

# 第五章 程序的控制流程

## 本章学习目标：

- ✓ 了解算法的概念及描述方法
- ✓ 掌握算法的 3 种基本结构
- ✓ 掌握顺序结构、条件结构、循环结构对应流程图的特征
- ✓ 熟练使用 if、if-else、switch、for、while、do...while 等语句。
- ✓ 能设计简单的算法，并根据算法编写程序
- ✓ 能读懂算法的流程图，并根据流程图编写程序

## 5.1 实践题

### 一、多路分支语句的应用

#### 实验目的

1. 熟练掌握用 switch 语句实现多分支结构程序设计。
2. 熟练掌握 switch 语句中 break 语句的作用。
3. 学会调试程序。

#### 实验步骤

问题描述：用多分支条件判断语句（switch）编程设计一个简单的计算器程序，具有加减乘除四则运算功能。输入由键盘输入，共 1 行，为需要计算的数学表达式。表达式由实数 a，运算符 opt，实数 b 组成，a 和 b 是操作数，opt 是算术运算符，包括"+"（加）"-"（减）"\*"（乘）"/"（除）。输入如：“1+2”，输出如：“1.000000+2.000000=3.000000”。

步骤 1：在 vs2012 中新建项目“Lab5-1”，

步骤 2：在项目 Lab5-1 中，选择【源文件】→【添加】→【新建项】，创建“calculator.c”

源文件

步骤 3：打开 calculator.c 源文件，添加如下代码：

```
#include<stdio.h>

#include<stdlib.h>

int main(void)
{
```

```
double a, b; //定义操作数
char op;      //定义操作符

printf("hint:输入需要计算的数学表达式,如 1+2\n");
scanf("%lf%c%lf", &a, &op, &b); //输入需要计算的数学表达式
```

```
switch (op)
{
    case '+':
    {
        printf("%lf + %lf = %lf\n", a, b, a + b);
        break;
    }
    case '-':
    {
        printf("%lf - %lf = %lf\n", a, b, a - b);
        break;
    }
    case '*':
    {
        printf("%lf * %lf = %lf\n", a, b, a * b);
        break;
    }
    case '/':
    {
        if (b != 0)
            printf("%lf / %lf = %lf\n", a, b, a / b);
        else
            printf("除数不能是 0! \n");
        break;
    }
    default:
        printf("操作符错误! \n");
}
```

```

    }

    system("pause");

    return 0;

}

```

## 实验结果/结论

### 1. 实验结果

- ✓ 测试加法运算，显示结果如下：

```

hint:输入需要计算的数学表达式,如 1+2
3+5
3.000000 + 5.000000 = 8.000000
请按任意键继续. . .

```

- ✓ 测试减法运算，显示结果如下：

```

hint:输入需要计算的数学表达式,如 1+2
6-8
6.000000 - 8.000000 = -2.000000
请按任意键继续. . .

```

- ✓ 测试乘法运算，显示结果如下：

```

hint:输入需要计算的数学表达式,如 1+2
8*9
8.000000 * 9.000000 = 72.000000
请按任意键继续. . .

```

- ✓ 测试除法运算，显示结果如下：

```

hint:输入需要计算的数学表达式,如 1+2
50/3
50.000000 / 3.000000 = 16.666667
请按任意键继续. . .
hint:输入需要计算的数学表达式,如 1+2
50/0
除数不能是 0!
请按任意键继续. . .

```

- ✓ 测试非法操作符，显示结果如下：

```

hint:输入需要计算的数学表达式,如 1+2

```

50>3

操作符错误!

请按任意键继续. . .

