目录

第一章 C 语言的基础知识

第一节 C 语言知识初步

1、C 语言编写的程序称为**源程序**，又称为**编译单位**。
2、C 语言书写格式是自由的，每行可以写多个语句，可以写多行。
3、一个 C 语言程序有且只有一个 main 函数，是程序运行的起点。
4、变量必须先定义后使用。
5、C 语言语句以分号结束；
6、每个 C 语言程序写完后，都是**先编译，后链接，最后运行**。（.c---→.obj---→.exe）这个过程中注意.c 和.obj 文件时无法运行的，只有.exe 文件才可以运行。（常考！）

第二节 标识符 常量 变量

VC 是软件，用来运行写的 C 语言程序，上机考试的系统为 VC6.0。
标识符（必考内容）：
合法的要求是由字母，数字，下划线组成。并且第一个必须为字母或则是下划线，不可以是数字。
2、标识符分为**关键字、预定义标识符、用户标识符**。
关键字：不可以作为用户标识符号，都是小写。main、define、scanf、printf 都不是关键字。
迷惑你的地方 If 是可以做为用户标识符。因为 If 中的第一个字母大写了，所以不是关键字。
预定义标识符：背诵 define scanf printf include。记住预定义标识符可以做为用户标识符。
用户标识符：基本上每年都考，详细请见书上习题。
常量：
整形常量 （1）十进制整形常量 如 7
（2）八进制整形常量 如：07
(3)十六进制整形常量 如 ox7
小数的合法写法：**C 语言小数点两边有一个是零的话，可以不用写。**
a、1.0 在 C 语言中可写成 “1. ”
b、0.1 在 C 语言中可以写成“ .1”
2) 实型整形常量：
a、2.333e-1 就是合法的，且数据是 2.333×10^{-1}。

6、考试口诀：e 前 e 后必有数，e 后必为整数。请结合书上的例子。

long int x; 表示 x 是长整型。

unsigned int x; 表示 x 是无符号整型。

3）字符常量　用单引号括起来的单个字符；

字符：有单单和转义字符之分。

①字符数据的合法形式:：单单（单引号里面单个字符）

'0' 的 ASCII 数值表示为 48，'a' 的 ASCII 数值是 97，'A'的 ASCII 数值是 65。

一般考试表示单个字符错误的形式：'65'　"1"　记住口诀：单单。

字符是可以进行算术运算的，记住：　'0'-0=48

大写字母和小写字母转换的方法：　'A'+32='a'　相互之间一般是相差 32。

②转义字符：单引号里面用 \ 加上另外字母形成新的组合。

转义字符分为一般转义字符、八进制转义字符、十六进制转义字符。

一般转义字符：背诵\0、 \n、 \'、 \"、 \\。

八进制转义字符：　'\141' 是合法的，　前导的 0 是不能写的。

十六进制转义字符：'\x6d' 才是合法的，前导的 0 不能写，并且 x 是小写。

字符串常量　用双引号括起来的若干个字符；

符号常量　用一个符号表示一个固定的值；

第三节 数制转换

1、十进制转换成二进制、八进制、十六进制。

2、二进制、八进制、十六进制转换成十进制。

3、C 语言中只有八、十、十六进制，没有二进制。但运行时，所有进制都要转成二进制来处理。

a、C 语言中的八进制规定要以 0 开头。018 的数值是非法的，八进制不可以出现 8。

b、C 语言中的十六进制规定要以 0x 开头。要看懂 0xff。

第四节 变量及数据类型

字符型变量　　如 char　x='a'　字符型变量占一个字节

短整形变量　　如 short　x=2　占用 2 个字节

基本整形 变量　　如 int　x==2　占用 4 个字节

长整形变量　　如　long　x=2　占用 4 个字节

单精度实型变量　　如　float　x=2.4　占用 4 个字节

双精度实行变量　　如 double x=2.4　　占用 8 个字节

注意：字符型和整数是近亲：两个具有很大的相似之处

第五节　表达式　变量复制　类型转换

核心：表达式一定有数值！

1、算术表达式：+，-，*，/，%　，考试重点为　/　和　%这两个。

考试一定要注意：“/”　两边都是整型的话，结果取整。　3/2 的结果就是 1.

“/”　如果有一边是小数，结果为小数。　3/2.0 的结果就是 0.5

“%”符号请一定要注意是余数，考试最容易算成了除号。

“%”符号两边要求是整数。不是整数就错了。

2、赋值表达式：赋值表达式的结果是最左边的数值，a=b=5;该表达式为 5，常量不可以赋值。

1、int x=y=10:；错啦，定义时，不可以连续赋值。

2、int x,y;

