



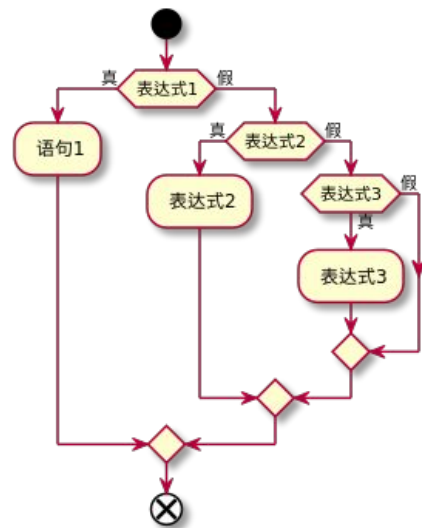
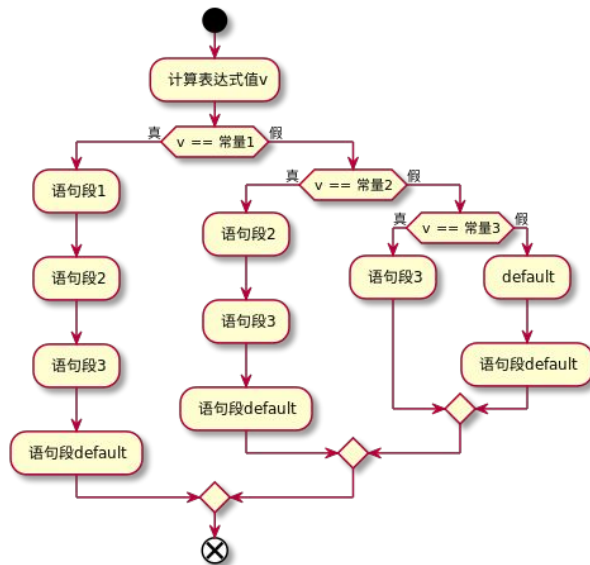
# C语言程序设计基础

林川

# 回顾分支结构



- if
- if/else
- if/else if/.../else if/else
- switch





## 第四章 循环结构

1. while循环结构
2. do-while循环结构
3. break语句和continue语句
4. 多重/嵌套循环

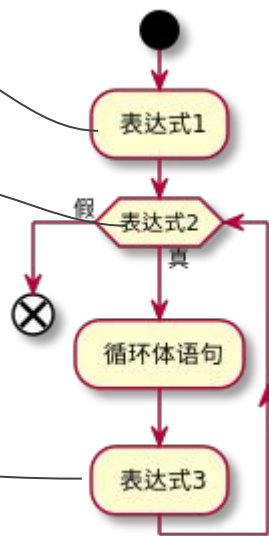
# for语句回顾



for( 表达式1; 表达式2; 表达式3 )

循环体语句

```
for( i=1; i<n; i++ )  
{  
    item = 1.0/i;  
    sum = sum + item;  
}
```





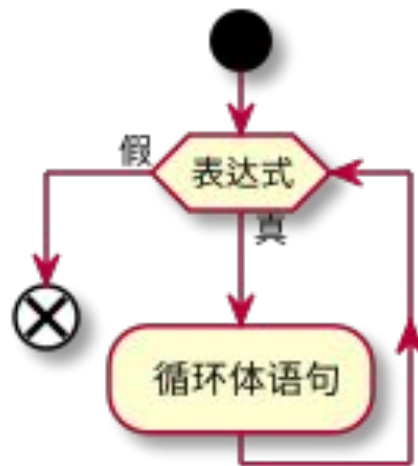
# while循环语句

while( 表达式 )

循环体语句

当表达式成立的时候

执行循环体语句



# for与while对比



for( 表达式1; 表达式2; 表达式3 )

循环体语句

while( 表达式 )

循环体语句

for改while

```
表达式1;  
while( 表达式2 )  
{  
    循环体语句  
    表达式3;  
}
```



# while语句应用( I )

格雷戈里公式求PI的值, 要求最后一项的绝对值  $< 10^{-4}$

while( 表达式 )