## 2. 实验结论

- ✓ 通过本实验, 熟练掌握 switch 语句的语法及使用场景。
- ✓ 学会 switch 语句和 break 语句的结合使用。当 case 后面的常量表达式的值等于 switch 语句中的表达式的值时, 将执行该 case 后面的语句。值得注意的是, 如果语句后面有 break, 那么执行完语句就退出 switch 语句, 否则将继续往下执行其他 case 后面的语句, 直到遇到 break;

## 二、 循环语句 while 的应用

### 实验目的

1. 理解循环语句的执行过程。
2. 熟练掌握用 while 语句实现循环结构程序设计。
3. 熟练掌握循环语句的应用场景。

### 实验步骤

问题描述: 利用  $e=1+1/1!+1/2!+1/3!+\dots+1/n!$ , 编程实现计算 e 的近似值, 直到最后一项的绝对值小于  $10^{-5}$  才停止循环, 输出 e 的最后值, 并给出 n (即总共计算次数)。

步骤 1: 在 VS2012 中新建项目 “Lab5-2”,

步骤 2: 在项目 Lab5-2 中, 选择【源文件】→【添加】→【新建项】, 创建 “ComputingE.c”

源文件

步骤 3: 打开 ComputingE.c 源文件, 添加如下代码:

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

int main(void){
    double e=1;
    double fac=1;//求阶乘, 并存入 fac 中
    int i=1;
    while((1/fac)>=(1e-5))
    {
        e=e+1/fac;
        i++;
        fac=fac*i;
    }
    printf("n=%d,e=%f\n",i-1,e);

    system("pause");
    return 0;
}
```

### 实验结果/结论

### 1. 实验结果

✓ 结果如下：

n=8,e=2.718279

请按任意键继续. . .

### 4. 实验结论

通过本次实验，掌握如下知识点：

- ✓ 在循环语句中要有使其终止的语句，否则将会陷入死循环；
- ✓ 当循环体有多条语句时，应该用花括号将所有循环体的语句括起来，否则只执行第一条语句；
- ✓ 能够分析：循环体执行了多少次？循环结束后，i 的值是多少？

## 5.2 理论题

### A 类

#### 一、填空题

1. 请写出 float x 与“零值”比较的 if 语句\_\_\_\_\_。
2. 当 m=2,n=1,a=1,b=2,c=3 时，执行完 d=(m=a!=b)&&(n=b>c)后，n 的值为\_\_\_\_\_，m 的值为\_\_\_\_\_。
3. 分析下述程序的运行结果\_\_\_\_\_。

```
int main(void)
{

    char x='B';
    switch(x)
    {
        case 'A': printf("It is A.\n");
        case 'B': printf("It is B.\n");
        case 'C': printf("It is C.\n");
        default: printf("other.\n");
    }

    system("pause");
    return 0;
}
```

4. 分析以下程序，写出程序的运行结果\_\_\_\_\_。

```
int main(void)
```

```

{
    int i = 0;
    switch (i)
    {
        default:
            printf("default\n"); break;
        case 1:
            printf("case 1\n"); break;
        case 2:
            printf("case 2\n"); break;
        case 3:
            printf("case 3\n"); break;
    }
    system("pause");
    return 0;
}

```

5. 以下程序的运行结果是\_\_\_\_\_。

```

#include <stdio.h>
main()
{
    int a=1;
    if (a++>1)
        printf("%d\n", a);
    else
        printf("%d\n", a--);
}

```

## 二、选择题

- 若  $x$  是 `int` 型变量,  $y$  是 `float` 型变量, 所用的 `scanf` 调用语句格式为:  
`scanf("x=%d,y=%f",&x,&y);` 则为了将数据 10 和 66.6 分别赋给  $x$  和  $y$ , 正确的输入应是 ( )。  
 A.  $x=10,y=66.6$ <回车>                      B. 10 66.6<回车>  
 C. 10<回车>66.6<回车>                      D.  $x=10$ <回车> $y=66.6$ <回车>
- 若有定义 `double x=1, y;` 则以下的语句执行的结果是 ( )。  
`y=x+3/2; printf("%f",y);`  
 A. 2.500000                      B. 2.5  
 C. 2.000000                      D. 2
- 程序设计的三种基本结构是 ( )  
 A. 顺序结构、分支结构、选择结构  
 B. 递归结构、循环结构、选择结构  
 C. 顺序结构、循环结构、逻辑结构

D. 顺序结构、循环结构、选择结构

4. 如下所示，以下程序的运行结果是 ( )

