

# 第2章

例子驱动的C语言语法元素概览

# 变量与表达式

```
#include <stdio.h>
#include <math.h>
/* 打印一个周期内，三角函数的离散值表 */
int main()
{
    int x;
    double y;
    int start, end, step;
    start = 0; /* 角度下限 */
    end = 360 ; /* 角度上限 */
    step = 30; /* 步长 */
    x = start;
    while (x <= end) {
        y = sin(x*3.1415926/180);
        printf("%d\t\t%f\n", x, y);
        printf("\n");
        x = x + step;
    }
    return 0;
}
```

```

#include <stdio.h>
#include <math.h>
/* 打印一个周期内，三角函数的离散值表 */
int main()
{
    int x;
    double y;
    int start, end, step;
    start = 0; /* 角度下限 */
    end = 360 ; /* 角度上限 */
    step = 30; /* 步长 */
    x = start;
    while (x <= end) {
        y = sin(x*3.1415926/180);
        printf("%d\t\t%f\n", x, y);
        printf("\n");
        x = x + step;
    }
    return 0;
}

```

0	0.000000
30	0.500000
60	0.866025
90	1.000000
120	0.866025
150	0.500000
180	0.000000
210	-0.500000
240	-0.866025
270	-1.000000
300	-0.866025
330	-0.500000
360	-0.000000

输出

```
#include <stdio.h>
#include <math.h>
/* 打印一个周期内，三角函数的离散值表 */
int main()
{
    int x;
    double y;
    int start, end, step;
    start = 0; /* 角度下限 */
    end = 360; /* 角度上限 */
    step = 30; /* 步长 */
    x = start;
    while (x <= end) {
        y = sin(x*3.1415926/180);
        printf("%d\t\t%f\n", x, y);
        printf("\n");
        x = x + step;
    }
    return 0;
}
```

关键字



# 编译预处理

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



```
/* 打印一个周期内，三角函数的离散值表 */
```

```
int main()
```

```
{
```

```
    int x;
```

```
    double y;
```

```
    int start, end, step;
```

```
    start = 0; /* 角度下限 */
```

```
    end = 360; /* 角度上限 */
```

```
    step = 30; /* 步长 */
```

```
    x = start;
```

```
    while (x <= end) {
```

```
        y = sin(x*3.1415926/180);
```

```
        printf("%d\t\t%f\n", x, y);
```

```
        printf("\n");
```

```
        x = x + step;
```

```
    }
```

```
    return 0;
```

```
}
```

```
#include <stdio.h>
#include <math.h>
/* 打印一个周期内，三角函数的离散值表 */
```

```
int main()
```

```
{
```

```
    int x;
```

```
    double y;
```

```
    int start, end, step;
```

```
    start = 0; /* 角度下限 */
```

```
    end = 360; /* 角度上限 */
```

```
    step = 30; /* 步长 */
```

```
    x = start;
```

```
    while (x <= end) {
```

```
        y = sin(x*3.1415926/180);
```

```
        printf("%d\t\t%f\n", x, y);
```

```
        printf("\n");
```

```
        x = x + step;
```

```
    }
```

```
    return 0;
```

```
}
```

主函数



```
#include <stdio.h>
#include <math.h>
/* 打印一个周期内，三角函数的离散值表 */
int main()
{
    int x;
    double y;
    int start, end, step;
    start = 0; /* 角度下限 */
    end = 360; /* 角度上限 */
    step = 30; /* 步长 */
    x = start;
    while (x <= end) {
        y = sin(x*3.1415926/180);
        printf("%d\t\t%f\n", x, y);
        printf("\n");
        x = x + step;
    }
    return 0;
}
```

← 变量声明



```
#include <stdio.h>
#include <math.h>
/* 打印一个周期内，三角函数的离散值表 */
int main()
{
    int x;
    double y;
    int start, end, step;
    start = 0; /* 角度下限 */
    end = 360; /* 角度上限 */
    step = 30; /* 步长 */
    x = start;
    while (x <= end) {
        y = sin(x*3.1415926/180);
        printf("%d\t\t%f\n", x, y);
        printf("\n");
        x = x + step;
    }
    return 0;
}
```

变量赋值

```
#include <stdio.h>
#include <math.h>
/* 打印一个周期内，三角函数的离散值表 */
int main()
{
    int x;
    double y;
    int start, end, step;
    start = 0; /* 角度下限 */
    end = 360; /* 角度上限 */
    step = 30; /* 步长 */
    x = start;
    while (x <= end) {
        y = sin(x*3.1415926/180);
        printf("%d\t\t%f\n", x, y);
        printf("\n");
        x = x + step;
    }
    return 0;
}
```

