# C语言开卷总结资料

目录

[C语言开卷总结资料 1](#_Toc44327542)

[ASCII码表 2](#_Toc44327543)

[运算优先级 3](#_Toc44327544)

[运算时的类型转换 5](#_Toc44327545)

[内存分配状况 5](#_Toc44327546)

[常见算法 5](#_Toc44327547)

[进制转换算法 5](#_Toc44327548)

[数字逆序排列算法 6](#_Toc44327549)

[选择排序法(113页) 6](#_Toc44327550)

[冒泡排序法 7](#_Toc44327551)

[分离整型数的各位数字 7](#_Toc44327552)

[求最大公约数，最小公倍数 8](#_Toc44327553)

[生成随机数（73页随机生成加减运算，报告2实验7随机猜数字） 8](#_Toc44327554)

[学生成绩统计 9](#_Toc44327555)

[数据 11](#_Toc44327556)

[printf格式 11](#_Toc44327557)

[scanf格式 11](#_Toc44327558)

[整型数（常量） 11](#_Toc44327559)

[浮点数（常量） 12](#_Toc44327560)

[字符（常量） 12](#_Toc44327561)

[整型变量 13](#_Toc44327562)

[位运算 13](#_Toc44327563)

[整型数的二进制表示 14](#_Toc44327564)

[函数 14](#_Toc44327565)

[函数的原型声明 14](#_Toc44327566)

[函数的定义 14](#_Toc44327567)

[变量的存储类型 15](#_Toc44327568)

[指针 15](#_Toc44327569)

[void指针（无类型指针） 15](#_Toc44327570)

[定义的注意事项 16](#_Toc44327571)

[指针的运算 16](#_Toc44327572)

[多级指针 16](#_Toc44327573)

[指向函数的指针 16](#_Toc44327574)

[字符串 17](#_Toc44327575)

[其他 17](#_Toc44327576)

## ASCII码表

1.范围：一共128个字符，从0-127.

2.先出现ABCDE，后出现abcde.

## 运算优先级

|  |
| --- |
| C:\Users\conch\Documents\Tencent Files\1812071590\Image\Group2\JV\F7\JVF7G[SBO)DR9FMVB{@2{)G.jpg |
| C:\Users\conch\Documents\Tencent Files\1812071590\Image\Group2\JV\F7\JVF7G[SBO)DR9FMVB{@2{)G.jpg |

## 运算时的类型转换

**1.算数表达式的类型转换（包括比大小）**

有符号会自动变为无符号数

**2.赋值运算的类型转换**

右值会自动变为左值

## 内存分配状况

|  |  |  |
| --- | --- | --- |
| 程序代码区 | | |
| 数据区 | 静态存储区 | 外部变量和静态变量 |
| 动态存储区 | 堆区，用于动态内存分配 |
| 堆栈区 | 存放自动变量和函数的形参 |

## 常见算法

### 进制转换算法

#### 将N进制表示为十进制数

#include<stdio.h>

#define N 8

int transN\_10( char radix[] );

int transN\_10( char radix[] )

{

int tens=0;

int i=0;

while(radix[i]!=’\0’)

tens + = tens \* N + ( radix[i++] - ‘0’ );

return tens;

}

#### 将十进制表示为N进制数

#include<stdio.h>

#include<stdlib.h>

#define N 8

char\* trans10\_N( int );

char\* trans10\_N( int tens )

{

char radix[20];

itoa(tens, radix, N);

return radix;

}

### 数字逆序排列算法

#include<stdio.h>

int reverse\_number(int);

int reverse\_number(int a)

{

int rev[20];

int i, j, b, d;

for(i = 0; a ; i ++)

{

rev[i] = a % 10;

a /= 10;

}

for(j = i - 1, b = 0, d = 1; j >= 0; j --)

{

b += rev[j] \* d;

d \*= 10;

}

return b;

}

### 选择排序法(113页)

#include<stdio.h>

#define N 10

void func(int a[],int n)

{

int i,j;

int t;

for(i=0;i<n-1;i++)

{

for(j=i+1;j<n;j++)

{

if(a[i]>a[j])

{

t=a[i];

a[i]=a[j];

a[j]=t;

}

}

}

}

### 冒泡排序法

#include<stdio.h>

#define N 10

void func(int a[],int n)

{

int i,j;

for(i=0;i<n-1;i++)

{

for(j=0;i<n-1-I;j++)

{

if(a[j]>a[j+1])

{

t=a[j];

a[j]=a[j+1];

a[j+1]=t;

}

}

}

}

### 分离整型数的各位数字

方法1：用itoa函数，借助字符串转换数字。

#include<stdio.h>

#include<stdlib.h>

void divide1(int number, int\* array)