循环体语句

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

当表达式成立的时候

执行循环体语句

```
double sum, item;  
int denominator, flag;
```

绝对值: fabs() -> <math.h>



# 用格雷戈里公式求 $\pi$

```
sum = 0; denominator = 1; flag = 1;  
item = 1.0;
```

```
while( fabs(item) > 0.0001 )  
{  
    item = flag * 1.0/denominator;  
    sum = sum + item;  
    /* 准备下一项 */  
    flag = -flag;  
    denominator = denominator + 2;  
}
```

```
printf( "pi = %.4f\n", sum * 4 );
```

**item的初值有多重要？**





## while语句应用( II )

输入一批学生成绩, 以负数作为结束标志, 计算平均成绩, 统计不及格人数。

```
int num, failed;  
double grade, sum;
```

```
num = failed = 0;  
sum = 0;
```

```
printf("Enter grades:");
```

成绩个数未知!  
适合while循环

# 统计学生成绩



```
scanf("%lf", &grade);  
while( grade >=0 )  
{  
    sum = sum + grade;  
    num ++;  
    if( grade < 60 )  
        failed ++;  
    /* 准备下个数据 */  
    scanf("%lf", &grade);  
}
```

```
if( num != 0 )  
    printf("Grade average is %f\n", sum/num);  
printf("Number of failures is %d\n", failed);
```

# do-while语句



```
do {  
    循环体语句  
} while( 表达式 )
```

执行**循环体语句**,  
直到**表达式**为假

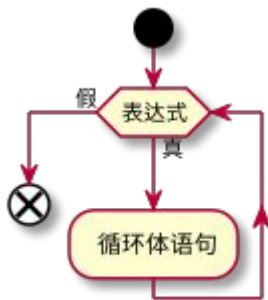




# While与do-while

while( 表达式 )

循环体语句



do {

循环体语句

} while( 表达式 )





# do-while语句应用

计算一个整数的位数。例如:0的位数为1, -99的位数为2。

```
int number, count;
```

```
printf("Enter a number: ");
```

```
scanf("%d", &number);
```

123 3

4567 4

-100 3

如何计算一个  
整数的位数？

不断用10去除  
它，直到零为  
止

# 计算整数的位数



```
count = 0;
do {
    number = number / 10;
    count ++;
} while ( number!=0 );

printf("it contains %d digitals.\n", count);
```

改为下面的while循环如何？

```
count = 1;
while ( abs(number)>9 )
{
    count ++;
    number = number / 10;
}
```

# 循环体中break和continue语句



素数的判定问题:除了1和自身,不能被别的数整除

```
int i, m;

scanf("%d", &m);

for( i = 2; i <= m/2; i++ )
    if( m % i == 0 )
        break;
```

```
if( i > m/2 && m != 1 )
    printf("Yes\n");
else
    printf("No!\n");
```



# 作业

用最少的计算步骤完成素数判断

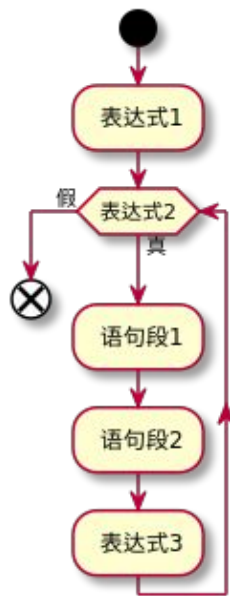
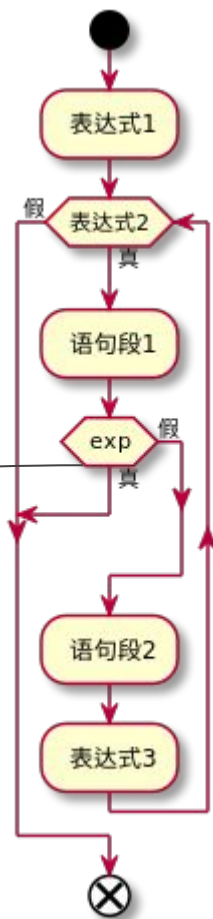
以取模运算最少作为标准





