C 语言知识点总结

一、 常量

- **数字常量**
  i. **普通数字**：1，35，2.7
  ii. **指数形式**：2.45e-2 等价于 $2.45*10^{-2}$ ***注意 e 大小写皆可，e 前面的数字不能省，就算是 1 也不能省，后面的数字一定要是整数***
  iii. **长整型，单精度浮点型**：3235L，32.5F 分别表示 3235 是长整型数据，32.5 是单精度浮点型左，若不写上 L，F 则表示 3235 是整型，32.5 是双精度浮点型，L，F 大小写皆可
- **字符常量**
  i. **普通字符常量**：用***单引号***把一个字符括起来，如'A','@'
  ii. **转义字符常量**：一对单引号括起来并以"\"开头的字符序列，如'\n'(回车)、'\123'(8 进制 123 对应的字符), '\x23'(16 进制 23 对应的字符)
- **字符串常量**
  用一对***双引号***把一个字符序列括起来，如 "ABCef"， 系统存放字符串常量，每个字符分配一个字节，各字符所占字节紧邻，并且字符串末尾会给再开一个字节里面放一个**'\0'做为结束标志**。
- **符号常量**
  定义格式 **#define 符号常量名 符号常量值**，如#define N 20 则定义了符号常量 N，其值为 20，***注意符号常量名和符号常量值之间是用空格隔开，而不是写上=号，#define 和符号常量名之间也有空格的***。

二、 标识符

- **命名规则**
  以数字，字母，下划线这三类字符组成，但***只能以字母或下划线开头***，而不能也数字开头，另外***不能将关键字做为标识符***。32 个关键字表在 P365 附录 B
- **变量名，函数名，符号常量名全都是标识符**

三、 变量

- **变量的定义格式**
  类型名 变量名;
  如 int a;定义了一个整型常量 a。变量名是由人类随便定义的，符合命名规则的前提下，爱写啥就写啥。所以什么 flag，cc，y1 或者函数名 fun,find 等全部是自定的用来做为名字而已，没有更特别的意义。
- **类型名**
  int 整型，**long** 长整型： 用于存放整数，只是数值范围不同
  float 单精度浮点型 **double** 双精度浮点型：用于存放实数，数值范围，精度不同
  char 字符型：用于存放字符
- **变量赋值，初始化**
  int a=3;定义的同时初始化
  a=6*9;定义后在程序中进行赋值
- **变量的值**

只有在赋值操作时才会被改变，即将其放在等号左边时才会改变它的值，*或自增自减操作*：a=5,a++,a--，像 a+3 并未改变 a 的值，只是使用了 a 的值而已.

- **自增自减运算**

  变量++,++变量，变量--，--变量　使变量的值自增 1 或自减 1

  等价于 变量=变量+1　变量=变量-1

  ++，--放于变量前后效果的区别：

  当自增自减运算做为表达式的一部分时，*++，--放在变量前面是先自增自减再使用变量的值，放在变量后面则是先使用变量的值，再自增自减*。

  如 x=3; printf(“%d”,++x);则相当于执行了++x; printf(“%d”,x);这样的操作所以打印出 4

  再如 x=3; printf(“%d”,x++);则相当于执行了 printf(“%d”,x); x++;这样的操作，则打印出 3，当然最后 x 的值还是 4。

四、 表达式

- **运算符和运算对象**

  一个运算符都有若干个运算对象，如 + 必然要跟两个运算对象才能进行加法运算：3+5。C 语言里称需要跟 n 个运算对象的运算符为 n 元运算符。

  一元运算符有：!，（类型名）

  二元运算符有：+,-,*,/,%(求余), =,+=，-=，*=，/=,%=,< , > , <=, >=, = =(等于), !=(不等于)，&&(且) , ||(或)