x=y=10；　　对滴，定义完成后，可以连续赋值。

3、int x=7.7；　对滴，x 就是 7。

4、float y=7；　对滴，x 就是 7.0。

5、赋值的左边只能是一个变量。x+y=10;这个写法是错的。

3、复合的赋值表达式：

int a=2；

a*=2+3；运行完成后，a 的值是 12。

一定要注意，首先要在 2+3 的上面打上括号。变成（2+3）再运算。

4、自加表达式：

自加、自减表达式：假设 a=5，++a（是为 6），　a++（为 5）；

考试口诀：++在前先加后用，++在后先用后加。

5、逗号表达式：优先级别最低（表达式的数值逗号最右边的那个表达式的数值）

（2，3，4）的表达式的数值就是 4。取最右边的值。

z=（2，3，4）(整个是赋值表达式)　这个时候 z 的值为 4。

z=　2，3，4　(整个是逗号表达式)　这个时候 z 的值为 2。

6、赋值　变量=表达式

注意：等号左边一定是一个变量。

7、类型转换

1）自动转换　如 int a=3.6

2)强制转换　如(int)3.6

强制类型转换：

a、一定是(int)a 不是 int (a)，注意类型上一定有括号的。

b、注意(int) (a+b)　和(int)a+b　的区别。　前是把 a+b 转型，后是把 a 转型再加 b。

补充：

4、三种取整丢小数的情况：

1 ）int a =1.6；　　2 ）　(int)a；　　3 ）1/2；　3/2；

结果 a 为 1　　整个数值取整　　0 和 1

第六节 自加自减运算符和逗号运算符

1、i++，先使用 i 的值，使用完了之后 i 再自动加一（即 i=i+1）。

2、++i，i 先自动加一（即 i=i+1），然后再使用 i 的值。

3、i--,和—i 同上。

4、逗号运算符的级别最低，从左向右依次运算即可。

第七节 位运算

1）位运算的考查：会有一到二题考试题目。（必考题型）

例 1：　char a = 6, b; b = a<<2;　解题时：先要把 a 化成二进制，再做位运算。
例 2：　一定要记住，异或的位运算符号”^ ”。0 ^ 1=1。0 ^ 0=0。
例 3：　在没有舍去数据的时候，<<左移一位表示乘以 2；>>右移一位表示除以 2。

第二章 顺序结构

第一节结构化程序设计

1、 结构化程序是由顺序结构、选择结构(分支结构)、循环结构三大结构组成。

第二节 语句

1、 表达式语句　如 3—5；
　　空语句　　如 ；. 空语句不可以随意执行，会导致逻辑错误。
2、 复合语句　　如{x=1;x++;}
3、 注释 ,注释是最近几年考试的重点，注释不是 C 语言，不占运行时间，没有分号。不可以嵌套！

第三节　数据输出

1、使用 printf 和 scanf 函数时，要在最前面加上#include “stdio.h”
2、printf 可以只有一个参数，也可以有两个参数。（选择题考过一次）这么背就可以。

3、printf（“ 第一部分 ”，第二部分 ）；

把第二部分的变量、表达式、常量以第一部分的形式展现出来！

4、printf（“a=%d，b=%d”，12, 34） 考试重点！

记住：是将第二部分的 12 和 34 以第一部分的形式在终端（也就是黑色的屏幕上）显示。

考试核心为：一模一样。在黑色屏幕上面显示为 a=12，b=34

printf（“a=%d，\n b=%d”，12, 34）那么输出的结果就是：a=12，
b=34

5、int x=017； （一定要弄清楚为什么是这个结果！过程很重要）

printf（“%d”， x）； 15 printf（“%o”， x）； 17

printf（“%#o”，x）； 017 printf（“%x”， x）； f

printf（“%#x”，x）； 0xf

6、一定要背诵的

格式说明	表示内容	格式说明	表示内容
%d	整型 int	%c	字符 char
%ld	长整型 long int	%s	字符串
%f	浮点型 float	%o	不带前导 0 八进制
%lf	浮点型 double	%#o	带前导 0 的八进制
%%	输出一个百分号	%x	不带前导 0x 十六进制
%5d	输出要求有五位	%#x	带前导 0x 的十六进制

