5

typedef struct node{
    unsigned id;
    char name[20];
    struct node *next;
}node_t;

int main(void)
{
    node_t *head, *pre, *cur, *tail, *del;
    char name[20];
    int i;

    for(i = 0; i < N; i++){
        cur = malloc(sizeof(node_t));
        if(cur == NULL){
            printf("malloc fail!\n");
            return 0;
        }
        cur->id = i + 1;
        printf("enter %d name:", i + 1);
        scanf("%s", cur->name);
        cur->next = NULL;

        if(i == 0)
            head = cur;
        else
            tail->next = cur;
        tail = cur;
    }

    printf("enter a name:\n");
    scanf("%s", name);

    cur = pre = head;
    while(cur != NULL){
        if(strcmp(cur->name, name) == 0){
            if(cur == head){
```

```
        del = cur;
        head = pre = cur = cur->next;
        free(del);
    }else{
        del = cur;
        pre->next = cur = cur->next;
    }
    }else{
        pre = cur;
        cur = cur->next;
    }
}

cur = head;
while(cur != NULL){
    printf("%d %s\n", cur->id, cur->name);
    del = cur;
    cur = cur->next;
    free(del);
}
return 0;
}
```

4. 简单的文件拷贝 (使用 fgets, fputs)

```
#include <stdio.h>
#define LEN 20

int main(int argc, char *argv[])
{
    FILE *fpr, *fpw;
    char str[LEN];

    if(argv[1] == NULL || argv[2] == NULL){
        printf("argv[1]:src file, argv[2]:dest file!\n");
        return 0;
    }

    fpr = fopen(argv[1], "r");
    fpw = fopen(argv[2], "w");

    if(!fpr || !fpw){
        printf("fopen fail!\n");
        return 0;
    }
}
```

```
while(fgets(str, LEN, fpr) != NULL)
    fputs(str, fpw);

fclose(fpr);
fclose(fpw);
return 0;
}
```

5. 简单的学生成绩管理。在文件中存储学员成绩信息（假定文件中学生信息不超过 15 条记录。

）：ID 姓名 数学成绩 语文成绩。从文件中读取记录存入结构体数组，结构体类型如下：

```
typedef struct student{
    int id;
    char name[20];
    int math;
    int chinese
    double ave;
} std_t;
```

完成如下功能：

1、显示学员成绩信息（包括数学、语文的平均成绩）

2、查找指定姓名的学生信息

假设有 record 文件，其内容如下（每行为一条学生记录，包括 id，姓名，数学成绩，语文成绩）：

```
1 lindaiyu 10 10
2 jiabaoyu 20 20
3 xuebaochai 30 30
4 lindaiyu 40 40
5 wangxifeng 50 50
```

程序：

```
#include <stdio.h>
#include <string.h>
#define LEN 15

typedef struct student{
    int id;
    char name[20];
```

```
int math, chinese;
double ave;
}std_t;

int main(int argc, char *argv[])
{
    FILE *fpr;
    std_t std[LEN];
    int i, len;
    char name[20];

    if(!argv[1]){
        printf("argv[1] record file!\n");
        return 0;
    }

    fpr = fopen(argv[1], "r");
    if(!fpr){
        printf("fopen fail!\n");
        return 0;
    }

    printf("ID\t%20s\tMATH\tCHI\t%6s\n", "NAME", "AVE");
    printf("-----\n");
    for(i = 0; EOF != fscanf(fpr, "%d %s %d %d\n", &std[i].id, std[i].name, &std[i].math, &std[i].chinese);
i++){
        std[i].ave = (std[i].math + std[i].chinese)/2;
        printf("%2d\t%20s\t%4d\t%3d\t%6.2f\n", std[i].id, std[i].name, std[i].math, std[i].chinese, std[i].ave);
    }
    len = i;

    printf("please input the name who you want to search:");
    scanf("%s", name);

    for(i = 0; i < len; i++){
        if(strcmp(std[i].name, name) == 0)
            printf("%2d\t%20s\t%4d\t%3d\t%6.2f\n", std[i].id, std[i].name, std[i].math, std[i].chinese,
std[i].ave);
    }
    fclose(fpr);
    return 0;
}
```

第八部分 数组、指针、结构体综合练习

- 1、编写一个 dispdate 函数，用来从命令行读取数据，要求用户输入的日期形式为：4/10/2011，并假设日期中没有空格，执行函数后显示的数据格式为：April 10， 2011，如果没有按照指定格式录入日期，那么程序显示出错信息。（可以使用 sscanf 函数从命令行截取出年、月、日）

