# 流程图

**常见的算法结构：顺序结构、分支结构、循环结构**

# 顺序结构

**C语言中的顺序结构：执行过程就是从上到下依次执行语句。**

**常见的C语言的语句：**

**表达式语句 label语句 复合语句 条件语句、switch语句**

**break语句、continue语句 return语句 goto语句**

**空语句 循环语句**

## 表达式语句

**表达式语句由表达式加分号构成。**

**常见的表达式语句有：赋值语句、自增语句、自减语句、函数调用语句。**

**例：观察下面的表达式语句。**

**（1）赋值语句：x = y + 3;**

**（2）自增语句：++i;**

**（3）自减语句：--i;**

**（4）函数调用语句：printf("请输入半径:\n");**

## 复合语句

复合语句由大括号中0个或多个**声明**和**语句列表**共同构成.

复合语句是**一条**语句

{

Int x=3;//声明语句

++x;//自增表达式语句

Printf(“pause”);//函数调用语句

}

## return语句

**return语句的基本形式为：return; 或 return 表达式;**

**return语句的作用是结束当前函数**

**#include** **<stdio.h>**

**int** **main(void)**

**{**

**…**

**printf("%d\n",** **x);**

**printf("%d\n",** **y);**

**return** **0;**

**}**

# 分支结构

C语言中的分支结构：执行过程就是**满足条件执行A分支顺序语句，否则执行B分支顺序语句**

## If语句

C语言中的多路分支结构：执行过程就是**根据判断条件的多个值**来判断走哪一条分支，一般为多分支。

**分支语句有两种基本形式：**

**if语句的一般形式：**

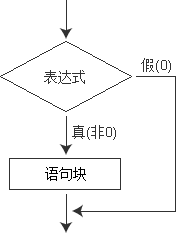
**if(表达式)**

**{**

**语句;**

**}**

**其语义是：如果表达式的值为真，则执行其后的语句，否则不执行该语句。其过程可表示为下图。**



**if – else 语句的一般形式：**

**if(表达式)**

**{**

**语句1;**

**}**

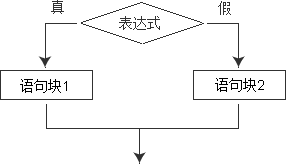
**else**

**{**

**语句2;**

**}**

**如果表达式的值为真，则执行语句1，否则执行语句2 。其执行过程可表示为下图。**



## Switch语句

switch语句被称为**多路分支语句**

**一般形式：**

**switch(表达式)**

**{**

**case 整型字面值1:**

**语句1(集合);**

**break;**

**case 整型字面值2:**

**语句2(集合);**

**break;**

**case 整型字面值n:**

**语句n(集合);**

**default:**

**语句n+1(集合);**

**}**

**注意：**

1. **case的标号不能够超过1023个**
2. **表达式的值为整型或者是字符型。**
3. **switch语句的执行方法如下：**

**若表达式的值等于switch 语句中某个case 标号中的整型字面值(每个case 标号中的字面值不能相同)则程序控制转移到该case 标号表示的点，从此点开始执行,即执行其后的语句，然后不再进行判断，继续执行后面所有case后的语句。直到遇见 break跳出switch。**

**若表达式的值不等于任何 case 标号中的字面值，则程序控制转移到default 标号（如果有的话）表示的点，从此点开始执行。**

**若表达式的值不等于任何 case 标号中的字面值，又没有 default 标号，则不执行switch 语句体中的语句。**

1. **在case后，允许有多个语句，可以不用{}括起来。**
2. **各case和default子句的先后顺序可以变动，而不会影响程序执行结果。**
3. **default子句可以省略不用。**