```
int main(void)
{
    int m = 5;
    if (m++ > 5)
        printf("%d\n", m);
    else
        printf("%d\n", m--);
}
```

A、4    B、5    C、6    D、7

5. 为了避免嵌套的 if-else 语句的二义性，C 语言规定 else 总是与 ( ) 组成配对关系。

A. 缩排位置相同的 if                      B. 在其之前未配对的 if  
C. 在其之前未配对的最近的 if            D. 同一行上的 if

6. 分析以下程序，下列说法正确的是 ( )。

```
main()
{
    int x=0, a=0, b=0;
    if(x==a+b) printf("* * * *\n");
    else printf("# # # #\n");
}
```

A. 有语法错，不能通过编译                      B. 通过编译，但不能连接  
C. 输出\* \* \* \*                                      D. 输出# # # #

7. 下述 for 循环语句，描述正确的是 ( )。

```
int main(void)
{
    int i,k;
    for(i=0,k=-1;k=1;i++,k++)
    {
        printf("* * * *\n");
    }

    system("pause");
    return 0;
}
```

A. 判断循环结束的条件非法                      B. 是无限循环  
C. 只循环一次                                      D. 一次也不循环

8. 下列程序段执行后 k 值为 ( )。

```
int k=0,i,j;
for(i=0;i<5;i++)
```

```
for(j=0;j<3;j++)
```

```
    k=k+1 ;
```

A. 15      B. 3      C. 5      D. 8

9. 程序段如下：则以下说法中不正确的是（ ）。

```
int main(void)
{
    int k=2;
    while(k<7)
    {
        if(k%2)
        {
            k=k+3;
            printf("k=%d\n",k);continue;
        }
        k=k+1;
        printf("k=%d\n",k);
    }
}
```

- A. k=k+3; 执行一次      B. k=k+1; 执行 2 次  
C. 执行后 k 值为 7      D. 循环体只执行一次

10. C 语言中 while 和 do while 循环的主要区别是（ ）。

- A. do-while 的循环至少无条件执行一次。  
B. while 循环控制条件比 do-while 的循环控制条件严格。  
C. do-while 允许从外部转入到循环体内。  
D. do-while 的循环体不能是复合语句。

### 三、综合题

- 编程实现：从键盘上输入一个字符，如果是大写字母，就将其转换为小写字母；否则不转换。
- 编程实现：由计算机生成一个 1~100 之间的整数请用户猜，若猜对则结束游戏，并输出用户猜测次数（输出格式：“Right, [猜测次数] time(s)”）；若猜错则给出提示——若用户所猜数偏大，则输出“Big”，否则输出“Small”。一次游戏最多可以猜 10 次。若经过 10 次仍未猜中的话，输出“over”并结束游戏。
- 编程用 for 循环打印如下格式的九九乘法口诀。

```
1*1=1;
```

```
1*2=2; 2*2=4;
```

```
1*3=3; 2*3=6; 3*3=9;
```

```
...
```



## B 类

### 一、填空题

1. 分析以下程序，写出程序的运行结果\_\_\_\_\_。

```
int main(void) {  
    int a=-1, b=4, k;  
    k=(a++>=0)&&(! (b--<=0));  
    printf("%d, %d, %d\n", k, a, b);  
  
    system("pause");  
    return 0;  
}
```

2. 功能：“按照顺序输入 A、B、C 三个数，如果  $A < B < C$ ，那么打印输出“输入数据呈现递增规律”的信息，否则打印输出“输入数据呈现非递增规律”的信息”。下述代码能否实现此功能？\_\_\_\_\_。

```
int main(void)  
{  
    int A,B,C;  
  
    printf("请依次输入A、B、C的值:");  
    scanf("%d%d%d",&A,&B,&C);  
  
    if(A<B)  
        if(B<C)  
            printf("输入数据呈现递增规律");  
    else  
        printf("输入数据呈现非递增规律");  
  
    system("pause");  
    return 0;  
}
```

3. 分析以下程序，写出程序的运行结果\_\_\_\_\_。

```
int main(void)  
{  
    int x=1,y=0,a=0,b=0;  
    switch(x)  
    {
```

```

        case 1:
            switch(y)
            {
                case 0: a++;break;
                case 1: b++;break;
            }
        case 2: a++;b++;break;
        case 3: a++;b++;
    }
    printf("a=%d,b=%d\n",a,b);
    system("pause");
    return 0;
}

```

## 二、选择题

1. 语句 `while(!E);` 中的表达式 `!E` 等价于 ( )。  
 A、 `E == 0`                      `E != 1`  
 C、 `E != 0`                      `E == 1`
2. 指出下列程序段所表示的逻辑关系是 ( )。