```
#include <stdio.h>

char *mon[] = {NULL, "Jan", "Feb", "March", "April", "May", "June", "July", "Aug", "Sep", "Oct", "Nov", "Dec"};

int main(int argc, char *argv[])
{
    int month, day, year, ret;

    ret = sscanf(argv[1], "%d/%d/%d", &month, &day, &year);
    if(ret < 3){
        printf("error!\n");
        return 0;
    }
    printf("%s %d, %d\n", mon[month], day, year);

    return 0;
}
```

- 2、从键盘输入一个字符串，然后按照下面的要求输出一个新字符串。新字符串是在原串中的每两个字符之间插入一个空格。要求在自定义函数 insert 中完成新串的产生。

```
#include <stdio.h>
#include <string.h>
#define LEN 50

char *insert(char *dest, char *src)
{
    int i, j;

    for(i = 0, j = 0; src[i]; i++, j++){
        if(j == i * 2 - 1)
            dest[j++] = ' ';
        dest[j] = src[i];
    }
}
```

```
    dest[j] = '\0';
    return dest;
}

char *ins(char *str)
{
    int len = strlen(str), i, j;

    for(i = len - 1, j = i * 2; i >= 0; i--, j--){
        if(j == i * 2 + 1)
            str[j--] = ' ';
        str[j] = str[i];
    }
    str[len * 2 - 1] = '\0';

    return str;
}

int main(void)
{
    char src[LEN], dest[2 * LEN];
    char str[2 * LEN];

    scanf("%s", src);
    printf("%s\n", insert(dest, src));

    strcpy(str, src);
    printf("%s\n", ins(str));
    return 0;
}
```

- 3、设有一个数列，包含 10 个整数，要求此数列已经按照升序排好。现编写程序，要求程序实现能把从指定位置开始的 n 个数按逆序重新排列并输出新的完整数列。

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

#define LEN 10

void init(int num[], int len)
{
    int i;
```



```
    srand(time(NULL));
    for(i = 0; i < len; i++)
        num[i] = rand() % 100;
}

void show(int num[], int len)
{
    int i;

    for(i = 0; i < len; i++)
        printf("%d ", num[i]);
    printf("\n");
}

void selection(int num[], int len)
{
    int i, j, pos, tmp;

    for(i = 0; i < len - 1; i++){
        pos = i;
        for(j = i + 1; j < len; j++)
            if(num[pos] > num[j])
                pos = j;
        if(pos != i){
            tmp = num[pos];
            num[pos] = num[i];
            num[i] = tmp;
        }
    }
}

void reverse(int num[], int start, int len)
{
    int tmp, i, j, end = start + len - 1;

    for(i = start, j = end; i <= j; i++, j--){
        tmp = num[i];
        num[i] = num[j];
        num[j] = tmp;
    }
}
```

```
int main(void)
{
    int num[LEN], start, len;

    init(num, LEN);
    show(num, LEN);

    selection(num, LEN);
    show(num, LEN);

    printf("enter start len:");
    scanf("%d %d", &start, &len);

    start = (start < 0)?0:start;
    len = (start + len > LEN)?(LEN - start):len;
    reverse(num, start, len);
    show(num, LEN);

    return 0;
}
```