{

int i;

char str[20];

itoa(number,str,10);

for(i=0;str[i]!=’NULL’;i++)

{

array[i]=str[i]-‘0’;

}

}

方法2：用计算分离各个数字

### 求最大公约数，最小公倍数

#include<stdio.h>

void main() /\* 辗转相除法求最大公约数 \*/

{

int m, n, a, b, t, c;

printf("Input two integer numbers:\n");

scanf("%d%d", &a, &b);

**m=a; n=b;**

**while(b!=0) /\* 余数不为0，继续相除，直到余数为0 \*/**

**{**

**c=a%b;**

**a=b;**

**b=c;**

**}**

printf("The largest common divisor:%d\n", a);

printf("The least common multiple:%d\n", **m\*n/a**);

}

### 生成随机数（73页随机生成加减运算，报告2实验7随机猜数字）

方法一：

1.用srand(time(NULL))进行播种。

2.用rand()产生随机数。

3.用取余运算来约束随机数的范围。a%b的结果在0~b-1之间。

**4.每生成一个随机数，都要额外重新播种。**

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

#define N 10

#define M 5

int main()

{

int a;

srand(time(NULL));

a=rand();//生成无范围的随机数

srand(time(NULL));

a=rand() % N;//利用取余运算，生成0~N-1的随机数

srand(time(NULL));

a=rand() % N +M;//利用取余运算和加法运算，生成M~N+M-1的随机数

return 0;

}

### 学生成绩统计

#include<stdio.h>

#include<stdlib.h>

#define N 10

struct student

{

char name[20];

char sex;

char number[20];

float score[4];

float average;

};

int main()

{

struct student stu[N];

struct student\* pstu[N+1];

return 0;

}

#### 输入成绩函数

void getch(struct student\*,int);

void getch(struct student stu[],int n)

{

int i,j;

char str[20];

for(i=0;i<n;i++)

{

printf(“name:\n”);

gets(stu[i].name);

printf(“sex:\n”);

gets(str);

stu[i].sex=str[0];

printf(“number:\n”);

gets(stu.number);

for(j=0;j<4;j++)

{

printf(“class %d:\n”,j+1);

gets(str);

stu[i].score[j]=atof(str);

}

}

}

#### 计算个人平均分函数

void aver(struct student[],int);

void aver(struct student stu[],int n)

{

int i,j;

for(i=0;i<n;i++)

stu[i].average=0;

for(i=0;i<n;i++)

{

for(j=0;j<4;j++)

stu[i].average+=stu[i].score[j];

stu[i].average/=4;

}

}

#### 计算单科平均分函数

void avers(struct student[],int);

void avers(struct student stu[],int n)

{

static float aver\_score[4]={0};

int i,j;

for(i=0;i<4;i++)

{

for(j=0;j<n;j++)

aver\_score[i]+=stu[j].score[i];

aver\_score[i]/=n;

}

}

## 数据

### printf格式

1.输出百分号反斜杠等字符，printf(“**%%** \\ \’ \’’ ”);

2.

%u，输出**无符号**十进制数

%o，输出**无符号**八进制数

%x，%X，输出**无符号**十六进制数。x的大小写标明a~f的大小写。

%e，%E，输出指数形式的浮点数，e的大小写对应指数中的大小写。

3.

printf(“%md”,a) 输出**至少**m位大小的整型数。若少，则**左**补空格。若多，则全部输出。

printf(“%-md”,a) 输出**至少**m位大小的整型数。若少，则**右**补空格。若多，则全部输出。

printf(“%m.nf”,a) 输出小数点后n位的小数，同时整个浮点数占m位。若少，则**左**补空格。若多，则整数部分输出。

printf(“%-m.nf”,a) 输出小数点后n位的小数，同时整个浮点数占m位。若少，则**右**补空格。若多，则整数部分输出。

printf(“%m.ns”,a) **输出一共占m列，只输出左端n个字符**，其余不再输出，空余部分左补空格。如果n比m还大，m自动变为n。

printf(“%-m.ns”,a) **输出一共占m列，只输出左端n个字符（不变）**，其余不再输出，空余部分右补空格。如果n比m还大，m自动变为n。

### scanf格式

1.scanf(“%md”,&a) 只输出m列数字，其余的不要了。

2.scanf(“%\*md”) 读入m个数字，但不传给任何变量。

### 整型数（常量）

有符号范围：-32768~32767

无符号范围：0~65535

%x十六进制数 0x12f 0x012f(x后面可多放一个0)