```

if(a<b)
{
    if(c==d)
        x=10;
}
else
    x=-10;

```

- A. 
$$x = \begin{cases} 10 & a < b \text{ 且 } c = d \\ -10 & a \geq b \text{ 且 } c \neq d \end{cases}$$
- B. 
$$x = \begin{cases} 10 & a < b \text{ 且 } c = d \\ -10 & a \geq b \end{cases}$$
- C. 
$$x = \begin{cases} 10 & a < b \text{ 且 } c = d \\ -10 & a < b \text{ 且 } c \neq d \end{cases}$$
- D. 
$$x = \begin{cases} 10 & a < b \text{ 且 } c = d \\ -10 & c \neq d \end{cases}$$

3. 分析以下程序，下列说法正确的是 ( )。

```

main()
{
    int x=5, a=0, b=1;
    if(x=a+b) printf("* * * *\n");
    else    printf("# # # #\n");
}

```

- A. 有语法错，不能通过编译
- B. 通过编译，但不能连接
- C. 输出\* \* \* \*
- D. 输出# # # #

4. 程序段如下：则以下说法中正确的是（）。

```

int main(void)
{
    int count=5;
    do{
        count--;
    }while(count<=0);
    system("pause");
    return 0;
}

```

- A. 循环执行 5 次
- B. 循环是无限循环
- C. 循环体语句一次也不执行
- D. 循环体语句执行一次

5. 程序段如下：则以下说法中正确的是（）。

```

int main(void)
{
    int k=-20;
    while(k=0)
    {
        k=k+1;
        printf("--\n");
    }
}

```

- A. while 循环执行 20 次
- B. 循环是无限循环
- C. 循环体语句一次也不执行
- D. 循环体语句执行一次

6. 下面程序段的运行结果是（）。

```

int main(void)
{
    int a=1;
    int b=2;
    int c=2;
    int t;

    while(a<b<c)
    {

```

```

        t=a;
        a=b;
        b=t;
        c--;
    }
    printf("%d,%d,%d",a,b,c);
}

```

- A. 1,2,0    B. 2,1,0    C. 1,2,1    D. 2,1,1

7. 下面程序段的运行结果是 ( )。

```

int main(void)
{
    int x=0;
    int y=0;
    while(x<15)
        y++,x+=++y;
    printf("%d,%d",y,x);
}

```

- A. 20,7    B. 6,12    C. 20,8    D. 8,20

8. 设有以下程序段，则以下说法中正确的是 ( )。

```

int main(void)
{
    int x=0,s=0;
    while(!x!=0)
        s+=++x;
    printf("s=%d\n",s);
}

```

- A. 运行程序段后输出 0                      B.运行程序段后输出 1  
C. 循环的控制表达式不正确              D.程序段执行无限次

9. 下面程序段的运行结果是 ( )

```

int main(void)
{
    int n=0;
    while(n++<=2);
    printf("%d\n",n);
}

```

- A. 2    B. 3    C. 4    D. 语法错误

## 二、综合题

- 编程实现：判断一个整数是奇数还是偶数，是奇数显示“奇数”，否则显示“偶数”。
- 指出下述程序的功能，并画出程序的流程图。