7、举例说明：

printf（“%2d”，123 ）； 第二部分 123 有三位，大于第一部分指定的两位，原样输出 123

printf（“%5d”，123 ）； 第二部分 123 有三位，小于第一部分指定的五位，左边补两个空格 123

printf（“%10f”，1.25 ）； 小数要求补足 6 位的，没有六位的补 0,。结果为 1.250000

printf（“%5.3f”，1.25 ）； 第一部分指定小数三位，整个五位，结果为 1.250（小数点算一位）

printf（“%3.1f”，1.25 ）； 第一部分指定小数一位，整个三位，结果为 1.3（要进行四舍五入）

第四节 数据输入

1、scanf（“a=%d，b=%d”，&a，&b） **（考试超级重点）**

考试核心为：一模一样。以第一个部分双引号里面为输入标准形式。

终端输入为：a=12，b=34 才可把 12 和 34 正确赋值给 a 和 b。

2、scanf（“%d，%d”，x，y）； scanf 的第二个部分一定要是地址（或是指针变量）！

scanf（“%d，%d”，&x，&y）；**注意写成这样正确！**

3、特别注意指针在 scanf 的考察（近几年重点）

例如： int x=2；int *p=&x；

scanf（“%d”，x）； 错误 scanf（“%d”，p）；正确

scanf（“%d”，&p）； 错误 scanf（“%d”，*p）错误

4、指定输入的长度 （考试重点）

终端输入：1234567

scanf（“%2d%4d%d”，&x，&y，&z）； x 为 12，y 为 3456，z 为 7

终端输入：1 234567 由于 1 和 2 中间有空格，所以只有 1 位给 x

scanf（“%2d%4d%d”，&x，&y，&z）； x 为 1 ，y 为 2345，z 为 67

5、字符和整型是近亲：

int x=97；

printf（“%d”，x）；　　结果为 97

printf（“%c”，x）；　　结果为 a

6、输入时候字符和整数的区别（考试超级重点）

scanf（“%d”，&x）；这个时候输入 1，特别注意表示的是整数 1

scanf（“%c”，&x）；这个时候输入 1，特别注意表示的是字符’1’，ASCII 为整数 48。

7、补充说明：

1）scanf(“%d%d%*d%d”,&a,&b,&c); 跳过输入的第三个数据。

2）putchar ,getchar 函数的考查。前是输出一个字符，后是获得一个字符。

3）交换两个数 t=x；x=y；y=t。当成单词去背。

第三章　选择结构

第一节　关系运算符与逻辑运算

1、逻辑值

特别要注意：1、C 语言中是用非 0 表示逻辑真，0 表示逻辑假的。

2、C 语言有构造类型，没有逻辑类型。

3、关系运算符号：注意<=的写法，==和=的区别！（考试重点）

4、if 只管后面一个语句，要管多个，请用大括号！

2、关系运算符及关系表达式

1）关系运算符　〉，〉=，＜,<=,==,!=

2）关系表达式：

a、表达式的数值只能为 1（表示为真），或 0（表示假）。

如 9>8 这个关系表达式是真的，所以 9>8 这个表达式的数值就是 1。

如 7<6 这个关系表达式是假的，所以 7<6 这个表达式的数值就是 0

b、考试最容易错的：就是 int x=1,y=0,z=2;x<y<z 是真还是假？带入为 1<0<2，从数学的角度出发肯

定是错的，但是如果是 C 语言那么就是正确的！因为要 1<0 为假得到 0，表达式就变成了 0<2 那

么运算结果就是 1，称为了真的了！

c、等号和赋值的区别！一定记住“=”就是赋值，“= =”才是等号。做错了，我一定会强烈鄙视你！

3、逻辑表达式：

共有&&　||　！三种逻辑运算符号

核心：表达式的数值只能为 1（表示为真），或 0（表示假）。

a、注意短路现象。考试比较喜欢考。详细请见书上例子，一定要会做例 1 和例 2。

b、表示 x 小于 0 大于 10 的方法。（考试非常容易错的）

0<x<10 是不行的（一定记住），他永远为真。(0<x)&&(x<10)才是正确表示方法。

a、else 是与最接近的 if 且没有 else 的语句匹配。

b、交换的程序写法：t=x；x=y；y=t；

c、if（a<b）t=a;a=b;b=t;

if（a<b）{t=a;a=b;b=t;}　两个的区别，考试多次考到了！

d、单独的 if 语句：if（a<b）t=a；

标准的 if 语句：if（a<b）min=a；

else　min=b；

嵌套的 if 语句：if（a<b）

if（b>c）printf("ok!")；

多选一的 if 语句 if（a= =t）printf("a");

else if（b= =t）printf("b");

else if（c= =t）printf("c");

else pritnf("d");

通过习题，要熟悉以上几种 if 语句！

4）条件表达式： 表达式 1 ？表达式 2 ：表达式 3

a、考试口诀：真前假后。

b、int a=1，b=2，c=3，d=4，e=5；

k=a>b？c：d>e？d：e;　　求 k 的数值时多少？　答案为 5

5）switch 语句：（考试重点）

a) 执行的流程一定要弄懂！上课时候详细的过程讲了，请自己一定弄懂!

b）注意有 break 和没有 break 的差别，break 在 C 语言中就是分手，一刀两断的意思。

c) switch 只可以和 break 一起用，不可以和 continue 用。

d) switch(x)　x 是整型常量，字符型常量，枚举型数据。

{case 1: ….　不可以是变量。

case 2: ….

}

e）switch 是必考题型，请大家一定要完成书上的课后的 switch 的习题。

6）goto 语句，是关键，不可以随便使用，已经淘汰。

第四章 循环结构

1）三种循环结构：

a）for（）；　while()；　do- while() 三种循环。

b）for 循环当中必须是两个分号，千万不要忘记。

c）写程序的时候一定要注意，循环一定要有结束的条件，否则成了死循环。

d）do-while()循环是至少执行一次循环。

2) break 和 continue 的差别（考试重点，一定会考）

break：　是打破的意思，（破了整个循环）所以看见 break 就退出一层循环。

continue：是继续的意思，（继续循环运算），但是要结束本次循环，就是循环体内剩下的语句不再执行，跳到循环开始，然后判断循环条件，进行新一轮的循环。

3）嵌套循环

就是有循环里面还有循环，这种比较复杂，要一层一层耐心计算，一般记住两层是处理二维数组的。

4) while（（c=getchar()）!='\n'） 和 while（c=getchar() !='\n'）的差别

先看 a = 3 != 2 和 （a=3）！=2 的区别：**考试注意点：**括号在这里的重要性。

（！=号的级别高于=号 所以第一个先计算 3！=2） 第一个 a 的数值是得到的 1；第二个 a 的数值是 3。

5）如何整除一个数： i%5==0 表示整除 5　　i%2==0 表示整除 2，同时表示是偶数！

6）输入 123，输出 321 逆序输出数据

```
int a=123;
while（i！=0）
{
  printf（“%d”，i%10）；
  i=i/10;}
```