表达式

分支

if

统计C语言程序设计课程期末考试各成绩段的人数，统计分成下述几档：

0-59 不及格

60-69 及格

70-79 中等

80-89 良好

90-99 优秀

100 满分

```
if (score < 60)
    grade[0]++;
else if (score < 70)
    grade[1]++;
else if (score < 80)
    grade[2]++;
else if (score < 90)
    grade[3]++;
else if (score < 100)
    grade[4]++;
else
    grade[5]++;
```

```
if (score < 60)
    grade[0]++;
else if (score < 70)
    grade[1]++;
else if (score < 80)
    grade[2]++;
else if (score < 90)
    grade[3]++;
else if (score < 100)
    grade[4]++;
else
    grade[5]++;
```

依次求值



switch



统计C语言程序设计课程期末考试各成绩段的人数，考试采用5分制

A 优秀

B 良好

C 中等

D 及格

E 不及格

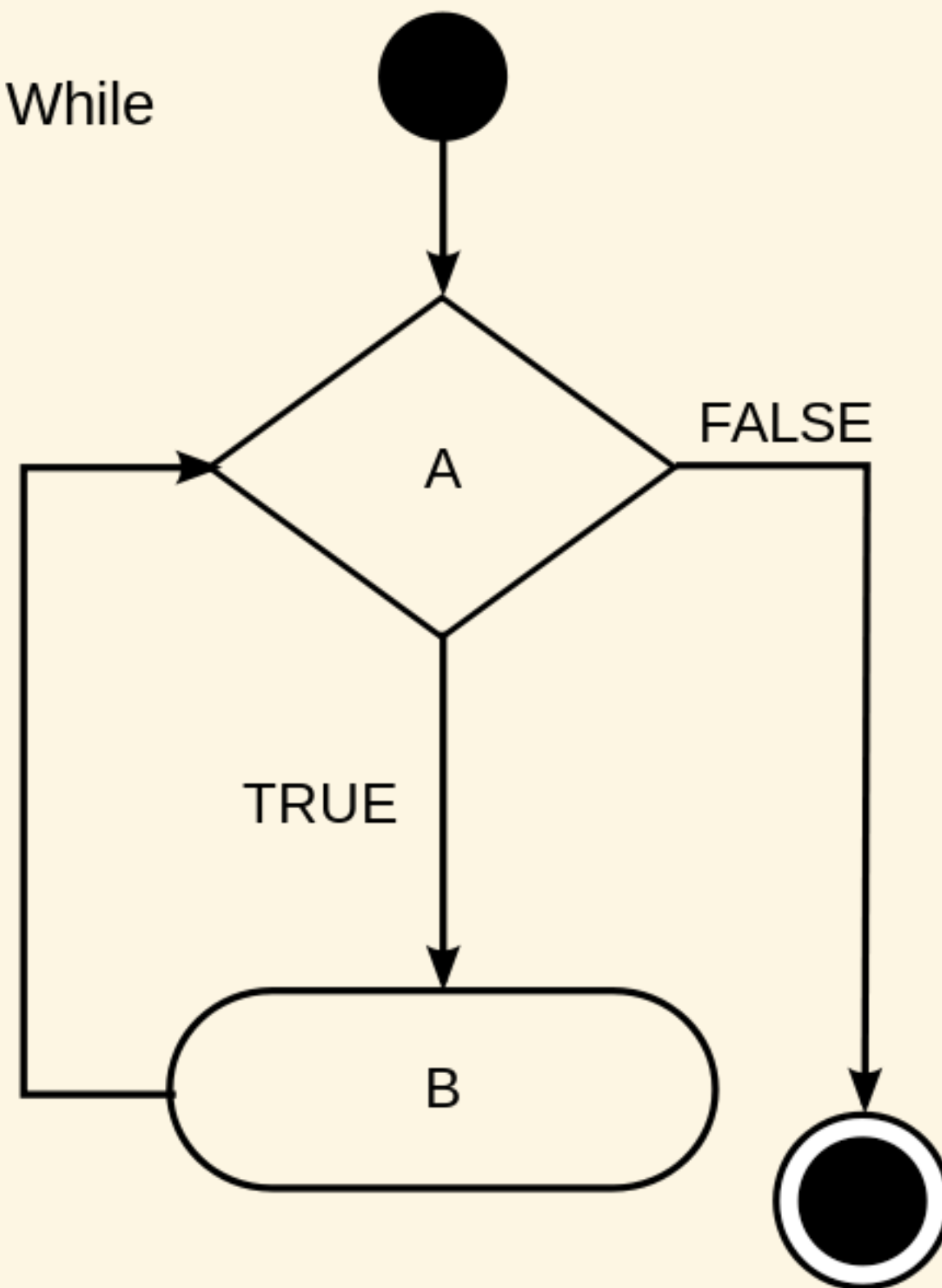
```
switch(score) {  
case 'A':  
    grade[0]++;  
    break;  
case 'B':  
    grade[1]++;  
    break;  
case 'C':  
    grade[2]++;  
    break;  
case 'D':  
    grade[3]++;  
    break;  
case 'E':  
    grade[4]++;  
    break;  
default:  
    break;  
}
```