- 4、调整某一数组中数字的顺序，使得所有奇数位于数组的前半部分，所有偶数位于数组的后半部分。

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define LEN 10
void sort(int num[], int len)
{
    int i, j, tmp;

    for(i = 0, j = len - 1; i <= j; i++){
        if(num[i] % 2 == 0){
            tmp = num[i];
            num[i] = num[j];
            num[j] = tmp;
            i--;
            j--;
        }
    }
}

void show(int num[], int len)
{

```

```
int i;
for(i = 0; i < len; i++)
    printf("%d ", num[i]);
printf("\n");
}

void init(int num[], int len)
{
    int i;
    srand(time(NULL));
    for(i = 0; i < len; i++)
        num[i] = rand() % 40;
}

int main(void)
{
    int num[LEN];

    init(num, LEN);
    show(num, LEN);

    sort(num, LEN);
    show(num, LEN);

    return 0;
}
```

5、统计子串 substr 在源串 src 中出现的次数

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

int main(int argc, char *argv[])
{
    int count = 0, len_src = strlen(argv[1]), len_sub = strlen(argv[2]);
    char *tmp, *src = argv[1], *substr = argv[2], *findp;

    for(tmp = src; tmp < src + len_src; ){
        if(findp = strstr(tmp, substr)){
            count++;
            tmp = findp + len_sub;
        }else
            break;
    }

    printf("count = %d\n", count);
}
```

```
    return 0;
}
```

6、使用一个链表来表示一个任意长度的超长正整数，然后实现两个正整数（等长）的加法。

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

typedef struct node{
    unsigned int n;
    struct node *next;
}node_t;

node_t *creat(node_t *head, char *str)
{
    char *bkup = str;
    node_t *cur;

    while(*bkup){
        cur = malloc(sizeof(node_t));
        if(cur == NULL){
            printf("malloc error!\n");
            exit(1);
        }
        cur->n = *bkup - '0';

        cur->next = head;
        head = cur;
        bkup++;
    }

    return head;
}

void show(node_t *head)
{
    node_t *cur = head, *del;

    while(cur){
        printf("%d ", cur->n);
        cur = cur->next;
    }
    printf("\n");
}
```

```
node_t *add(node_t *head1, node_t *head2, node_t *head)
{
    node_t *cur, *cur1 = head1, *cur2 = head2;
    int oldcarry = 0, sum = 0;

    printf("add...\n");
    printf("add...head1 = %p, head2 = %p\n", head1, head2);
    show(head1);

    while(cur1 || oldcarry){

        if(cur1){
            sum = cur1->n + cur2->n;
            cur1 = cur1->next, cur2 = cur2->next;
        }
        sum += oldcarry;

        printf("sum = %d, oldcarry = %d\n", sum, oldcarry);
        cur = malloc(sizeof(node_t));
        if(sum >= 10){
            cur->n = sum - 10;
            oldcarry = 1;
        }else{
            cur->n = sum;
            oldcarry = 0;
        }
        sum = 0;

        cur->next = head;
        head = cur;
    }

    return head;
}

int main(int argc, char *argv[])
{
    node_t *head1 = NULL, *head2 = NULL, *head = NULL;

    head1 = creat(head1, argv[1]);
    head2 = creat(head2, argv[2]);
    printf("main...head1 = %p, head2 = %p\n", head1, head2);

    show(head1);
```

```
show(head2);

head = add(head1, head2, head);
show(head);
return 0;
}
```

7、链表实现约瑟夫环问题

解法 1：

```
#include <stdio.h>
#include <stdlib.h>
#define PERSON 5
#define COUNT 3

typedef struct node{
    int item;
    struct node *next;
}node_t;

node_t *make_node(int item, node_t *next)
{
    node_t *t = malloc(sizeof(node_t));
    t->item = item;
    t->next = next;
    return t;
}

int main(int argc, char *argv[])
{
    int i;
    node_t *cur, *del;

    cur = make_node(1, NULL);
    cur->next = cur;

    for(i = 2; i <= PERSON; i++)
        cur = cur->next = make_node(i, cur->next);

    while(cur != cur->next){
        for(i = 1; i < COUNT; i++)
            cur = cur->next;
        del = cur->next;
        cur->next = cur->next->next;
    }
```

```
        printf("%d kill!\n", del->item);
        free(del);
    }
    printf("%d survived!\n", cur->item);

    return 0;
}
```

解法 2 :

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

#define PERSON 10
#define COUNT 3

typedef struct node{
    unsigned n;
    struct node *next;
}node_t;

node_t *creat(node_t *head)
{
    node_t *cur, *tail;
    int i;

    for(i = 0; i < PERSON; i++){
        cur = malloc(sizeof(node_t));
        cur->n = i + 1;
        cur->next = NULL;

        if(i == 0)
            head = cur;
        else
            tail->next = cur;

        tail = cur;
    }
    tail->next = head;

    return head;
}

int main(void)
{