7)for 只管后面一个语句：

```
int i=3；
for（i=3；i<6;i++）  ；          循环控制这个空语句，空语句是循环体！
printf(“#”)：                  请问最终打印几个#号？答案为一个！
```

8）不停的输入，直到输入# 停止输入！

```
while( (x=getchar())!=’#’)
```

不停的输入，直到输入$停止输入！

```
while( (x=getchar())!=’$’)
```

不停的输入，直到遇到？停止输入！

```
while( (x=getchar())!=’?’)
```

解说：一定要注意这种给出了条件，然后如何去写的方法！

9）for 循环和 switch 语句的和在一起的考题！

10）多次出现的考题：（超级重点，一定会考）

```
int k=1                          int k=1；
while（- -k）；                    while（k- -）；
printf（“%d”，k）；                  printf（“%d”，k）；
```

结果为 0　　结果为-1

第五章 函数

1、函数：是具有一定功能的一个程序块，是 C 语言的基本组成单位。

2、函数不可以嵌套定义。但是可以嵌套调用。

3、函数名缺省返回值类型，默认为 int。

4、C 语言由函数组成，但有且仅有一个 main 函数！是程序运行的开始！

5、如何判断 a 是否为质数：背诵这个程序！

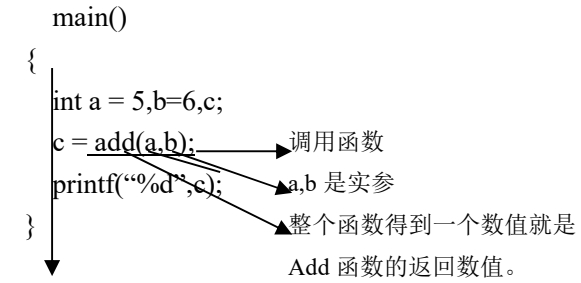
```
void   iszhishu（ int a ）
{   for（i=2；i<a/2；i++）
    if(a%i==0) printf（“不是质数”）；
    printf(“是质数！”);
}
```

6、如何求阶层：n! 背诵这个程序！

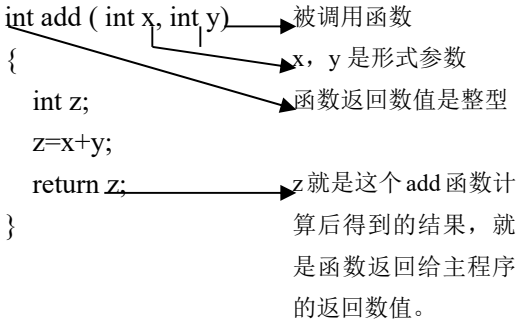
```
int   fun(int n)
{   int p=1;
    for(i=1;i<=n;i++) p=p*i;
    return p;
}
```

7、函数的参数可以是常量，变量，表达式，甚至是函数调用。

8、 函数的参数，返回数值（示意图）：



程序是在从上往下顺序执行，当碰到了函数 add 后，把 a，b 的数值穿给调用函数，程序暂时中断等待返回数值。当得到了返回数值后，再顺序的往下执行



9、一定要注意参数之间的传递。实参和形参之间传数值，和传地址的差别。（考试的重点）

传数值的话，形参的变化不会改变实参的变化。

传地址的话，形参的变化就 98%会改变实参的变化。

10、函数声明的考查：

一定要有：函数名，函数的返回类型，函数的参数类型。

不一定要有：形参的名称（可写，可不写，可乱写）。

11、要求掌握的库函数：

abs()、sqrt()、fabs()、pow()、sin()其中 pow(a，b)是重点。2^3 是由 pow(2，3)表示的。

第六章　指针

指针变量的本质：放地址。　变量三要素：名称、内容、地址。

1、int　*p　中　　*p 和 p 的差别：简单说*p 是数值，p 是地址！

*p 可以当做变量来用，*的作用是取后面地址 p 里面的数值

p 是当作地址来使用。可以用在 scanf 函数中：scanf（“%d”，p）；

2、*p++ 和 （*p）++的差别：(考试重点)

*p ++　是地址会变化。　　口诀：取当前值，然后再移动地址！

(*p)++ 是数值会变化。　　口诀：取当前值，然后再使数值增加 1。

例题：　int *p,a[]={1,3,5,7,9};

p=a;

请问*p++和(*p)++的数值分别为多少？

*p++：　本身为 3，然后挪到下个地址，地址变动。

(*p)++：　本身为 3，然后再把 3 变成 4，地址不动。

3、二级指针：

*p：一级指针：存放变量的地址。

**q：二级指针：存放一级指针的地址。

常考题目：　int x=7;

int*p=&x，**q=p;

问你：*p 为多少？*q 为多少？**q 为多少？

7　　p　　7

再问你：**q=&x 的写法可以吗？

不可以，因为二级指针只能存放一级指针的地址。

4、三名主义：（考试的重点）

数组名：表示第一个元素的地址。**数组名不可以自加，他是地址常量名。（考了很多次）**

函数名：表示该函数的入口地址。

字符串常量名：表示第一个字符的地址。

5、移动指针（经常加入到考试中其他题目综合考试）

char *s=“meikanshu”

while（*s）{printf（“%c”，*s）；s++；}

s++是地址移动，打印了一个字母后，就会移动到下一个字母！

6、指针变量两种初始化（一定要看懂）

方法一：int a=2，*p=&a；（定义的同时初始化）

方法二：int a=2，*p；　（定义之后初始化）

p=&a；

7、传数值和传地址（每年必考好多题目）

```
void fun（int a，int b）
{ int t ；
  t=a；a=b；b=t；
}
main（）
{ int x=1，y=3，
  fun（x，y）；
  printf（“%d，%d”，x，y）；
}
```