```

#include <stdio.h>
#include <stdlib.h>

```

```

int main(void)
{
    int a,n;          //a 是数字 1~9;n 是任意的整数
    int sum=0,x=0; //最后的求和结果, x 当前参与计算的值
    int i=0;          //临时变量

    printf("输入 a 值, a 是数字 1~9:");
    scanf("%d",&a);

    printf("输入 n 值, n 是任意的整数:");
    scanf("%d",&n);

    for(i=0;i<n;i++)
    {
        x=x*10+a;    //要参与求和计算的值 a..a(i+1 个 a,0<=i<n)
        sum+=x;      //求和
    }
    printf("sum=%d\n",sum);

    system("pause");
    return 0;
}

```

3. 设有一张厚度为 0.1mm、面积足够大的纸, 将它不断地对折。问对折多少次后, 其厚度可以超过 1m。
4. 编程由计算机生成一个 1~100 之间的整数请用户猜, 若猜对则输出用户猜测次数 (输出格式: “Right, [猜测次数] time(s)”)。若猜错则给出提示——若用户所猜数偏大, 则输出 “Big”, 否则输出 “Small”。一次游戏最多可以猜 10 次, 若经过 10 次仍未猜中的话, 输出 “over”; 一轮游戏结束之后, 询问用户是否进行下一轮游戏 (是否进行下一轮游戏? 是 (Y)), 若是进行下一轮游戏。
5. 有 0、1、2...9 十个数字, 编程计算这 10 个数字能组成多少个互不相同且无重复数字的三位数 (如: 123,124,125...), 并列举所组成的三位数

## 本章答案

### A 类

#### 一、填空题

1. const float EPSINON = 0.00001;

if ((x >= - EPSINON) && (x <= EPSINON))

分析: 不可将浮点变量用 “==” 或 “!=” 与数字比较, 应该设法转化成 “>=” 或 “<=” 此类形式。

2. 0 1  
3. It is B.  
It is C.  
other.

分析：本题考查的是不带 break 语句的 switch 语句的执行过程。case 'B': printf("It is B.\n"); 后面没有跟 break; 程序继续向下执行。

4. default。分析：本题考查的知识点是，在 switch 语句中，case 标号和 default 标号顺序是任意的。只要表达式不等于任何 case 标号的字面值，则程序控制转移到 default 标号。

5. 2

## 二、选择题

1.A 2.C 3.D

4.C。分析：if 表达式中参与比较的是 m 与 5，m 不大于 5，表达式的值为假，判断完成后 m 自增 1 即 m 值为 6，执行 else 分支后，输出 m 当前的值，所以输出结果为 6，且 m 自减 1，而此时 m 值为 5。

5.C 6.C 7.B

8.A。分析：本题是考查嵌套循环的时间复杂度。外循环执行五次，内循环执行三次，所以循环体 k=k+1 共执行 15 次。

9.D。k=k+3; 执行一次；. k=k+1; 执行 2 次，所以循环体共执行了三次。

10.A。

## 三、综合题

1. 代码如下：

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

int main(void)
{
    char c;
    c=getchar();

    //判断输入测字符是否为大写字母
    if(c>='A' && c<='Z')
    {
        //将大写字母转换为小写字母
        c=c+32;
    }

    printf("the char is: %c\n",c);
    system("pause");
    return 0;
}
```

分析：本题为单分支结构，只处理表达式为真的情况，其他情况不予处理。

2.代码如下:

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

int main(void)
{
    //guess 代表用户输入的值, ra 代表随机数值, maxTime 代表最大次数 10
    int guess,ra,maxTime;
    int i;

    ra = rand()%100;
    maxTime=10;

    printf("please guess a number in 0~100\n");

    for(i=1;i<=maxTime;i++)
    {
        printf(">");
        scanf("%d",&guess);

        //用户猜的值与随机生成数相同
        if(guess == ra)
        {
            printf("Right, [猜测次数] time(s): %d\n",i);
            break;
        }
        else if(guess < ra)
            printf("Small\n");
        else
            printf("Big\n");
    }

    if(i>maxTime)
        printf("over");

    system("pause");
    return 0;
}
```

3.代码如下:

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

int main(void)
```

```

{
    int i,j;

    //嵌套循环，输出乘法表
    for(i=1;i<10;i++)
    {
        for(j=1;j<=i;j++)
        {
            printf("%d*%d=%2d; ",j,i,i*j);
        }
        printf("\n");
    }
    //暂停屏幕
    system("pause");
    return 0;
}