循环

While (A = TRUE) Do

B

End While



while

```
while (cond_exp)
    loop_body_statement
```

```
while (x <= end) {
    ...
}
```

```
while (i < j)
    i = 2 * i
```

for

```
#include <stdio.h>
#include <math.h>
/* 打印一个周期内，三角函数的离散值表 */
int main()
{
    int x;
    double y;
    for (x = 0; x <= 360; x=x+30) {
        y = sin(x*3.1415926/180);
        printf("%d\t\t%f\n", x, y);
        printf("\n");
    }
    return 0;
}
```



符号常量

```
#include <stdio.h>
#include <math.h>
/* 打印一个周期内，三角函数的离散值表 */
int main()
{
    int x;
    double y;
    for (x = 0; x <= 360; x=x+30) {
        y = sin(x*3.1415926/180);
        printf("%d\t\t%f\n", x, y);
        printf("\n");
    }
    return 0;
}
```

Magic Number

# 宏定义

```
#include <stdio.h>
#include <math.h>
/* 打印一个周期内，三角函数的离散值表 */
#define PI 3.1415926
#define START 0
#define END 360
#define STEP 30

int main()
{
    int x;
    double y;
    for (x = START; x <= END; x=x+STEP) {
        y = sin(x*PI/180);
        printf("%d\t\t%f\n", x, y);
        printf("\n");
    }
    return 0;
}
```

为什么要用宏定义？

输入输出

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

```
/* 打印一个周期内，三角函数的离散值表 */
```

```
#define PI 3.1415926
```

```
int main()
```

```
{
```

```
    int x;
```

```
    double y;
```

```
    int start, end, step;
```

```
    printf("请输入周期起始点、终止点和步长\n");
```

```
    scanf("%d%d%d", &start, &end, &step);
```

```
    for (x = start; x <= end; x=x+step)
```

```
    {
```

```
        y = sin(x*PI/180);
```

```
        printf("%d\t\t%f\n", x, y);
```

```
        printf("\n");
```

```
    }
```

```
    return 0;
```

```
}
```

输出

输入

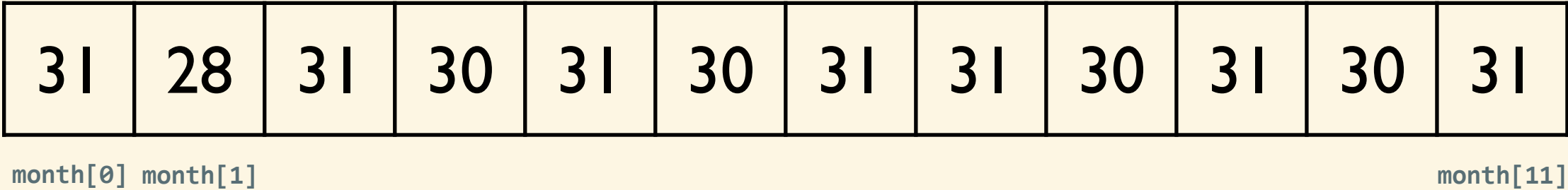


# 数组

表示12个月天数

```
int jan = 31;  
int feb = 28;  
int mar = 31;  
int apr = 30;  
int may = 31;  
int jun = 30;  
int jul = 31;  
int aug = 31;  
int sep = 30;  
int oct = 31;  
int nov = 30;  
int dec = 31;
```

```
int month[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
```



# 函数

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

```
int factorial (int n);
```

```
int main()
```

```
{
```

```
    int i;
```

```
    for ( i =0; i < 10; ++i )
```

```
        printf("%d的阶乘是: %d\n", i, factorial(i) );
```

```
    return 0;
```

```
}
```

```
int factorial(int n)
```

```
{
```

```
    int i, p;
```

```
    p = 1;
```