  多元运算符有： ，

- **运算符的优先级和结合性**

  i. 优先级：同一个运算对象左右两边若同时有两个运算符，则这两个运算符优先级高的先进行运算。

  ii. 结合性：若同一个运算对象左右两边的两个运算符优先级相同，则根据结合性判断先进行哪个运算，自左自右结合性的先算左边的运算符，自右自左的先算右边的运算符。

  iii. 各运算符的优先级和结合性见 P365 附录 C

- **强制类型转换**

  格式： (类型名) 表达式 。将后跟的表达式的值的数据类型转换为与圆括号内的类型名一致的类型。*注意类型名一定要用（） 括起来*。

- **算术表达式**

  i. 算术运算符：+,-,*,/,%(求余)

  ii. 由算术运算符加上运算对象构成算术表达式,如 3+3*6-9/2

  iii. 值：跟我们小学时学的一样，就是表达式的计算结果

  iv. 整数除以整数结果取整数部分，故 1/3 得到的值是 0

  v. 5%3 结果为 2，想想小学除法，求余得到的是余数不是商。

- **赋值表达式**

  i. 赋值运算符：=，+=，-=，*=，/=,%=

  ii. 赋值表达式：变量=表达式，如 x=3+6，x+=6-9，x+=x*=3+4 *注意等号左边只能是变量*

  iii. 复合赋值运算符的运算：以/=为例：x/=表达式 等价于 x=x/(表达式)

  iv. 值：=号左边的变量最终的值

- **关系表达式**
  i. **关系运算符**：< , > , <=, >=, = =(等于), !=(不等于)
  ii. 由关系运算符加上运算对象构成**关系表达式**,如 3>=4, 2==a
  iii. **值**：满足相应运算符所指定的关系的值为 1，否则为 0
- **逻辑表达式**
  i. **逻辑运算符**：&&(且) , ||(或) , !(非)
  ii. 由逻辑运算符加上运算对象构成**逻辑表达式**,如 3&&4, x||!y
  iii. **值**：满足相应运算符所指定的关系的值为 1，否则为 0
  iv. 进行 ||或 运算时，若||左边的表达式值为 1，则不再对右边的表达式进行运算。
  v. 进行 &&且 运算时，若&&左边的表达式值为 0，则不再对右边的表达式进行运算。
- **逗号表达式**
  i. **逗号运算符**： ，
  ii. 用逗号将各种表达式连续起来构成**逗号表达式**，如 3+4,a=9,8*a
  iii. **值**：组成逗号表达式的各个表达式中的最后一个的值，如上例为 8*a
- 题目：P7—11~17　P8—18~33

五、 输入输出函数

- **scanf(“格式控制串”，变量地址表列);**
  如 scanf(“%d%c%d”,&a,&ch,&b);　　scanf(“%4f”,&x);
  注意：
  第二个参数给的是地址，即要么是&+变量名或数组元素名的形式，要么就是一个数组名或指针变量名，如 int *p,a; p=&a; scanf(“%d”,p);
  考试时注意看题目给你写好的 scanf 的格式
  1. 若其格式控制串内各格式符用“，”隔开如 scanf(“%d，%c，%d”,&a,&ch,&b);那输入时也要用逗号隔开，如此例输入时应：3,+,5
  2. 若是这种格式 scanf(“%d %d”,&a,&b);则输入时应：3 5;
  3. 若是这种格式 scanf(“%d%c%d”,&a,&ch,&b);则输入时应 3+5 *（自己上机运行看看结果）*
- **printf(“格式控制串”，输出项表列);**
  如 float x=7.5; printf(“%8.2f”,x);此处的意思是将 x 打印出来，且占 8 列，保留两位小数。自己上机运行看看效果。
- **常用格式符汇总：**
  i. **%d**：输入输出整型数据，**%ld**：输入输出长整型数据
  ii. **%c**：输入输出字符型数据
  iii. **%f**：输出单（双）精度浮点型数据，输入单精度型数据。　　**%lf**：输入双精度型数据
  iv. **%s**：输入输出一个字符串，用 printf 输出字符串时，输出项书写时可为字符串常量，或字符数组名。如 printf(“%s”,”hello”);或 char str[10]=”hello”; printf(“%s”,str);
  v. **%u**：输入输出无符号整型，**%o**：输入输出八进制数，**%x**:输入输出十六进制数

- **getchar();**
  函数调用后返回用户输入的一个字符，故需再定义一个变量来存放这个字符，即使用时应 char c; c=getchar()；意思就是接收用户输入的一个字符，并将其赋值给变量 c。
- **putchar(字符常量或字符变量名);**
  如 char c='A'; putchar(c);或 putchar('A')；都会向屏幕输出字符 A。

六、 C 语言的语句

- 表达式语句：由表达式末尾加上分号构成。
- 函数调用语句：由函数调用表达式加上分号构成。
- 空语句： ；
- 选择结构语句：if 语句 switch 语句
- 循环语句：for 语句 while 语句 do while 语句
- 复合语句：用花括号｛｝将以上任意语句括起来构成一条复合语句。