```

## B 类

### 一、填空题

1. 输出结果：0 0 4。分析：本题考的是逻辑与短路的情况。A&&B，当 A 为 false 时，不去计算 B 的值而直接返回 false；当 A 为 true 时，计算 B 的值。表达式 a++>=0 中，a 先参与运算，再进行++运算，那么表达式 a++>=0 的结果为 0（a 的值也变为 0），最终 k 的值也为 0，根据逻辑与短路规则，将不再对表达式！（b--<=0）进行运算，所以 b 的值为 4。

2. 不能。

分析：当依次输入 A、B、C 的值：3 2 1 时，运行结果中不会输出“输入数据呈现非递增规律”信息。原因是代码中的 else 配对出现了问题，程序的本意是将 else 与第一个 if 语句配对，但是按照 if 语句的标准，else 应该与它最近的 if 语句配对。正确代码如下：

```

int main(void)
{
    int A,B,C;

    printf("请依次输入A、B、C的值:");
    scanf("%d%d%d",&A,&B,&C);

    if(A<B)
        if(B<C)
            printf("输入数据呈现递增规律");
        else
            printf("输入数据呈现非递增规律");
    else
        printf("输入数据呈现非递增规律");
}

```



```

        system("pause");
        return 0;
    }

```

3. 答案: a=2,b=1。分析: 本题考查的是 switch 语句的嵌套使用, 以及在 switch 中 break 语句的作用。switch 语句的执行方法: 若表达式的值等于 switch 语句中某个 case 标号中的整型字面值, 则程序控制转移到该 case 标号表示的点, 从此点开始执行。若碰见 break 语句, 则退出 switch 语句。

## 二、选择题

1. A 分析: 本题目考察对于逻辑真假值的理解, 题目中 while 循环内的条件若为真, 即!E 为真, 则 E 为假, 所以答案为 E==0。

2. B 3. C 分析: if 语句条件表达式  $x = a+b$ , 其中  $a+b$  的值为 1, 当执行赋值语句  $x=a+b$ ,  $x$  的值变为 1,  $if(x=a+b)$  等价于  $if(1)$ , 条件表达式成立, 执行 if 语句后面的内容。若将 if 后面的表达式改为  $x == a+b(5==1)$  则表达式的结果为 0,  $if(x=a+b)$  等价于  $if(0)$ , 条件表达式不成立, 执行 else 后面的内容。

4. D。分析: 本题考查 do...while 语句。do...while 语句是先执行后判断。不管条件如何, 至少先执行一次循环体内的语句, 然后再判断 while 后括号内表达式的值是否为真; 若为真, 继续重复执行循环体语句, 直到表达式的值为假为止。针对本题, 限制性循环体语句  $count--$  (循环体语句执行一次), 执行后 count 值为 4。判断表达式的值 " $count \leq 0$  (即  $4 \leq 0$ )" 为假, 跳出循环体。所以循环体语句共执行一次。

5. C。分析: 本题本意是当变量 k 等于 0 时, 退出循环体; 否则, 执行循环体内容。条件表达式应该写成  $while(k==0)$ , 但是由于将符号 == 写成 = 而造成程序错误。

建议书写判断条件时, 写成  $while(0==k)$  的形式, 这样写可以避免将符号 == 写成 = 而造成程序错误; 因为 0 不是左值, 不可以出现在赋值符号左边, 如果写成  $0 = k$ , 编译器会报语法错误; 如果写成  $k = 0$  则编译器不会报错。

4. A。分析, 本题要熟练掌握运算符的结合律, 才能正确的解题, "<" 的结合律是左结合, 即表达式  $a < b < c$  等价于  $(a < b) < c$ 。

5. D。分析: 本题考查逗号表达式及 ++ 元素符的使用。

6. B。分析: 注意运算符的优先级, 在表达式  $!x != 0$  中, ! 的优先级高于 !=。

7. D。分析: 本题考查前置 ++ 的使用及 ";" 语句的使用。在 while 语句的条件表达式满足的情况下, 执行的是 ";" 语句。每执行一次变量 n 加 1。当  $n=2$  时,  $n++ \leq 2$  条件表达式满足, 继续执行 ";" 语句, 此时 n 的值为 3。再一次判断  $n++ \leq 2$  条件表达式, 条件不满足, 退出 while 循环。虽然条件不成立, 但是由于执行了 ++ 操作, 此时 n 的值为 4。所以最终输出结果为 4。

