

同济大学 C 语言辅导班笔记

考研加油站收集整理 <http://www.kaoyan.com>

教师: 陈雨申

C 语言要点与难点

- 数据类型与转换
- 函数和程序结构
- 变量的存储类型
- 数组越界
- 指针

一. 基本数据类型:

char	1 字节	-128~127
int	2 字节	-32768~32767
long	4 字节	
float	4 字节	
double	8 字节	

二. 转义符:

'A' 'a' (区分大小写) 'n' 't' '\b' '\r' '\f' '\n'

例 char *p = "it1r1110181w";

for (; *p != '\0'; p++) 循环执行次数为 7 次。

三. 运算符

只识别前 3 个字符

1. 转换

double ← float	
↑	
long	即使相同类型运算, 也先
↑	转换为左边的 结果转成
unsigned	
↑	
int ← char, short	右边的

2. 自增 自减运算符

++ i

例: 略

3. 运算符优先级:

括号 / 整体 > 双目运算符 > 关系 > 与 > 或 > 条件运算符 >

赋值运算符 > 逗号

左结合 (除了以下提到的)

右结合 (幂, 条件, 赋值三个)

4. 逻辑运算 && ||

且左侧的为0, 右侧就不计算, 尽管右侧的优先级很高.

例: $y=1, x=0, z = x \&\& (y=99)$ 依旧 $y=1$

5. 移位运算:

例: 将 int 类型变量 w 的高字节放入结果 int 类型变量 m 中的高字节, 将 w 的低字节放入 m 的低字节

$m = (w \ll 8 | high) \gg 8$

6. 条件运算:

用一个表达式 x, y, z 的最小值 (2002年)

$min = (x < y ? x : y) < z ? (x < y ? x : y) : z$

四. 语句

1. 输入和输出

printf() scanf() 格式有 %d, %o, %x, %u,

%c %s %f

getchar() putchar() 头文件须 include <stdio.h>

2. 重要语句:

控制语句: if else, switch 等

复合语句:

break (停止循环) continue (重执循环)

例: 既是完全平方数 又有两位相同的三位整数

main()

```
{ int n, k, a, b, c;
```

```
for (k=1; k<=n; k++)
```

```
{ n=k*k;
```

```
if (n < 100)
```

```
continue;
```

```
if (n > 999)
```

```
break;
```

```
a = n / 100;
```

```
b = n / 10 % 10 或 n % 100 / 10;
```

```
c = n % 10;
```

```
if (a == b || a == c || b == c)
```

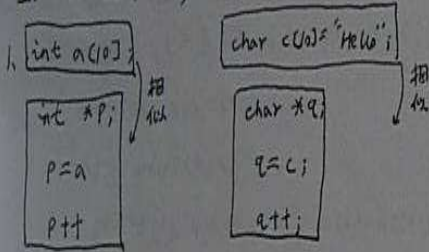
```
printf("N = %d")
```

```
}
```

```

flag (int x, int y, int z)
{ return ((x-y)*(y-z)*(x-z)); }
}
    
```

五. 数组与指针初步



char c[]="Hello" 自动加一个 '\0'

2. 字符串:

字符串函数: gets puts strlen strcpy strcat strcmp

gets与scanf(%s)的区别: gets可含空格, scanf不可
带空格

例: 编程将字符串根据空格分离单词

```

main()
{ char *s1="asd fdh k(n tu d";
  char *s2[20], *t, *temp;
  int i=0, j;
  for(i *s1!='\0'; s1++)
    { for (temp=t=s1; *t!=' '; t++)
    }
    }
    
```

```

*t='0';
s2[i++] = temp;
s1 = s1 + strlen(temp);
for(j=0; j<(i-1); j++)
printf("s2[%d]=%s\n", j, s2[j])
}
}
    
```

3. 数组排序 (选择, 冒泡)

六. 函数与程序结构:

98年

```

void f(int i, char c)
{ if(i!=0)
  { if(i>1) f(i-2, c);
    else f(i-1, c);
    printf("%c", c);
  }
}
main()
{ f(6, 'c'); } // 输出结果 egi
    
```


19年一

```
#include <stdio.h>

int orderi (int a[], int i)
{ return a[i] < a[i+1]; }

main ()
{ int a[] = {5, 8, 7, 4, 20, 10, 15, 18}, i;
  for (i=0; i<7; i++)
    if (!orderi (a, i))
      printf ("%4d%4d\n", a[i], a[i+1]);
}
```

输出结果: 4 20 8 15 7 10 5 18



2000年三.1.