七、 C 程序的基本结构

void main()
{ 声明部分：用来定义变量和声明自定义函数的原型，需以“；”结尾，如 int x;
执行语句部分：第六点里介绍的各种语句，如 x=3；printf("%d",x);
}
main 函数外可写自定义函数。如

```
int max()
{
    return 0;
}
```

八、 选择结构语句

- **if(表达式) 语句 1 else 语句 2**
  如果 if 语句的圆括号内的表达式值为非 0，则执行语句 1，值为 0 则执行语句 2。
  i. 表达式可为任意表达式，if 语句执行的实质是判断表达式的值是否为 0 来决定执行语句 1 还是语句 2。*另外请在此处表达严重关切，不管是高手还是菜鸟经常会把判断两个数相等的符号“==”写成了一个等号“=”成为了赋值运算，这样的写法不会引发编译错误，但结果会与原意大大不同，所以考试前请再三提醒自己。*
  ii. 语句 1 和语句 2 都只能是一个语句，*若要跟多条语句，切记用一对{}括起来，构成复合语句；也不要随便在圆括号后加“;”，因“;”构成一条空语句,这会使后面跟的语句 1 不再属于 if 语句的组成部分*。
  iii. if 语句的三种结构
    1. 单边： if(表达式) 语句
    2. 双边：if(表达式) 语句 1 else 语句 2
    3. 多层(重点掌握)：

```
if(表达式 1) 语句 1
else if(表达式 2) 语句 2
```

else if(表达式 3) 语句 3
…
else 语句 n

- **条件运算符 表达式 1? 表达式 2 : 表达式 3**

若表达式 1 的值非 0，则取表达式 2 的值做为整个表达式的值，否则取表达式 3 的值为整个表达式的值。如 3>4? 1:2 该表达式的值为 2

- **switch 语句**

switch(表达式)
{
case 表达式 1：语句
case 表达式 2：语句
…
case 表达式 n：语句
default: 语句
}

语句执行过程：先计算表达式的值，然后判断该值与表达式 1 到表达式 n 中的哪个相等，若与表达式 i 的值相等，则执行表达式 i 后的所有语句，**当遇到 break；语句时结束整个 switch 语句的执行**。表达式 1 到表达式 n 的值都不相等的情况下执行 default 后跟的语句。每个 case 后可跟多条语句。

九、 循环结构

- **for 循环语句**

for(表达式 1；表达式 2；表达式 3) 循环体语句

语句执行过程：

1. 计算表达式 1
2. 判断表达式 2 的值是否为 0，若为 0，语句执行结束，若不为 0，进入步骤 3
3. 执行循环体语句（*需注意的是循环体语句只能有一个语句，若要包含多个语句要用一对{}括起来，构成一条复合语句，此处也不要随便加上“；”，因一个“;”可构成一条空语句，这会使得后面真正的循环体语句不属于 for 循环语句的部分*）。进入步骤 4
4. 计算表达式 3，然后重新进入步骤 2

- **while 循环语句 do while 循环语句**

i. while(表达式) 循环体语句

执行过程：

1. 判断表达式的值是否为非 0，若是进入步骤 2，否则结束语句执行。
2. 执行循环体语句，重新回到步骤 1。

ii. do 循环体语句 while(表达式);

执行过程：

1. 执行循环体语句，进入步骤 2
2. 判断表达式的值是否为非 0，若是重新回到步骤 1，否则结束语句执行。

这里要注意的地方跟 for 语句一样，即循环体语句只能有一个语句，若要包含多个语句要用一对{ }括起来，构成一条复合语句，此处也不要随便加

上“；”，因一个“;”可构成一条空语句，这会使得后面真正的循环体语句不属于 while 循环语句的部分，另外 do while 循环的 while(表达式)后是要加“;”的。

- **break 语句：***放在循环体内实现的功能是结束其所在的那层循环的执行。*

十、 数组

- **定义格式：数据类型 数组名[整型常量]；**如 int a[10]; 定义了一个整型数组，数组名为 a，这个数组含有 10 个元素。
- **引用数组元素：** 格式：**数组名[下标]** **切记下标值从 0 开始**。下标可为常量，表达式，变量等，如 int i=3； a[0]=5；a[3*2]=9； a[i]=7；
- **初始化：数据类型 数组名[整型常量]={数据表列}；**将数据表列的各个值依次赋值给数组的各个元素。如 int a[5]={0,1,2,3,4}；则数组 a 各元素 a[0]到 a[4]的值分别为 0，1，2，3，4
- **遍历数组元素**