这个题目答案是 1 和 3。

传数值，fun 是用变量接受，所以 fun 中的交换不会影响到 main 中的 x 和 y 。

传数值，形参的变化不会影响实参。

```
void fun（int *a，int *b）
    { int t ；
      t=*a；*a=*b；*b=t；
    }
main（）
{ int x=1，y=3，
   fun（&x，&y）
     printf（“%d，%d”，x，y）；
 }
```

这个题目的答案就是 3 和 1。

传地址，fun 用指针接受！这个时候 fun 中的交换，就会影响到 main 中的 x 和 y。

传地址形参的变化绝大多数会影响到实参！

8、函数返回值是地址，一定注意这个*号（上机考试重点）

```
int *fun（int *a，int *b）
{ if（*a>*b）return a；
  else return b；
}
```

可以发现函数前面有个*，这个就说明函数运算结果是地址

return a 可以知道返回的是 a 地址。

```
main（）
{ int x=7，y=8，*max；
  max = fun（&x，&y）；
  printf（“%d，%d”，）
}
```

由于 fun（&x，&y）的运算结果是地址，所以用 max 来接收。

9、考试重要的话语：

指针变量是存放**地址**的。并且指向哪个就等价哪个，所有出现*p 的地方都可以用它等价的代替。

例如：int a=2，*p=&a；

*p=*p+2；

（由于*p 指向变量 a，所以指向哪个就等价哪个，这里*p 等价于 a，可以相当于是 a=a+2。

第七章　一维数组

数组：　像停尸房一样，一格一格的！地址连续，类型一致。

1、一维数组的初始化：

int a[5]={1,2,3,4,5};　合法

int a[5]={1,2,3,};　　合法

int a[]={1,2,3,4,5};　合法,常考，后面决定前面的大小！
int a[5]={1,2,3,4,5,6}; 不合法，赋值的个数多余数组的个数了

2、一维数组的定义；

int a[5]；重要考点，定义数组不可以是变量。

int x=5,int a[x]; 不合法，因为个数是 x，是个变量，非法的，
define P 5　int a[P] 合法，define 后的的 P 是符号常量，只是长得像变量

第八章 二位数组

3、二维数组的初始化：

int a[2][3]={1,2,3,4,5,6}; 合法，
int a[2][3]={1,2,3,4,5, }; 合法，后面一个默认为 0。
int a[2][3]={{1,2,3,} {4,5,6}}; 合法，
int a[2][3]={{1,2,}{3,4,5}}; 合法，
int a[2][3]={1,2,3,4,5,6,7}; 不合法，赋值的个数多余数组的个数了。
int a[][3]={1,2,3,4,5,6}; 合法，可以缺省行的个数。
int a[2][]={1,2,3,4,5,6}; 不合法，不可以缺省列的个数。

4、补充：

1）数组的重要概念：

a[10]的讨论。（一维数组的讨论）

1、a 表示数组名，是第一个元素的地址，也就是元素 a[0]的地址。（等价于&a）
2、a 是地址常量，所以只要出现 a++，或者是 a=a+2 赋值的都是错误的。
3、a 是一维数组名，所以它是列指针，也就是说 a+1 是跳一列。

a[3][3]的讨论。（二维数组的讨论）

1、a 表示数组名，是第一个元素的地址，也就是元素 a[0][0]的地址。
2、a 是地址常量，所以只要出现 a++，或者是 a=a+2 赋值的都是错误的。
3、a 是二维数组名，所以它是行指针，也就是说 a+1 是跳一行。
4、a[0]、a[1]、a[2]都是地址常量，不可进行赋值操作，同时它们都是列指针，a[0]+1，a[1]+1，a[2]+1 都是跳一列。
5、注意 a 和 a[0]、a[1]、a[2]是不同的，它们基类型是不同的。前者是一行元素，后三者是一列元素。

2) 二维数组做题目的技巧：

如果有 a[3][3]={1,2,3,4,5,6,7,8,9}这样的题目。
步骤一：把他们写成：

	第一列	第二列	第三列	
a[0]→	1	2	3	—>第一行
a[1]→	4	5	6	—>第二行
a[2]→	7	8	9	—>第三行

步骤二：这样作题目间很简单：
*(a[0]+1)我们就知道是第一行的第一个元素往后面跳一列，那么这里就是 a[0][1]元素，所以是 2。
*(a[1]+2)我们就知道是第二行的第一个元素往后面跳二列。那么这里就是 a[1][2]元素，所以是 6。
一定记住：只要是二维数组的题目，一定是写成如上的格式，再去做题目，这样会比较简单。