%o**无符号**八进制数。

**后缀l/L u/U**

123456789l长整型数

123u无符号整型数

### 浮点数（常量）

两种表达形式：十进制小数，指数形式。

**小数形式：**

1.必须有小数点

例如0.0 3.14 300.**(小数点后可以不加数字)** 11**(小数点前可以不加数字)**

**指数形式：**

1.可以不加小数点，必须有e/E

2.e后不加正负号则默认正号

3.阶码不能省去

整数部分.小数部分e/E±n(阶码)后缀

例如：

1.23456e+4

1.23456e+04**(阶码可以前置0，不代表八进制**)

.234e+12(可以不要整数部分)

1.e+3(可以不要小数部分)

25e5(可以不要小数点，**可以不要阶码的正负号**)

**后缀l/L**

### 字符（常量）

字符实际是对应ASCII码的整型数

**用多种方法表示字符’A’**

1.’A’(一般字符表示)

2.65(十进制ASCII码)

3.0101(八进制ASCII码)

4.0x41(十六进制ASCII码)

5.’\101’(转义字符八进制)

6.’\x41’(转义字符十六进制)

### 整型变量

范围

有符号数：-32768-32767

无符号数：0-65535

**取余运算，%后面不能是0。除法运算更不能了。**

### 位运算

1.所有位运算符的运算量都必须是整型或字符串型数据。

#### 按位与&

两者同时为1时才为1

#### 按位或|

两者有一个为1就为1

#### 按位异或^

两者不同即为1，不同则为0

#### 二进制左移<<

a<<b 将二进制的a左移b位

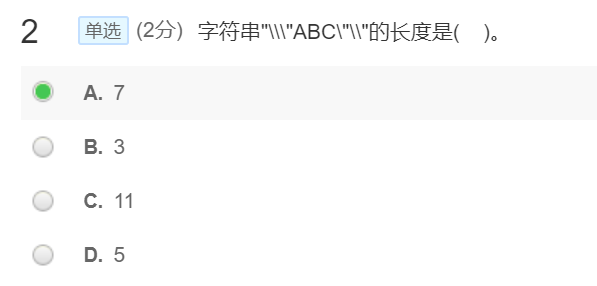
#### 二进制右移>>

a>>b 将二进制的a右移b位

#### 按位取反~

1变成0，0变成1

**数字符串长度时不包括‘\0’**



### 整型数的二进制表示

正数就是其原码

负数用其绝对值的补码形式表示。

补码：原码按位取反，然后加1.

## 函数

### 函数的原型声明

1.函数声明可以在函数内部，也可以在函数外部。

2.函数原型声明的格式：函数数据类型 函数名 （参数类型…） **不包括存储类型！**

### 函数的定义

1.函数的定义由函数头和函数体两部分组成。

<存储类型><数据类型>函数名(<形式参数及说明>)

{

说明语句；

执行语句；

}

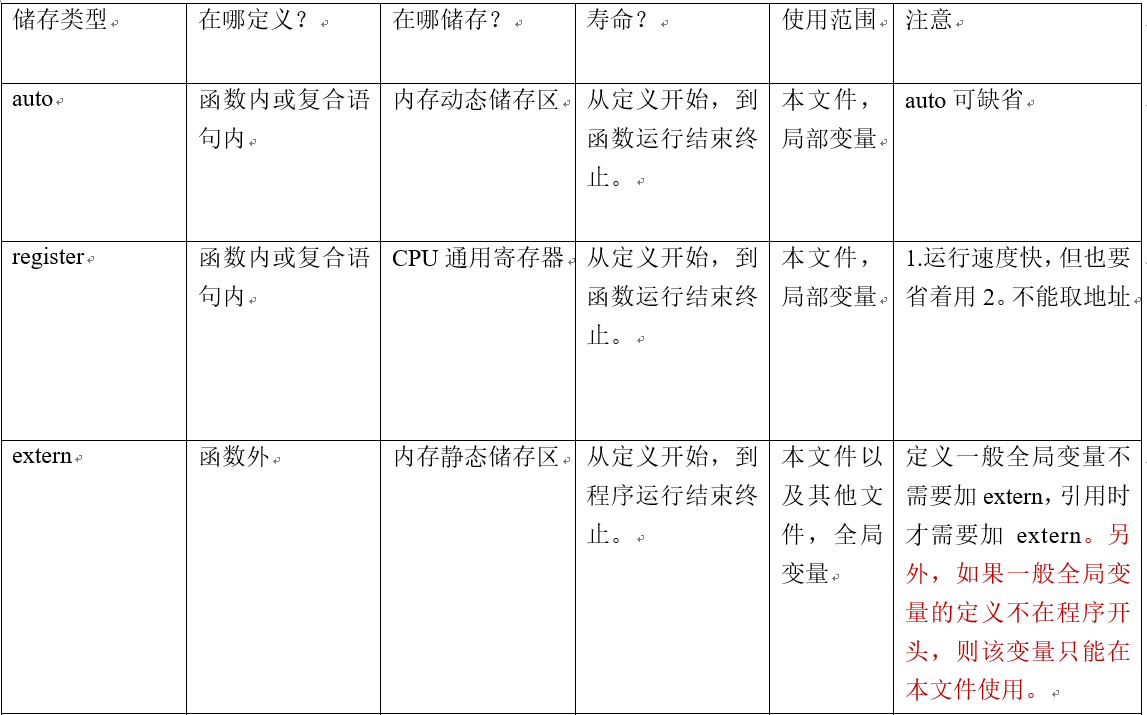
2.存储类型只有extern和static两种。extern可以被任何源文件调用。缺省时默认为extern。

