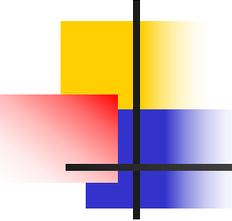


复习课

周水庚

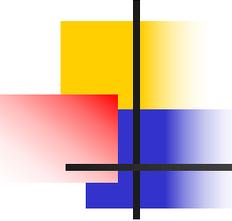
计算机学院

2024年10月31日



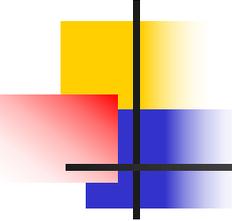
期中考试时间/地点

- 时间: 11月7日 (周四) 下午13: 30-15: 30
- 地点: **????**
- 方式: 闭卷考试
- 题型: 读程序和写程序
 - 选择题
 - 改错题 (改语句)
 - 填空题 (计算填空、代码填空)
 - 编程题



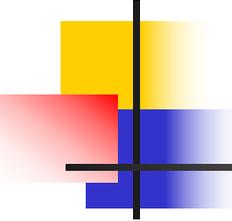
C语言关键词与函数

- C语言中有一些保留使用的**关键词**
- 预处理命令行中的词不是关键词，但也是保留使用的
- 除了main函数外，大部分编程时调用的是环境提供的函数，不是C语言所固有的



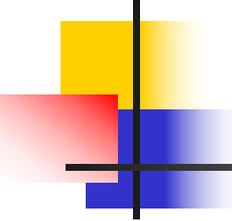
字符与整数

- 字符变量可以赋整数值
 - `char c=9; int i=c,j; j=c-0; printf("%d,%d", i,j);`
 - `char c='9';int i=c, j; j=c-'0'; printf("%d,%d", i,j);`
- 字符表示: 转义字符
 - `\ddd` /* 8进制表示
 - `\xdd` /* 16进制表示
 - `char c='\12'`
 - `char s[]="AB\6CD\\\0EF"; printf("%d,%s",strlen(s), s)`



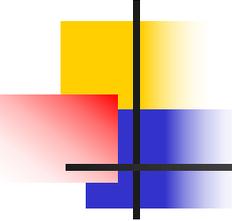
整数的表示

- `int i1=100, j1=-100, i2=0xaaaa, j2=-0xaaaa`
- `printf("%0x, %0x", i1, j1);`
 - 64, fffffff9c
- `printf("%0x, %0x", i2, j2);`
 - aaaa, ffff5556



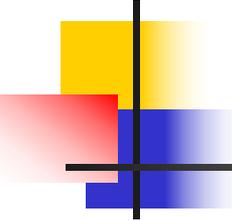
优先级

- 初等运算符: `()`、`[]`、`..`、`->`
- 单目运算符
 - 地址/指针运算: `*/&`
 - 逻辑/位运算: `!/~`
 - 算术运算: `+/++`、`-/--`
 - `sizeof` 变量、`sizeof(类型名)`
 - (强制类型)
- 双目运算符
 - 算术运算: `*`、`/`、`%`、`+`、`-`
 - 移位运算: `<<`、`>>`
 - 关系运算: `<`、`<=`、`>`、`>=`、`==`、`!=`
 - 位运算: `&`、`^`、`|`
 - 逻辑运算: `&&`、`||`
- 条件运算: `?:`
- 复合/赋值运算: `=`、`Θ=`
- 逗号运算: `,`



基本输入输出

- 掌握scanf()和printf()基本使用方法
 - `char s[10], x[10]; int a[20]`
 - `scanf(“%s”, &s[0]);`
 - `scanf(“%s”, x);`
 - `scanf(“%c”, &x[0]);`
 - `scanf(“%d”, &a[1]);`

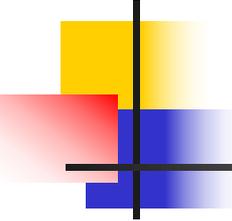


控制条件与循环终止

- `int x; if (x==0)x=1 else x=0; printf(“%d”,x);`
- `int x; if (x=0)x=1 else x=2; printf(“%d”,x);`

- `int x,y; for(x=0,y=10;x=0;x++,y--)x++;`
- `int x,y; for(x=0,y=3;x=y;x++,y--)x++;`

- `for()`循环语句中的`break`与`continue`
 - `for(i=0;i<10;i++) if(i==5)break;`
 - `for(i=0;i>=0;i++) if(i!=5)continue;`



一维数组初始化

- `int e[5] = {0, 1, 2};`
- `int g[] = {5, 6, 7, 8, 9};`
- `int c[5] = {0, 1, 2, 3, 4, 5};`

二维数组初始化

- 按行给二维数组的**全部**元素赋初值

```
int a1[2][3]={ {1, 2, 3},{4, 5, 6}};
```

- 按元素存储顺序给数组元素赋初值

```
int a2[2][3] = {1, 2, 3, 4, 5, 6};
```

- 按行给数组的**部分**元素赋初值

```
int a3[2][3]={ {1, 2}, {0, 5}}; /*其余均为 0*/
```

- 按元素存储顺序给前面**部分**元素赋初值

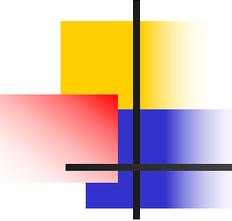
```
int a4[2][3] = {1, 2, 3, 4}; /*其余均为 0*/
```

- 按元素存储顺序，给数组部分或全部元素赋初值，并且不指定第一维的元素个数

```
int a5[][3] = {1, 2, 3, 4, 5};/* 两行 */
```