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

```
        p = p * i;
```

```
    return p;
```

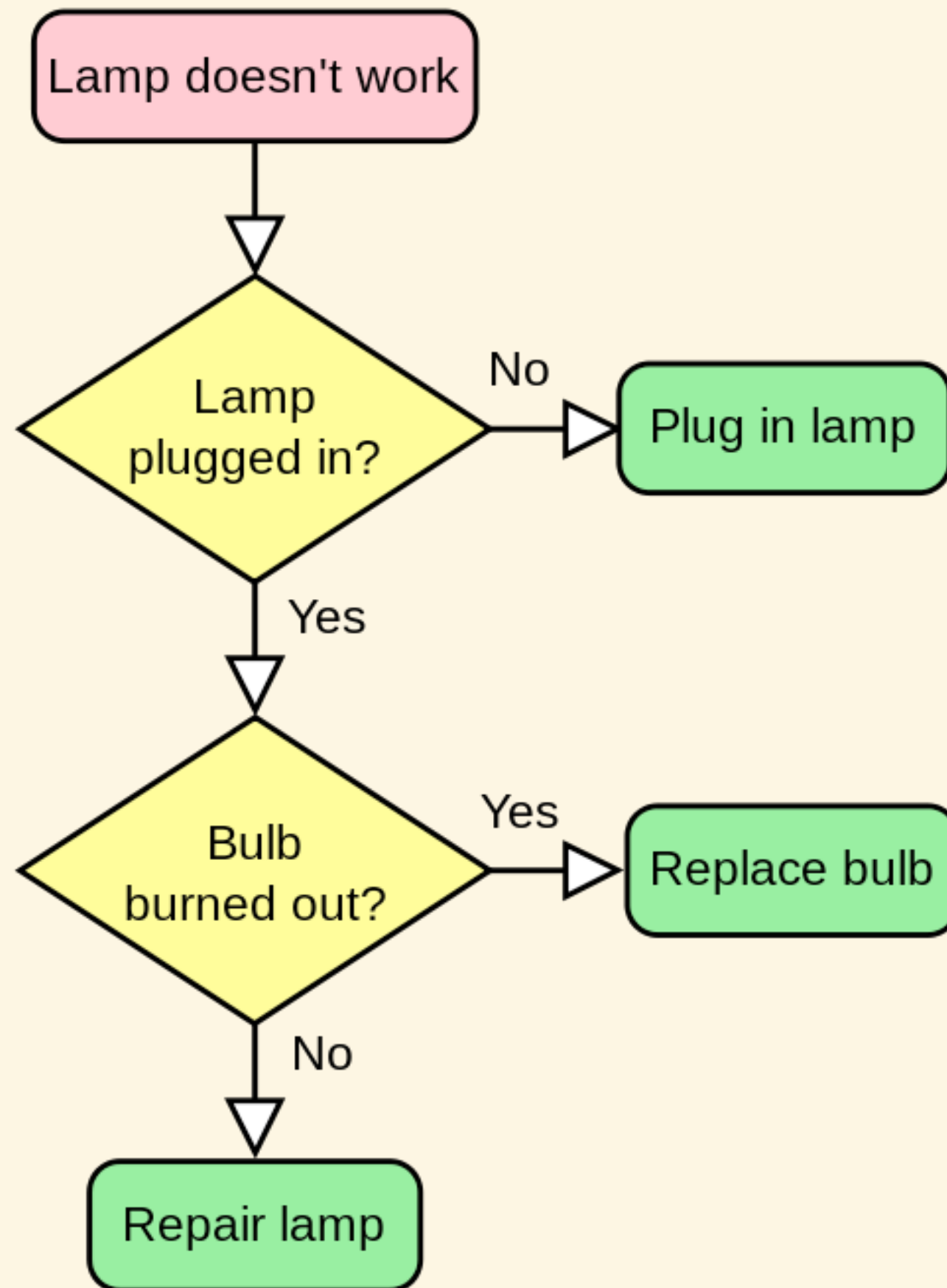
```
}
```

为什么要用函数？

# 算法



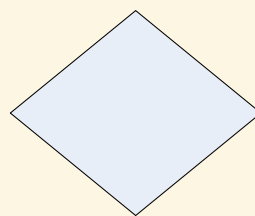
- 算法 (Algorithm) 为一个计算的具体步骤，常用于计算、数据处理和自动推理。精确而言，算法是一个表示为有限长列表的有效方法。算法应包含清晰定义的指令用于计算函数。