3) 数组的初始化，一维和二维的，一维可以不写数字，二维第二个一定要写出列，可以不写行。

int a[]={1，2} 合法。　int a[][4]={2，3，4}合法。　但 int a[4][]={2，3，4}非法。

4) 二维数组中的行指针：二位数组名是行指针。

int a[1][2]；
其中 a 现在就是一个行指针，a+1 跳一行数组元素。 **搭配(*)p[2]指针**
a[0]，a[1]现在就是一个列指针。a[0]+1 跳一个数组元素。**搭配*p[2]指针数组使用**

5) 还有记住脱衣服法则：（超级无敌重要）

a[2] ~~—变成→~~*(a+2)　a[2][3]~~—变成→~~*(a+2) [3]~~—再可以变成→~~*(*(a+2)+3)
这个思想很重要！

第九章 字符串

1）字符串的 strlen()和 strcat()和 strcmp()和 strcpy()的使用方法一定要记住。他们的参数都是地址。

strlen()： 统计字符串长度，以'\0'，为结束条件。	strlen("wo\0guale") 长度为 2. strlen("woguale") 长度为 7
strcat()： 连接两个字符串。	
strcpy()： 把后面的复制到前面	strcat("wo\0ai","heshang") 连接后新的字符串为：woheshang
strcmp()： 比较两个字符串中对应字符的 ASCII。	strcat("abcd","abbbb") 运行结果是>0 的口诀：大大小小，不是比长度

2）strlen 和 sizeof 的区别也是考试的重点；

3）define f(x) (x*x)和 define f(x) x*x 之间的差别。替换的时候有括号和没有括号是考试超级重点。

4）int *p;

p = （int *）malloc（4）；

p = （int *）malloc（sizeof（int））；以上两个等价。

当心填空题目，malloc 的返回类型是 void *

6）函数的递归调用一定要记得有结束的条件，并且要会算简单的递归题目。要会作递归的题目

7）链表中的节点是有两个域，**一个放数值，一个放指针。**

8）函数指针的用法：口诀：使用时，有*就有括号。(*f) (3)； 这个括号一定要有。

9）两种重要的数组长度：

char a[]={'a','b','c'}; 数组长度为 3，字符串长度不定。sizeof(a)为 3 。

char a[5]={ 'a','b','c'} 数组长度为 5，字符串长度 3 。sizeof(a)为 5 。

10）scanf 不可以处理空格，gets 可以处理。

11）指针迷惑的考点：超级重点！

char ch[]="iamhandsome";

char *p=ch;

问你 *(p+2) 和 *p+2 的结果是多少？

'm' 'k' 结果是这两个，想不通的同学请作死的想！想通为止！

12）数组中放数组一定要看懂:

int a[8]={1,2,3,4,4,3,2,2};

int b[5]={0};

b[a[3]]++ 这个写法要看懂，结果要知道是什么？b[4]++,本身是 0，运行完后，b[4]为 1 了。

15）字符串的赋值：C 语言中没有字符串变量，所以用数组和指针存放字符串：

1、char ch[10]={"abcdefgh"}; 对

2、char ch[10]="abcdefgh"; 对

3、char ch[10]={'a','b','c','d','e','f','g','h'}; 对

4、char *p="abcdefgh"; 对

5、char *p; 对

p="abcdefgh";

6、char ch[10]; 错了！数组名不可以赋值！考试重点

ch="abcdefgh";

7、char *p={"abcdefgh"}; 错了！不能够出现大括号！

16）字符串赋值的函数背诵：一定要背诵，当心笔试填空题目。

把 s 指针中的字符串复制到 t 指针中的方法

1、while（ （*t=*s）！=NULL ）{s++；t++；} 完整版本

2、while（ *t=*s ）{s++；t++；} 简单版本

3、while（ *t++=*s++）； 高级版本

17）typedef 是取别名，不会产生新的类型，他同时也是关键字

考点一：typedef int qq 那么 int x 就可以写成 qq x

考点二：typedef int *qq 那么 int *x 就可以写成 qq x

18）static 考点是一定会考的！复习相关的习题。

static int x；默认值为 0。

int x：默认值为不定值。

19）函数的递归调用一定会考！至少是 2 分。

第十章 对 C 语言的深入讨论

1、编译预处理包括三个方面：

1）不带参数的宏 如#define PI 3.14

2）带参数的宏 如 #define MUL(a,b) ((a)*(b))

….

M=MUL(10-5,5+1);

3）文件包含

2、变量存储分类

Auto register static

3 全局变量

4、动态存储分配 1）int *p; p=(int*)malloc(4); 或者 int *p; p=(int*)malloc(sizeof(int));

2) int *p; p=(int*)calloc(10,sizeof(int)):;

5、函数指针

```
int fun(int a , int * b)
{
    …
}
main( )
{   int (*pfun )( int, int*), a, b, s;        /*此处定义了 pfun 这个函数指针*/
    pfun=fun;                                 /*将函数名 fun 赋给指针 pfun，即让 pfun 指向
fun*/
    …
    s=(*pfun)(a, &b);                         /*此处通过 pfun 指针来调用 fun 函数，相当于调用
                                                s=fun(a,&b) */
   …
```