- (1) *old != '10'
- (2) *s1 = *s2
- (3) *c0 != *c1
- (4) *s2++ (5) old = *old + strlen(sub);

变量作用域:

局部变量: 调用时开辟空间 函数内部

全局变量: 不释放 函数外部

static 在函数结束时不释放 可以保持上次访问后的值

能初始化一次

例: 编程 void produce (char *s, int n) 函数, 用于

随机产生 n 个不相同的小写字母, 存放在字符串中

随机函数为 rand(), 产生 0 ~ 32767 间整数

#include "math.h"

```
void produce (char *s, int n)
{ int i, j; char c;
  for (i=0; i<n; i++)
    { re: c=(int) (rand() / 32767.0 * 25 + 97);
      for (j=0; j<i; j++)
        if (c==s[j]) goto re;
      s[i]=c;
    }
  s[i]='\0';
}
```

注意: ① 全局变量 静态变量都不在栈中分配, 在数据区分配, 若不人为初始化, 自动初始为 0. 局部变量不

自动初始, 另自行得到地址的原值

② 全局变量加 static 表示对自己的文件有效, 别人不能调用

局部变量加 static 表示保持该变量的值

例: $i=1$ $f=(i++)*(i++)$

$\Rightarrow f=1, i=3$

② 函数调用: 参数传递是右到左的

$i=7$ void $f(i, i++)$

传入的参数分别是 8 和 7.

七. 编译预处理

#define #include

例 02. -3

#define DIV(x) x/x

main()

{ int a=0, k=3, m=1;

a = DIV(k+m) / DIV(k+m);

printf("%d, %d\n", DIV(k+m) / DIV(k+m), a);

}

结果为 4, 2.

八. 指针参考书 255页总结

九. 命令行参数

例 2002. 4

a[0] 0xEE00

*a+2 0xEE04

a+2 0xEE0C

*(&a[0]+1) = *(a+1) = 8

*(&a[0]+2) = 4

例: int a[3] = {1, 2, 3, 4, 5, 6, 7, 8, 9};

int *p = {a[0], a[1], a[2]};

main()

{ for (i=0; i<3; i++)

printf("%d, %d\n", *p[i], p[i]);

输出结果 1, 1, 4, 2, 7, 3

例: 2001. -3. 输出结果 20.

十. 结构 (联合不考)

结构主要用于链表 注意定义方式与语法规则

2002 年三.

① $t_1 \&\& t_2 \&\& t_3$

② $t_1 \rightarrow value < t_2 \rightarrow value$

③ $t_2 \rightarrow value < t_1 \rightarrow value$

④ $t_3 \rightarrow value < t_1 \rightarrow value$

⑤ $t_1 = t_1 \rightarrow next$

150 2000, 三, 2

(1) $p = q \rightarrow next^*$;

(2) $q_1 \rightarrow next = q_1$;

(3) $q_1 = p$;

(4) $h \rightarrow next = q_1$;

十一文件 (重点)

常用函数用法掌握, 多做题, 详见书上

```
#include "stdio.h"
```

```
void main()
```

```
{ char fBuf[40], gBuf[40];
```

```
FILE *f, *g;
```

```
f = fopen("e:\\TempFile", "r");
```

```
g = fopen("e:\\TempFile", "r");
```

```
for (int i=0; i<3; i++)
```

```
{ fread(fBuf, 1, 3, f);
```

```
fBuf[3] = '\0'; // ABC
```

```
fread(gBuf, 1, 5, g);
```

```
gBuf[5] = '\0'; // ABCDE
```

```
cout << fBuf << endl; // DEF
```

```
cout << gBuf << endl; // FGHIJ
```

```
}
```

```
fclose(f); fclose(g); // KLMNO
```

注意: 输入输出改用: 如从键盘改为从文件输入

```
mycopy < inputFile > outputFile
```

程序名 输入文件 输出文件

总结及其他问题:

1. Σ (亦和), 函数不再要求了
2. 没有 union, 枚举 typedef 只要能看懂即可
3. 在结构中通过结构指针引用结构要求
4. 链表不在 C 的范围内 (数据结构的)
5. 运算符的优先级, 结合方法, 短路运算符
6. 变量的存储类型
7. C 传递参数的特点 (传值, 传址)
8. 二维数组的数组名是二级地址
9. 指针
10. 文件函数及参数