```

```
node_t *head = NULL, *cur, *pre, *del;
int step = 1, remain = PERSON;

head = creat(head);
cur = pre = head;

while(remain > 0){
    if(step != COUNT){
        pre = cur;
        cur = cur->next;
        step++;
    }else{
        del = cur;
        printf("%d out!\n", del->n);
        pre->next = cur = cur->next;
        remain--;
        step = 1;
        free(del);
    }
}
return 0;
}
```

8、从命令行获取一个文件名称，统计一下内容

- 1)、文件中字符的个数
- 2)、文件中单词的数量 (“单词”指不含空白字符的任意序列)
- 3)、文件的行数

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    int letter = 0, line = 0, word = 0, c, flag = 0;
    FILE *fp = fopen(argv[1], "r");

    while(EOF != (c = fgetc(fp))){
        letter++;
        if(c == '\n')
            line++;
        if((c == ' ' || c == '\n') && flag == 1){
            word++;
            flag = 0;
        }
    }
}
```



```
        }else if(c != ' ' && c != '\n'){  
            flag = 1;  
        }  
    }  
    fclose(fp);  
    printf("letter = %d, word = %d, line = %d\n", letter, word, line);  
    return 0;  
}
```

第九部分 指针、字符串、结构体、文件综合练习

- 1、编程实现 `int atoi(char *str)`,该函数实现将整数字符串转换为一个整数，如“- 1234”转换为 - 1234。（要求能够处理从命令行获取的数字字符串以及用 `scanf` 读入的数字字符串）

```
#include <stdio.h>

int myatoi(char *str)
{
    int num = 0, flag, i = 0;

    flag = (str[i] == '-')?(++i,-1):1;
    printf("i = %d\n", i);
    for(; str[i] != '\0'; i++){
        if(str[i] >= '0' && str[i] <= '9')
            num = num * 10 + str[i] - '0';
        else
            break;
    }
    return (flag == 1)?num:-num;
}

int main(int argc, char *argv[])
{
    int num;

    num = myatoi(argv[1]);
    printf("num = %d\n", num);
    return 0;
}
```

- 2、编写程序实现以下功能，在字符串 `str` 中找出最大的字符并放在第一个位置上，并将该字符前的原字符往后顺序移动。

如 `str` 字符串内容为“chyab”,程序执行完毕，`str` 中的内容将成为“ychab”

```
#include <stdio.h>
#include <string.h>
#define LEN 50

int main(void)
{
    char str[LEN], max_c;
```

```
int i, max_i, len;

printf("please enter a string:");
fgets(str, LEN, stdin);

len = strlen(str);
if(str[len - 1] == '\n')
    str[len - 1] = '\0';

max_c = str[0];

for(i = 1; str[i] != '\0'; i++){
    if(max_c < str[i]){
        max_c = str[i];
        max_i = i;
    }
}
printf("max_c:%c, max_i:%d\n", max_c, max_i);
for(i = max_i - 1; i >= 0; i--){
    str[i + 1] = str[i];
}
str[0] = max_c;
printf("str:%s\n", str);
return 0;
}
```

3、从键盘接收一个字符串，然后按照字符顺序从小到大进行排序并删除重复字符。

如从键盘输入的内容为“agehiagehp”，执行完程序之后变为：“aeghip”

```
#include <stdio.h>
#include <string.h>
#define LEN 50

void sort(char *str)
{
    int i, j, len = strlen(str), pos;
    char tmp;

    for(i = 0; i < len - 1; i++){
        pos = i;
        for(j = i + 1; j < len; j++)
            if(str[pos] > str[j])
                pos = j;
        if(pos != i){
```