  数组定义后，我们*不能对数组进行整体的操作*，如 int a[10];不能用 a=3 这样的操作将数组的各元素都赋值为 3；而只能一个一个元素的进行赋值，如 a[0]=3;a[1]=3;a[2]=3…a[9]=3； 当然此时我们就可以借助于一个 for 循环来控制下标的变化从而对数组的各个元素进行赋值

  for(i=0;i<10;i++) a[i]=3;

  当然这只是用 for 循环遍历数组各元素的最简单的例子，一般考试考的是找出数组元素的某种特性的极值，比如最大值，最小值，或对数组各元素进行排序，这时我们就可以使用 for 循环来遍历数组的各元素，然后在当前循环中得到一个元素再对其进行处理。如 i=2 时访问到的元素是 a[2]，你就可以问问它，你是不是最小值啊。
- **整型数组**

  int a[10]; 整型数组里的各个元素存放的是整数。a[3]=3;
- **字符型数组**

  char str[20]；字符型数组里的各个元素存放的是字符。 str[3]='A';

十一、 字符串函数

- **gets(字符数组名或字符指针变量);**

  如 char str[10],* str2; str2=str;则 gets(str)；或 gets(str2)；都是接收用户输入的字符串如“ABC”存入到字符数组 str 中
- **puts(字符数组名或字符指针变量或字符串常量);**

  如 char str[10]="china"; char *str2;str=str2; 则 puts(str); 或 puts(str2); 或 puts("china");都会在屏幕上打印出 china
- **strlen(字符数组名或字符指针变量);字符串测长函数**

  char str[20]="hello world!";

  int len;len=strlen(str);得出的结果是 len 的值为 12
- **strcat(字符串 1 的地址，字符串 2 的地址);**

  将字符串 2 的内容连接到字符串 1 的尾部。

  char str1[20]="ABC",str2[20]="xyz";

  strcat(str1,str2);

  则程序运行的结果是 str1 内存放的字符串变为 ABCxyz，当然 str2 存放

的字符串还是 xyz。

- **strcmp(字符串 1 的地址，字符串 2 的地址)；**

  比较串 1 和串 2 哪个比较大。比较大小的依据是，两个字符串从左往右相应位置上第一个不相等的字符 ASCII 码值之差。

  char str1[20]=”ABCE”,str2[20]=”ABDE”;

  int i;

  i=strcmp (str1,str2);

  第一个不相等的字符为 str1 的‘C’和 str2 的‘D’，而二者相差-1，故-1 做为 strcmp 函数执行的结果返回到被调用的位置，该位置位于赋值表达式内，故将其值赋值给 i，即此时 i 的值就是-1.

- **strcpy(字符串 1 的地址，字符串 2 的地址)；**

  将字符串 2 的内容复制到字符串 1 内。

  char str1[20]=”ABC”,str2[20]=”xyz”;

  strcpy(str1,str2);此时 str1 的内容为”xyz”，当然 str2 的内容没变

  strcpy(str1,”uvw”);此时 str1 的内容又变成了“uvw“。

十二、 函数

- **函数定义**

函数类型 函数名(形式参数列表)

```
{
        内部变量定义和声明部分
        执行语句
}
```

如：

```
int  max (int  x , int  y )
{  int  z ;
    z= x > y ? x : y ;
    return  ( z ) ;
}
```

注意点：

1.**函数类型是指返回值的类型**，即要与 return 语句后跟的表达式的值的类型一致。***若函数类型为 void 则说明该函数无返回值，即函数体里可以不出现 return 语句***。

2.形式参数列表里定义的变量要记得给它们指定类型，而且***如果同时要定义多个，应在每个前面都分别指定类型名***，而不能写成 int x,y;

3.函数体里能写的语句跟 main 函数一样，在开头可定义所需要的变量，后面跟上一堆执行语句。

- **函数调用流程**

以上面的函数为例，在 main 函数进行调用：

```
void main()
{ int a,b,c;
  scanf(“%d%d”,&a,&b);
  printf(“%d”,max(a,b));或 c=max(a,b);printf(“%d”,c)
```