}

第一节　用户自定义类型、编译预处理

1、用户自定义类型 typedef

即使用 typedef 将用户自己定义的数据类型或现有的数据类型说明成一个简介的数据类型。

2、 编译预处理

1） 不带参数的宏（即符号常量的定义）

2） 带参数的宏（注意表达式中括号的使用）

3） 文件包含

第二节 存储分类和变量的作用域

1、 存储分类：auto,register,static,extern

2、 局部变量

Auto,register,static

3、 全局变量

在函数外部任意位置定义的变量，称之为全局变量。

第三节 动态存储分配

1、 malloc(size)函数

2、 calloc(n,size)函数

3、 free(p)函数

第十一章　结构体与共用体

1、 结构体的定义

```
struct    student
{
char    name[10];
char    sex ;
short    age ;
float    score ;
};
```

2、 结构体类型变量

```
struct    student
{
char    name[10] ;
char    sex ;
```

```
int     age ;
float   score ;
}s1 , *ps , stu[3] ;
```
或者
```
    typedef   struct
    {
        char   name[10] ;
        char   sex ;
        int    age ;
        float  score ;
    } STU ;
    STU   s1 , *ps , stu[3] ;
```
3、结构体变量赋初值
```
struct   student
{
char   name[10] ;
char   sex ;
int    age ;
float  score ;
}s1={"Jim" , 'M' , 20 , 89} ;
```
或者
```
    struct   student
    {
        char   name[10] ;
        char   sex ;
        int    age ;
        float  score ;
    }stu[3]={{"Jim" , 'M' , 20 , 89} , {"Sam" , 'W' , 21 , 78} , {"Bill" , 'M' , 22 , 85}};
            /*     stu[0]                stu[1]                stu[2]   */
```
4、结构体变量中成员的引用

有以下 3 种形式来引用结构体成员：

（1）结构体变量名.成员名；

（2）结构体指针变量->成员名；

（3）(*结构体指针变量).成员名。

如
```
    struct   stu
    {
        char name[10] ;
        char sex ;
        struct date
        {
            int year ;
            int mon ;
            int day ;
        }birthday ;
```

```
        int age ;
        float grade[3] ;
        float score ;
    }s1 , *ps , s[3];
    ps=&s1;
```

则可以对其成员做如下引用：s1.age

```
ps->age
(*ps).age
```

要引用 s1 结构体变量的 birthday 的 year 成员，则可以这样写 s1.birthday.year，或 ps->birthday.year 和(*ps).birthday.year。

5 、共用体定义

```
union   un
{
    int   i ;
    char   c ;
    double   f ;
}u1 ;
```

5、 共用体变量的引用

```
union un_1
{
short   a ;
char   c ;
float   fd ;
}un,*pun=&un;
```

6、 共用体变量的引用

与结构体成员访问方式相同，共用体也可以通过以下方式引用共用体的成员：

（1）共用体变量名.成员名；

（2）共用体指针变量名->成员名；

（3）(*共用体指针变量名).成员名。

7、共用体变量赋初值

```
union   un
{
    int   i ;
    char   c ;
    float   f ;
}u1;
```

有以上定义后，可以按如下方式来引用 u1 的各成员：

```
u1.i=20;
u1.c=getchar( );
scanf("%f" , &un.f) ;
```

注：共用体变量同样要满足先定义后使用的规则，共用体变量在程序执行过程中的某一时刻，只有一个成员的值有效，即最近存入的成员变量的值，而原来的成员变量的值被覆盖。

例如：union   change

```
{
```

```
    char c[2] ;
    short   a ;
}un ;
main( )
{
    un.a=16961 ;
    printf("%d,%c\n" , un.c[0] , un.c[0]) ;
    printf("%d,%c\n" , un.c[1] , un.c[1]) ;
}
```

分析：本例中共用体变量 un 中包含两个成员：字符数组 c 和 short 型变量 a。它们恰好都占据两个字节的存储单元，字符数组中 c[0]在低 8 位，c[1]在高 8 位，c[0]的最低位与 a 的最低位对齐。我们首先将 16961 转换为二进制数 0100001001000001，然后按照每 8 位一个字节放入两个字节的存储空间内，如图 11-4 所示。

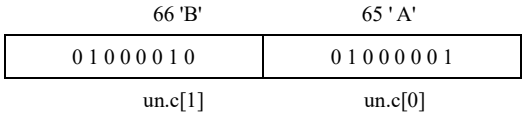


图 11-4

因此程序的最终运行结果为：

65 , A

66 , B

7、 链表的定义

例 1　一个简单的链表程序

```
#include<stdio.h>
struct link
{
char data ;
struct link *next ;
} ;
typedef struct link LINK ;
main( )
{
LINK a , b , c , *p ,*s;
a.data='A' ; b.data='B' ; c.data='C' ;
p=&a;
a.next=&b;
b.next=&c;
c.next=NULL;
s=p;
while(s!=NULL)                    /*  当 s 指向的是不为空  */
{
printf(" %c    " , s->data) ;     /*    输出 s 指向结构体变量的 data 成员*/
s=s->next ;                       /*     使 s 指向 s 的 next 成员指向的结构体变量*/
```

}
}
程序运行后结果为：
A B C

第十二章 文件

1、 文件指针：
文件指针的一般形式如下：
FILE *指针变量名;
例如：
FILE *fp,*fout,*fin;
2、 打开文件

```
if((fp=fopen("c:\\xiaoyuan.c","rb"))==NULL)
{
    printf(“打开文件失败！”);
}
```