```
        tmp = str[pos];
        str[pos] = str[i];
        str[i] = tmp;
    }
}

void delete(char *str)
{
    int i, j, k;

    for(i = 1, k = 1; str[i] != '\0'; i++){
        for(j = k - 1 ; j >= 0; j--){
            if(str[i] == str[j])
                break;
        }
        if(j < 0)
            str[k++] = str[i];
    }
    str[k] = '\0';
}

int main(void)
{
    char str[LEN];

    scanf("%s", str);

    sort(str);
    delete(str);
    printf("str:%s\n", str);

    return 0;
}
```

- 4、编写程序计算一个英文句子中最长单词的长度 (字母个数) max。假设该英文句子中只含有字母和空格，在空格之间连续的字母串称为单词。

```
#include <stdio.h>
#include <string.h>
#define LEN 50

int main(void)
{
    char str[LEN];
```

```
int i, max = 0, len, start, end;

fgets(str, LEN, stdin);
if(str[strlen(str) - 1] == '\n')
    str[strlen(str) - 1] = '\0';

len = strlen(str);
for(i = 0; i < len; i++){
    for(start = i; str[start] == ' '; start++)
        ;
    for(end = start; str[end] && str[end] != ' '; end++)
        ;
    i = end;
    if(max < end - start)
        max = end - start;
}
printf("max = %d\n", max);
return 0;
}
```

- 5、编写程序实现以下功能：从键盘上读入若干字符串，将其中的小写字母全部转换为大写字母，存储到磁盘文件“upper.txt”中保存。若输入的字符串为“!”则停止转换及写入文件，然后将upper.txt 文件中的内容显示在屏幕上。

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

#define LEN 30
char *toup(char *str)
{
    int i;

    for(i = 0; str[i] != '\0'; i++){
        if(str[i] >= 'a' && str[i] <= 'z')
            str[i] = str[i] - 32;
    }
    return str;
}

int main(void)
{
    char str[LEN];
    FILE *fp = fopen("upper.txt", "w+");
```

```
if(fp == NULL){
    printf("open upper.txt fail!\n");
    return 0;
}

while(1){
    printf("enter a string:");
    scanf("%s", str);

    if(strcmp(str, "!") == 0)
        break;

    toupper(str);
    fputs(str, fp);
}
rewind(fp);

while(fgets(str, LEN, fp))
    printf("%s", str);

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

6、按如下要求构造文件，之后读出文件中的数据存储为链表，文件内容要求

(字符串从 test01 增至 test10)(整形段等差 5)(字符段数据任意)

字符	整形	字符串
A	0	test01
B	5	test02
C	10	test03
D	15	test04

文件内容如下：

```
A 0 test01
B 5 test02
C 10 test03
D 15 test04
E 20 test05
F 25 test06
```

G 30 test07

H 35 test08

I 40 test09

J 45 test10

程序：

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

#define RECORDNUM 10
typedef struct node{
    char c, str[10];
    int n;
    struct node *next;
}node_t;

FILE *open(char *filename)
{
    FILE *fp = fopen(filename, "r");
    if(fp == NULL){
        printf("open fail!\n");
        exit(1);
    }
    return fp;
}

void close(FILE *fp)
{
    if(fp != NULL)
        fclose(fp);
}

node_t *create(node_t *head, FILE *fp)
{
    int i;
    node_t *cur, *tail;

    for(i = 0; i < RECORDNUM; i++){
        cur = malloc(sizeof(node_t));
        if(cur == NULL){
            printf("malloc fail!\n");
            exit(1);
        }
        fscanf(fp, "%c %d %s\n", &cur->c, &cur->n, cur->str);
        if(i == 0){
```

```
        head = cur;
    }else{
        tail->next = cur;
    }
    tail = cur;
}
return head;
}

void show_destroy(node_t *head)
{
    node_t *cur = head, *tmp;

    printf("show...\n");
    while(cur != NULL){
        printf("%c %d %s\n", cur->c, cur->n, cur->str);
        tmp = cur;
        cur = cur->next;
        free(tmp);
    }
}

int main(void)
{
    node_t *head = NULL;
    FILE *fp = NULL;

    fp = open("linklistinfo");
    head = create(head, fp);
    close(fp);
    show_destroy(head);

    head = NULL;

    return 0;
}
```