以上两种方法都会在屏幕中打印出 a，b 间的较大值。

调用函数的格式 函数名(实际参数列表)；调用的时候像什么函数类型，形式参数的类型就不要加上去了。max(a,b)中 max 就是函数名，写上变量名 a,b 是实际参数列表，执行这个调用语句时，会先把 a,b 的值给相应位置的形式参数即执行了 x=a,y=b 这样的操作，然后开始执行 max 函数的函数体的语句。当 max 函数体里执行到一个 return 语句时，则 max 函数结束执行，将 return 后的表达式的值返回给 main 函数调用 max 函数的那个位置，即若上面 a=3,b=5 则 max(a,b)return 后的表达式的值应该是 5 也就是说执行完 max 后把 5 返回到调用 max 的位置可看成 printf(“%d”，5)；或另一种解法的 c=5。

}

- **数组做函数参数：**
  - **（1） 数组元素做参数，同普通变量一样**
  - （2）数组名做参数

归纳起来，如果有一个实参数组，想在函数中改变此数组的元素的值，实参与形参的对应关系有以下四种情况：

实参	形参	示例	

数组名	数组名	main()
		{int a[10];
		…
		f(a,10);
		}
			f(int x[],int n)
			{
			…
			}

数组名	指针	main()
		{int a[10];
		…
		f(a,10);
		}
			f(int *x,int n)
			{
			…
			}

指针	指针	main()
		{int a[10],*p;
		p=a;
		…
		f(p,10);
		}
			f(int *x,int n)
			{
			…
			}

指针	数组名	main()
		{int a[10],*p;
		p=a;
		…
		f(p,10);
		}
			f(int x[],int n)
			{
			…
			}

十三、 指针

- **指针变量的声明： 类型名 * 指针变量名；**
- **通过指针变量访问它所指向的普通变量的值**

  先将普通变量的地址赋值给指针变量，再通过指针运算符* 得到普通变量的值。

  int *p,x,y;

  x=3;

  p=&x;

  则 printf(“%d”,*p);会打印出 3 即 x 的值

  y=*p；则 y 的值变为 3

  *p=5；则 x 的值变为 5
- **指针变量加上（减去）一个位移的效果**

若指针变量存入的是数组元素的地址，则其*加一减一得到的是那个数组元素下一个或前一个元素的地址*。

int a[10]; p=&a[3];

*p 得到的是 a[3]的值。

若 p++；此时 p 存放的是 a[4]的地址&a[4]; *p 得到的就是 a[4]的值。

或 p--；此时 p 存放的是 a[2]的地址&a[2]，*p 得到的就是 a[2]的值。

- **数组指针、指针变量与数组元素之间的关系：**

- **字符串的处理：**

（1）用字符数组存放一个字符串。

char c[]="I have a book!";

（2）用字符指针指向一个字符串。

char *p="I have a book!";

或 char *p;

p="I have a book!";

（3）区别：

（2）中字符指针 p 是一个变量，可以改变 p 使它指向不同的字符串，但不能改变 p 所指的字符串常量。

（1）中 c 是一个数组，可以改变数组中保存的内容。

十四、 宏定义

- **无参宏定义：#define 标识符 值** 定义后，出现所定义的标识符的地方都将以定义时指定的值来代替。

```
#define M 2+3
main()
{ int x;
```

x=M*M;　则 x 的值为 2+3*2+3=11 若想得到的结果是(2+3)*(2+3)则定义时也写成这样 #define M (2+3)

}

注意#define、标识符、值之间都要用空格隔开，且宏定义结尾不需加分号。

- **带参宏定义　#define 标识符(参数表)　值**

```
#define   S(x,y)   x*y
main( )
{   int a=3,b=4,c=5,d=6;
     printf("a*b=%d\n", S(a,b));   //此时会打印出 a*b=12
     printf("a+b*c+d=%d\n" , S(a+b,c+d)); //此时会打印出 a+b*c+d=29
 }
```

带参宏定义执行时是将 a+b 这样一个表达式代替 x，c+d 这样一个表达式代替 y，所以 S(a+b,c+d)进行的是 a+b*c+d 的运算，而不是将 a+b 的值给 x，c+d 的值给 y 然后再做 x*y，*这点跟函数调用传递参数是不一样的*。

十五、　结构体，共用体，枚举类型

- **结构体**
  - **i.　结构体类型的定义及变量的定义**

```
struct   结构体名
 {类型  成员 1;
  类型  成员 2;
  ……
  类型  成员 n;
 }变量名;
```