3、 文件的打开方式
（1）"r"：以读的方式打开一个文本文件，当用这种方式打开文件时，对打开的文件只能进行读操作。如果指定的文件不存在或不允许读，函数返回 NULL。
（2）"rb"：以读的方式打开一个二进制文件。其余功能与"r"相同。
（3）"w"：以写的方式打开一个文本文件。如果指定的文件不存在，系统将用指定的文件名建立一个新文件；如果指定的文件已存在，则将从文件的起始位置开始写，文件中原有的内容将全部消失。
（4）"wb"：以写的方式打开一个二进制文件。其余功能与"w"相似，但从指定位置开始写。
（5）"a"：以添加数据的方式打开一个文本文件。如果指定的文件不存在，系统将用指定的文件名建立一个新文件；如果指定的文件已存在，则文件中原有的内容将不变，新的数据写在原有内容之后。
（6）"ab"：以添加数据的方式打开一个二进制文件。其余功能与"a"相同。
（7）"r+"：以读和写的方式打开一个文本文件。如果文件不存在，函数返回 NULL。用这种方式打开文件，可以对指定的文件进行读和写的操作。读和写总是从文件的起始位置开始。在写新的数据时，只覆盖数据所占的空间，其后的老数据并不丢失。
（8）"rb+"：以读和写的方式打开一个二进制文件。功能与"r+"相同。只是在读和写时，可以由位置函数设置读和写的起始位置，也就是说不一定从文件的起始位置开始读和写。
（9）"w+"：使用这种方式打开文件将建立一个新文件，如果指定的文件已存在，则原有的内容将全部消失。然后可以对文件进行读和写的操作。
（10）"wb+"：功能与"w+"相同，只是在读和写时，可以由位置函数设置读和写的起始位置。
（11）"a+"：功能与"a"相同，只是在文件尾部添加新的数据之后，可以从头开始读。
（12）"ab+"：功能与"a+"相同，只是在文件尾部添加新的数据之后，可以由位置函数设

置开始读的起始位置。

4、文件的关闭

fclose(fp);

4、 文件操作

1）C 语言提供了库函数 feof，用来判断文件是否结束。feof 函数的一般调用形式如下：

feof(文件指针);

如果文件结束，函数返回 1，否则返回 0。

2）fseek 函数用来设置文件的位置指针的位置，接着的读或写操作将从此位置开始。函数的调用形式如下：

fseek(文件指针, 位移量, 移动起始点);

FILE *fp=fopen("c:\\xiaoyuan.c","rb");

则

fseek(fp, 5, SEEK_SET);

表示将文件 fp 的位置移动到文件起始位置后面的第 5 个字节的位置上；

fseek(fp, -5, SEEK_END）;

表示将文件 fp 的位置移动到文件末尾位置之前的第 5 个字节的位置上。

对于文本文件，位移量必须为 0。例如：

FILE *fp=fopen("c:\\xiaoyuan.c","r");

5、fscanf 函数和 fprintf 函数（绝对的重点）

fscanf 函数只能从文本文件中按格式输入。fscanf 函数和 scanf 函数相似，只是输入的对象是磁盘上文本文件中的数据。函数的调用形式如下：

fscanf(文件指针, 格式控制字符串, 输入项表);

例如，若文件指针 fp 已指向一个已打开的文本文件，a、b 分别为整型变量，则以下语句从 fp 所指的文件中读入两个整数放入变量 a 和 b 中：

fscanf(fp, "%d%d",&a,&b);

注意：文件中的两个整数之间用空格（或跳格符、回车符）隔开。

fprintf 函数按格式将内存中的数据转换成对应的字符，并以 ASC II 代码形式输出到文本文件中。fprintf 函数和 printf 函数相似，只是输出的内容将按格式存放在磁盘的文本文件中。函数的调用形式如下：

fprintf(文件指针, 格式控制字符串, 输出项表)

例如，若文件指针 fp 已指向一个已打开的文本文件，x、y 分别为整型变量，则以下语句将把 x 和 y 两个整型变量中的整数按%d 格式输出到 fp 所指的文件中：

fprintf(fp, "%d %d",x,y);

注意：为了便于以后读入，两个数之间应当用空格隔开。同时也是为了便于读入，最好不要输出附加的其他字符串。

背诵的内容

1、取别名：可能考　　2、预处理：一定会考　　3、全局变量：可能考　　4、static：一定会考

5、分配对象：可能会考　　6、函数指针：可能会考　　7、递归调用：一定会考

单　　单：单引号里面有单个！是合理的字符。字符是单引号‘a’，字符串是双引号“a”。

两大近亲：　1、整型和字符型　　2、数组和指针

三名主义：核心纲领：名称表示地址

1、数组名：　表示第一个数组元素的地址。

2、字符串名：表示第一个字符的地址。

3、函数名：　表示函数入口地址。

脱衣服法则：a[2]　变成　*(a+2)　　a[2][3]　变成　*(a+2) [3]　再可以变成*(*(a+2)+3)

ASCII：　　每一个字符对应一个数字。‘a’为 97　　‘A’为 65　　‘0’为 48

以上内容请大家反复复习，认真的复习哈！

人生，最主要是**态度认真、用对方法、做到坚持**。这个无论是你们以后考研、找工作、创业、为人处世、谈恋爱，追寻自己幸福都是这样的。