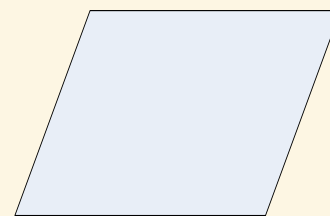
# 流程图



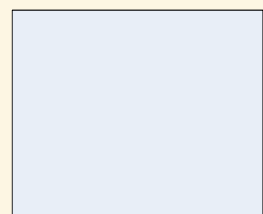
起始框



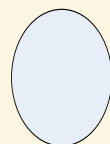
判断框



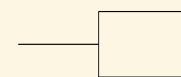
输入、输出框



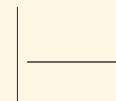
执行框



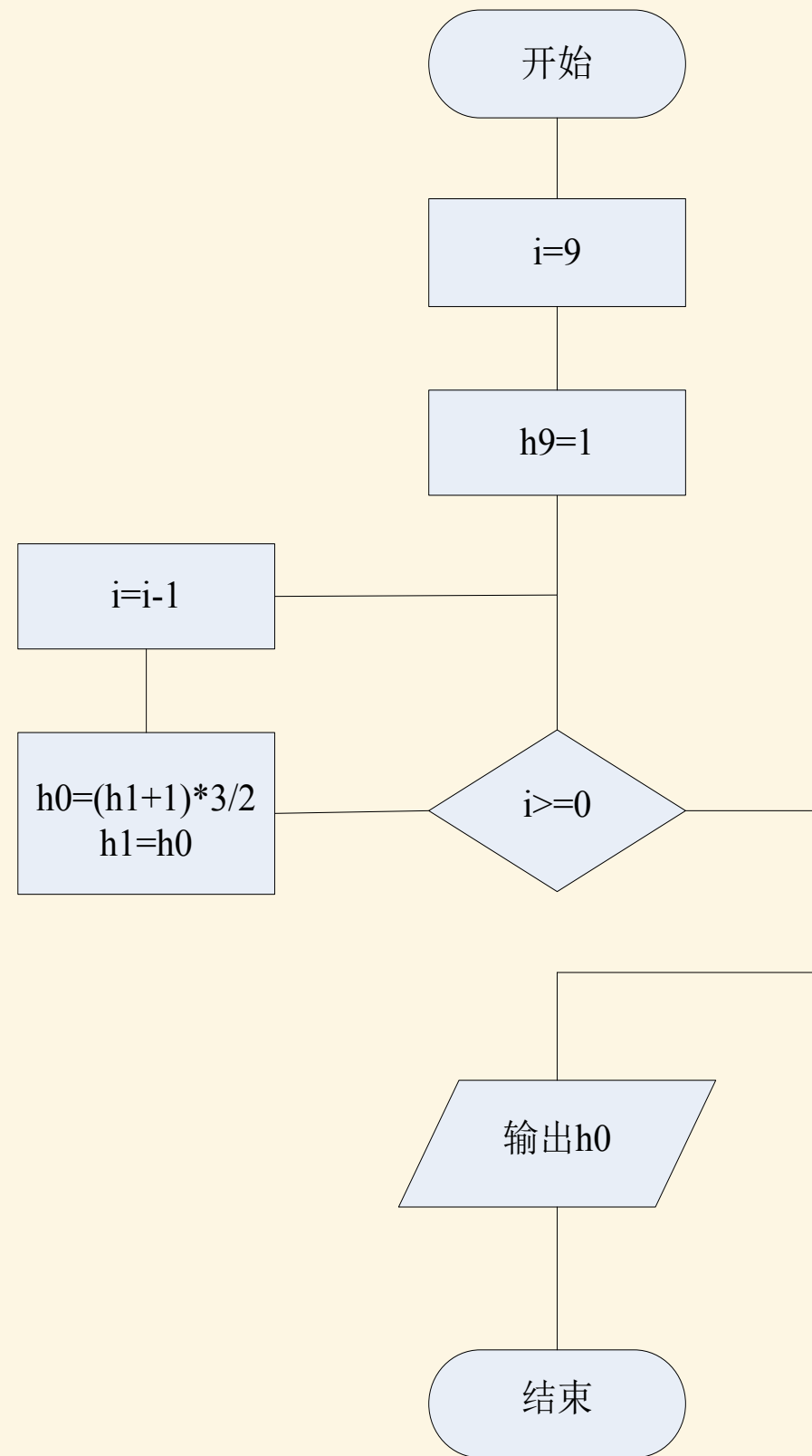
连接点



注释框



流程线



# 课后练习

- 安装Visual Studio C++
- 测试运行书上的例子