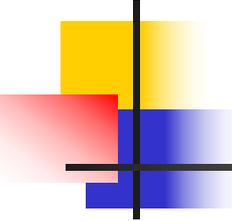
- 用按行赋初值方法，对各行的部分或全部元素赋初值，并省略第一维的元素个数

```
int a6[][3] = {{0, 2}, {}};/* 两行 */
```



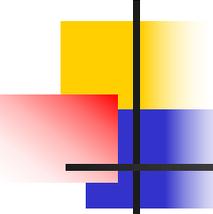
字符数组 (1)

- 字符数组初始化
 - `char s[]={ 'F', 'u', 'd', 'a', '\n' };`
 - `char s[]={ "Fudan" };`
 - `char s[]="Fudan";`
 - `char s1[5]="Fudan";`



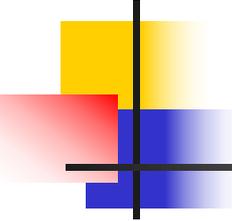
字符数组 (2)

- 字符数组存储字符串，可以整体输出
 - `char s[]="Fudan";`
 - `printf("%s",s);`
- 而仅仅存储字符的字符数组不能整体输出
- 字符数组可以直接输入字符串
 - `char s[10];`
 - `scanf("%s",s);` /* 输入的字符串不能超过9个有效字符



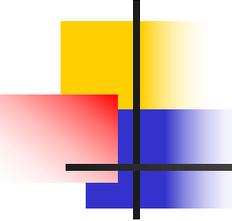
字符数组 (3)

- `char str1[10] = "fudan";`
- `char str2[20] = "fudan\0university";`
- `char str3[20] = "fudan\01university";`
- `char str4[20] = "fudan\012university";`
- `char str5[20] = "fudan\0123university";`
- `char str6[20] = "fudan\x123university";`
- `int i1, j1, i2, j2, i3, j3, i4, j4, i5, j5, i6, j6;`
- `i1=sizeof(str1) ; i2=sizeof(str2) ; i3=sizeof(str3) ; i2=sizeof(str4) ;
i2=sizeof(str5) ; i2=sizeof(str6);`
- `j1=strlen(str1) ; j2=strlen(str2) ; j3=strlen(str3) ; j4=strlen(str4) ;
j5=strlen(str5) ; j6=strlen(str6);`
- `printf("%d,%d , %d,%d , %d,%d , %d,%d , %d,%d , %d,%d", i1,j1 ,
i2,j2 , i3,j3 , i4,j4 , i5,j5 , i6,j6);`



函数与函数调用

- 函数不可以嵌套定义，但可以递归调用
- 调用函数时，只能把实参的值传递给形参，形参的值不能传递给实参
- 例子：`int i, f(int x, int y);`
- 调用
 - `i=f(3, 4);`
 - `i=f(2+1, 2*2);`
 - `i=f((4, 3), (3,4));`



文本文件与二进制文件

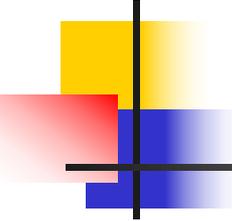
- 假设在某一计算机中，**int**用4个bytes表示
- **int i=100, j=-100**
 - 那么，**i**和**j**在二进制文件中，只占4bytes
 - 而在文本文件中，则分别占3bytes, 4bytes
- 假设在某一计算机中，**float**用4个bytes表示
- **float f1=100.2345, f2=-1.115**
 - 那么，**f1**和**f2**在二进制文件中，只占4bytes
 - 而在文本文件中，则分别占8bytes, 6bytes

数组元素排序

■ 冒泡排序

- 每一轮要相邻数据两两比较并交换位置。其实可以不用每次交换位置，每一轮只要把最大或最小的挑出来即可
- 假设有`int x[M]`，`M`是一个正整数，要对其中的元素从小到大排序

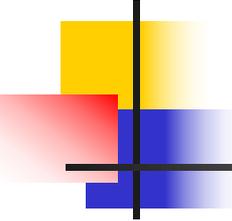
```
int i, j, min, temp;
for(i=0;i<M-1;i++){
    for(j=i+1,min=i;j<M;j++)
        if(x[j]<x[i])min=j;
    if(i!=min){
        temp=x[i]; x[i]=x[min]; x[min]=temp;
    }
}
```



倒序输出一个正整数

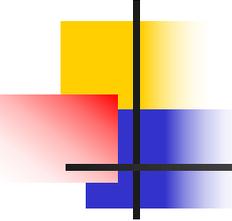
```
int x, i, j
```

```
for (j=0;x>0; x/=10){  
    i=x%10;  
    j++;  
    printf(“%c”, ‘0’+i);  
}
```



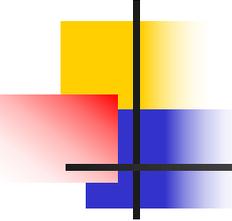
顺序输出一个正整数

```
void show-int(int n)
{ char c;
  c = n % 10 + '0';
  if (n >= 10) show-int(n/10);
  printf("%c", c);
}
```



课本中的几个算法

- 冒泡排序，有三个版本
 - 仔细看课件
- 二分搜索算法
 - 课件中也有
- 求两个整数的最大公因数



其它

- 一些函数

- printf(),scanf()
- fopen(), fclose()
- fgetc(),getchar(),putchar()
- gets(),puts()
- fprintf(),fscanf()
- malloc(),calloc()