# 循环结构

## for循环语句

**for循环语句的基本形式为：**

**for(初始表达式1;条件表达式2;循环表达式3)**

**{**

**语句;** **//** **被称为for** **语句体**

**}**

**执行过程：**

1)执行表达式1

2)判断是否满足表达式2，满足进入，否则退出循环

3)顺序执行循环体内容

4)执行表达式3，然后转入2

注意事项：

for循环中三个表达式都不是必须，不管是否有内容，执行过程不变

循环编写的时候，要特别注意边界

逻辑错误很难排查

## While循环语句

while语句是另一种循环语句,其一般形式如下

**while(表达式)**

**{**

**语句；**

**}**

**执行过程：**

**1)判断表达式是否为真，满足进入2)，否则退出循环**

**2)顺序执行循环体内容，完毕后进入1)**

在这里，*while* 循环的关键点是循环可能一次都不会执行。当条件为 false 时，会跳过循环主体，直接执行紧接着 while 循环的下一条语句。



## Do—while语句

do…while语句是第三种循环语句，基本形式：

**do**

**{**

**语句;** **//** **被称为do** **语句体**

**}** **while(表达式);**

**注意：do...while循环语句至少执行一次(与前面两种语句根本区别:结构不同)**

**执行过程：**

**1)先执行一次循环体内容**

**2)判断表达式是否为真，满足进入3)，否则退出循环**

**3)顺序执行循环体内容，完毕后进入2)**

## 中断语句

**循环语句的执行方式可以被改变：**

break：停止整个循环的执行，执行循环后的内容

continue：停止当次循环执行，执行下一次循环

goto：跳到指定标号执行

**goto语句的一般形式为：goto 命名标号;**

标号及紧随其后的语句形成了一个标号语句，标号语句的一般形式是：

标号：

-语句

# 5、代码练习

## 练习一：输入三个数，将这三个数从小到大的依次输出

#include<stdio.h>

#include<stdlib.h>

int main(void)

{

int x,y,z,t;

printf("请输入三个整数（以空格隔开）:");

scanf("%d %d %d",&x,&y,&z);

if (x>y)

{

t=x;

x=y;

y=t;

}

if (x>z)

{

t=x;

x=z;

z=t;

}

if (y>z)

{

t=y;

y=z;

z=t;

}

printf("%d,%d,%d",x,y,z);

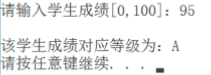
system("pause");

return 0;

}

## 练习二：输入成绩的级别

**编写代码实现：根据用户输入的百分制成绩，要求输出成绩等级“A”、“B”，“C”，“D”，“E”。90 分及以上的为：“A”级，80~89 分的为“B”，70~79 分的为“C”，60~69 分的为“D”，60 分以下的为“E”。运行效果：**



#include<stdio.h>

#include<stdlib.h>

int main(void)

{

int x;

printf("请输入学生成绩[0,100]:");

scanf("%d",&x);

if (x>=90&&x<=100)

{

printf("该学生成绩对应等级为：A");

}

else

{

if (x>=80&&x<=89)

{

printf("该学生成绩对应等级为：B");

}

else

{

if (x>=70&&x<=79)

{

printf("该学生成绩对应等级为：C");

}

else

{

if (x>=60&&x<=69)

{

printf("该学生成绩对应等级为：D");

}

else

{

if (x<=60&&x>=0)

{

printf("该学生成绩对应等级为：E");

}

else

{

printf("您输入的不在范围之内");

}

}

}

}

}

system("pause");

return 0;

}

## 练习三：判断某一天是星期几

#include<stdio.h>

#include<stdlib.h>

int main(void)

{

int x;

printf("请输入今天是星期几：（用数字进行表示）");

scanf("%d",&x);

switch(x)

{

case 1:

printf("Today is Monday");

break;

case 2:

printf("Today is Tuseday");

break;

case 3:

printf("Today is Wednesday");

break;

case 4:

printf("Today is Thursday");

break;

case 5:

printf("Today is Friday");

break;

case 6:

printf("Today is Saturday");

break;

case 7:

printf("Today is Sunday");

break;

default:

printf("您输入的数据不符合要求！！！");

}

system("pause");

return 0;

}

## 练习四：输入某年某月某日，判断这一天是这一年的第几天？

#include <stdio.h>

#incldde<stdlib.h>

int main(void)

{

int day,month,year,sum,leap;

printf("\n请输入年、月、日，格式为：年,月,日（2015,12,10）\n");

scanf("%d,%d,%d",&year,&month,&day); // 格式为：2015,12,10

switch(month) // 先计算某月以前月份的总天数

{

case 1:sum=0;break;

case 2:sum=31;break;

case 3:sum=59;break;

case 4:sum=90;break;

case 5:sum=120;break;

case 6:sum=151;break;

case 7:sum=181;break;

case 8:sum=212;break;

case 9:sum=243;break;

case 10:sum=273;break;

case 11:sum=304;break;

case 12:sum=334;break;

default:printf("data error");break;

}

sum=sum+day; // 再加上某天的天数

if(year%400==0||(year%4==0&&year%100!=0)) {// 判断是不是闰年

leap=1;

} else {

leap=0;

}

if(leap==1&&month>2) { // \*如果是闰年且月份大于2,总天数应该加一天

sum++;

}

printf("这是这一年的第 %d 天。",sum);

printf("\n");

system(“pause”);

return 0;

}

## 练习五：判断2000年到2500年之间有多少个闰年

#include<stdio.h>

#include<stdlib.h>

int main(void)

{

int x,i;

i=0;

x=2000;

for (;x<=2500;x++)

{

if (x%400==0||(x%100!=0&&x%4==0))

i++;

}

printf("%d",i);

system("pause");

return 0;

}