3.当函数返回值为int型时，数据类型可缺省。不需要返回值时，为void。

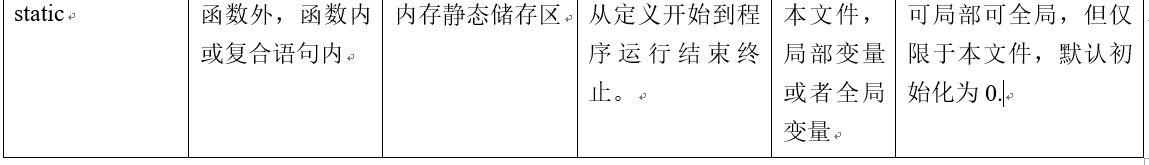
4.不允许在函数体内定义另一个函数。（包括main函数体内）

5.形如 int printf(char\* format,…)意为参数数目可变。

### 变量的存储类型



初始化默认为0



## 指针

### void指针（无类型指针）

如：

int a=1;

int\* pint;

void\* pvoid;

pint=&a;

pvoid=pint;//无类型指针可以任意指向其他数据

pint=(int\*)pvoid;//无类型指针中的数据必须经过类型转换才能赋给有类型的指针

### 定义的注意事项

char\* a,b; 则a是指针，b是一般字符变量。

char a,\* b; 则a是一般字符变量，b是指针。

### 指针的运算

1.指针与指针的加法无意义，指针与指针的减法得到数据的距离。&a[1]-&a[5]=-4

2.\*和自增自减的运算优先级相同，结合律从右至左。

3.指针与整数的加减，移动的都是对应的数据大小的倍数。

4.关系运算 &a[1]<&a[8]

### 多级指针

1.多级指针，c语言不能将int\*\*类型的值分配到int\*的实体。

2.**二级指针可以直接等于指针数组int\* []**，但不能直接等于数组指针int(\*)[]。

3.二级指针也可以用pp[i][j]来表示其中的数据。

4.int a[];**二级指针p不能等于&a**

### 指向函数的指针

**1.定义**

形如：

float func(int,int);

main()

{

float (\*pfun)(int,int)=func;

}

<存储类型> <函数返回值的数据类型> (\*指针名)(参数1,参数2…)=<函数名>

**2.使用**

以下形式均可：

1.(\*pfun)(1,2); \*是取内容操作

2.pfun(1,2);

3.func(1,2);

以上效果等价。函数名实际就是函数定义的地址，所以pfun(1,2)跟func(1,2)实际是一回事。

3. 对函数指针使用++等是无意义的.

## 字符串

1.char str[20]=”hello”;是合法的。

## 其他

1.后置的自增在逗号或分号后开始自增

例如：a=1,b=1; (a++,a+b)的结果是3

2.

a=a++;

这是一个未定义行为。结果是a不会自增。

3.switch中default情况

4.int x=1,y=x;是合法的

int x=y=1;不合法

5.putchar不自动输入一个回车。

6.

有符号数-1变为无符号数2^16-1

有符号数-19变为无符号数2^16-19

7.

char i;

for(i=0;i<n;i++)

{…}

这是正确的。**不过要注意n不能超过256**

8.

**注意自定义函数是传递了指针还是数字。如果只是传进去了数字，不可能对main函数造成影响。**

9.所有的递归函数都是可以写成循环的.

10. strstr(s1,s2)函数找到s1中第一次出现s2的地方，如果s2是空指针，那么就返回s1，**如果没有出现s2，则返回NULL**

11.二维数组也是“一维的”！**a[2][3],那么a[0][2]地址后面紧跟着的就是a[1][0]!**

12.sizeof运算

1)sizeof(int)，sizeof(a)//a为某变量,则得到的结果是这类变量的字节数。

char，unsigned char是一字节，其他字节数不同编译器不同。

2)sizeof(数组)，则得到数组的总长度，如果是字符数组，注意是否有额外的’\0’