如

```
struct student
{   long   num;
    char name[10];
    int score[4];
}st1;定义类型时同时定义变量
struct student st2;定义类型后，用类型名定义变量
```

还有一种

```
struct
{   long num;
    char name[10];
    int score[4];
}st3;不给类型名，直接定义变量
```

  - **ii.　结构体变量所占字节数：*各成员各占字节数之和*，如以上 st1,st2,st3 的字节数皆为 10+20+2*4=38**
  - **iii.　结构体变量初始化：struct student a={20030001, "张三", 70,75,80,85};**
  - **iv.　结构体数组定义及初始化：**

```
struct student a[3]={{20030001,"zhang",89,90,91,92},
```

{20030002,"liu",68,69,70,71},
{20030003,"li",57,58,59,60} };

v. 结构体变量成员的访问

结构体变量名. 成员名 如 st1.num

- 共用体
  - **i. 共用体类型的定义及变量的定义**

union 共用体名
{ 类型 成员名 1;
…
类型 成员名 n;
};
变量的定义与结构体类似，也有三种方法。
union data
{
int i;
char ch;
float f;
}d1; 定义类型时同时定义变量
union data d2; 定义类型后，用类型名定义变量
union
{
int i;
char ch;
float f;
}d3; 不给类型名，直接定义变量

ii. 共用体变量所占字节数：*各成员所占字节数的最大值*,如上 d1,d2,d3 所占字节数皆为 4.（单精度浮点型变量所占字节数最多为 4）.

iii. 共用体变量成员的访问

1. 共用体变量名. 成员名 如 d1.i

2. 使用共用体类型的指针：

(*共用体指针名) . 成员 或 共用体指针名–>成员名

union data *dd; dd=&d1; dd->i 或(*dd).i

iv. 共用体成员的值：由于共用体各成员共用同一段内存区，故同一时刻只有一个成员的值是正确的。如 d1.i=5;d1.ch='a';则此时 d1.i 的值就不是 5 了，而是其他值了，d1.ch 的值是'a'

十六、 文件

1. 概述

文件是存储在外部介质的数据结合。从组织形式上，C 文件可分为文本文件和二进制文件。对文件的处理主要是指对文件进行读写操作。文件的读写方式有顺序读写和随机读写。

2. 文件的操作顺序

(1) 定义文件指针；
(2) 打开文件，并判断是否成功打开，若打开文件失败，程序退出运行状态；
(3) 对文件进行读写等操作；
(4) 关闭文件。

根据以上步骤，可写出对文件操作的常用代码：

```
FILE *fp;                    // 定义文件指针
if(fp=fopen("1.dat","r"))= =NULL)// 以"只读"方式打开，并判断是否成功打开
{       printf("Can't open this file!"); //打开失败，退出运行
        getch();
        exit(0);
}
文件的读写等操作;
 fclose(fp);                                    //关闭文件
```

3. 本章所介绍的文件相关函数汇总。

常用文件相关函数汇总

类别	函数名	常用格式	功能	返回值
打开	fopen	fp=fopen("f.txt","r");	以指定方式打开指定文件	成功时返回文件指针，否则返回空指针NULL
关闭	fclose	fclose(fp);	关闭 fp 指向的文件并释放文件缓冲区	成功时返回 0，否则返回 EOF
读写	fgetc	ch=fgetc(fp);	从 fp 指向的文件中读取一个字符	成功时返回所得字符，否则返回 EOF
	fputc	fputc(ch,fp);	将字符 ch 输出到 fp 指向的文件中	成功时返回该字符，否则返回 EOF
	fgets	fgets(str,n,fp);	从 fp 指向的文件中读取一个长度为 n-1 的字符串，存入起始地址为 str 的空间	成功时返回地址 str，遇文件结束或出错返回 NULL
	fputs	fputs(str,fp);	将 str 指向的字符串输出到 fp 指向的文件中	成功时返回非负值，否则返回 EOF
	fscanf	fscanf(fp,"%d",&a);	从 fp 指向文件中读入一个整数存放在变量 a 中	成功时返回已读入数据个数，否则返回 EOF
	fprintf	fprintf(fp,"%d",a);	把变量 a 的值以"%d"格式输出到 fp 指向的文件	成功时返回实际输出的字符数；失败时返回负值。

重要算法：

1. 求和，如 1+2+3+….+n
2. 求阶乘，如 5！等。
3. 找最大（小）数、排序（选择 冒泡）
4. 判断素数