# for循环中的break语句

```
for( 表达式1; 表达式2; 表达式3 )  
{  
    语句段1  
  
    if( exp )  
        break;  
  
    语句段2  
}
```

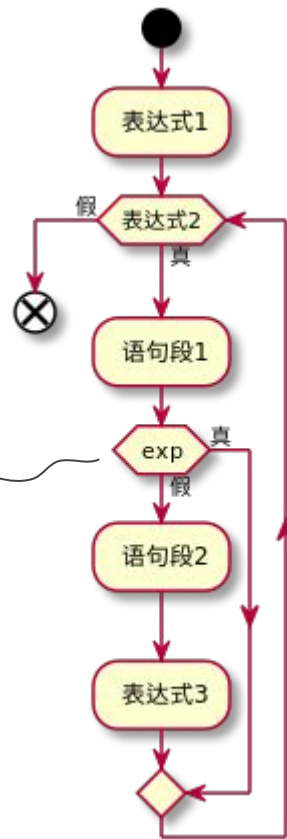


without break

# for循环中的continue语句



```
for( 表达式1; 表达式2; 表达式3 )  
{  
    语句段1  
    if( exp )  
        continue;  
    语句段2  
}
```



# while循环中的break语句



```
while( 表达式 )  
{  
    语句段1  
  
    if( exp )  
        break;  
  
    语句段2  
}
```

# while循环中的continue语句



```
while( 表达式 )  
{  
    语句段1  
  
    if( exp )  
        continue;  
  
    语句段2  
}
```



# 循环体中break和continue语句

- 适用于所有的循环结构
  - for, while, do-while
- 用于灵活控制循环结构的执行流程



# 多重嵌套循环

- for, while, do-while 能够相互嵌套
  - 构成多重循环
- 处理多维数据
- 处理复杂过程



# 多重嵌套循环

假设有一幅宽度为`width`、高度为`height`的图像，使用`for-for`嵌套语句，对它的每一个像素进行处理。（例如亮度值放大）

```
for( x = 0; x < width; x ++ )    每个像素用(x,y)坐标表示
{
    for( y = 0; y < height; y ++ )
    {
        /* 处理坐标为(x,y)的图像像素 */
        ... .. /* 具体代码略 */
    }
}
```

循环体被执行了多少次？  
`width * height`



# 循环结构的设计

- 循环程序的实现要点：
  - 归纳出哪些操作需要反复执行？ 循环体
  - 这些操作在什么情况下重复执行？ 循环条件
- 选用合适的循环语句  
for while do-while
- 循环具体实现时考虑（循环条件）：
  - 事先给定循环次数，首选for
  - 通过其他条件控制循环，考虑while或do-while
    - 至少执行1次，用do-while
    - 否则，用while



# n个成绩的最高分



```
int n, i, grade, max;

printf("Enter n: ");
scanf ("%d", &n);
printf("Enter %d marks: ", n);

scanf ("%d", &mark);
max = mark;

/* 剩余成绩循环处理 */
/* 次数未知, 适合while语句 */
```

# n个成绩的最高分



```
for( i = 1; i < n; i++ )
{
    scanf ("%d", &mark);

    if( max < mark )
        max = mark;
}

printf("Max = %d\n", max);
```

# 如果一批成绩以负数结束 求最高分



```
int grade, max;
```

```
printf("Enter marks: ");
```

```
scanf ("%d", &mark);
```

```
max = mark;
```

```
/* 剩余成绩循环处理 */
```

```
/* 次数未知, 适合while语句 */
```

# 如果一批成绩以**负数结束** 求最高分



```
while( mark>=0 )
{
    if( max < mark )
        max = mark;

    scanf ("%d", &mark);
}

printf("Max = %d\n", max);
```

# 将整数按照数字逆序输出



例如:  $x = 12345$  的逆序为 54321

如何得到呢?

从低位开始逐个计算

$$5 = x \% 10, x \square x/10 = 1234$$

$$4 = x \% 10, x \square x/10 = 123$$

$$3 = x \% 10, x \square x/10 = 12$$

$$2 = x \% 10, x \square x/10 = 1$$

$$1 = x \% 10, x \square x/10 = 0 \text{ [结束]}$$

# 将整数按照数字逆序输出



```
int x;
```

```
printf("Enter x: ");
```

```
scanf ("%d", &x);
```

```
/* 对每一位数字进行循环处理 */
```

```
/* 次数未知, 适合while语句 */
```

# 将整数按照数字逆序输出



```
while( x != 0 )  
{  
    printf( "%d", x%10 );  
    x = x/10;  
}
```

# 将整数按照数字逆序输出



```
/* 用 do-while 实现 */  
do {  
    printf( "%d", x%10 );  
    x = x/10;  
} while( x != 0 )
```

用 do-while 实现更好，对0也work!!



# 求100以内的素数

## 每行输出10个



- 需要考察的整数范围

2,3,4,...,100

- 素数: 没有真因子。

对于整数  $m$ , 真因子的范围是:

- $1 < \text{真因子} < m$
- $1 < \text{真因子} \leq m/2$
- 如非素数, 必有真因子满足:  $1 < \text{真因子} \leq m^{1/2}$

- 每行输出10个

- 当个数是10的倍数时, 输出换行符  
`printf("\n");`



```
#include<stdio.h>
#include<math.h>

int count, m, n, i;

count = 0 ;
```

```
for( m = 2; m<100; m++ )
{
    n = sqrt(m);
    for( i=2; i<=n; i++ )
        if( m%i==0 )
            break;
    if( i<=n )
        continue;
    printf("%6d", m);
    count ++;
    if( count%10==0 )
        printf("\n");
}
```

# 计算并输出:斐波那契数列前10项



- 1 1 2 3 5 8 13 21 ...
- 从第3项起, 等于前2项之和

```
int x1, x2, x, i;  
x1 = x2 = 1;  
printf("%6d%6d",x1,x2);
```

# 计算并输出:斐波那契数列前10项



```
for( i=3; i<=10; i++ )
{
    x = x1 + x2;
    printf("%6d",x);
    x1 = x2;
    x2 = x;
}
printf("\n");
```



# 穷举算法(搬砖)

- 男人:3块/人
- 女人:2块/人
- 小孩:1块/2人
- 问:45人搬45块砖, 有多少种搬法?

```
int men, women, child;
```



# 穷举算法(搬砖)

```
for( men=0; men<=45; men++ )  
for( women=0; women<=45; women++ )  
for( child=0; child<=45; child++ )  
    if( men+women+child==45 &&  
        men*3+women*2+child*0.5==45 )  
    {  
        printf("men=%d, women=%d, child=%d\n",  
            men, women, child);  
    }
```

合计执行多少  
次循环操作？  
 $45*45*45$

# 改进[1]



合计执行多少  
次循环操作？

45\*45

```
for( men=0; men<=45; men++ )  
for( women=0; women<=45; women++ )  
{  
    child = 45-men-women;  
    if( child>=0 &&  
        men*3+women*2+child*0.5==45 )  
    {  
        printf("men=%d, women=%d, child=%d\n",  
            men, women, child);  
    }  
}
```

## 改进[2]

编写循环程序时  
注意代码的执行效率



```
for( men=0; men<=15; men++ )  
for( women=0; women<=22; women++ )  
{  
    child = 45-men-women;  
    if( men*3+women*2+child*0.5==45 )  
    {  
        printf("men=%d, women=%d, child=%d\n",  
               men, women, child);  
    }  
}
```

合计执行多少  
次循环操作？

15\*22



# 要点



- for
- while
- do-while
- break
- continue
- 多重循环
  - 循环次数的计算
  - 执行效率

# 作业