## 三、综合题

1. 编程实现。

```

#include<stdio.h>
#include<stdlib.h>

int main(void)
{

```

```

int number;

printf("请输入一个整数\n");
scanf("%d",&number);

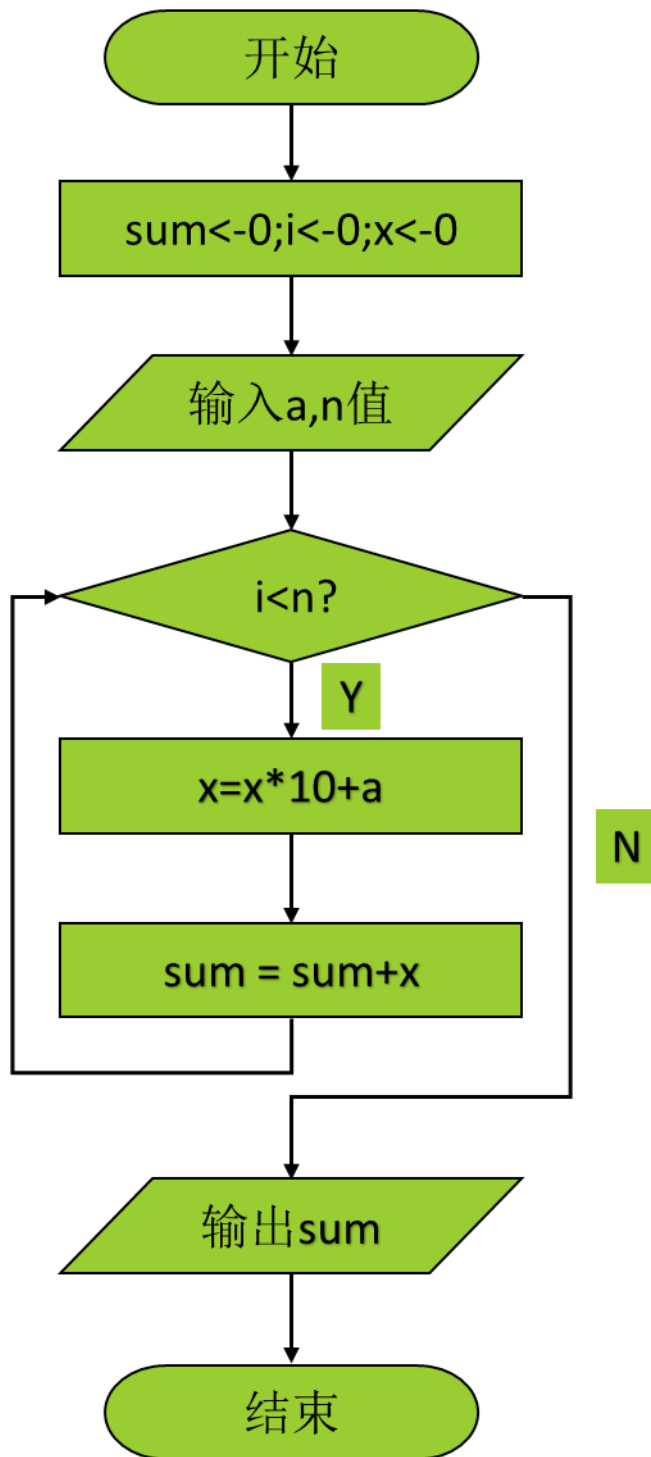
//判断输入的整数是否为偶数
if(number%2 == 0)
{
    printf("偶数");
}
else
{
    printf("奇数");
}

system("pause");
return 0;
}

```

本题为 if 语句的双分支结构：当表达式为真时，执行 if 后面的语句；当表达式结果为假时，执行 else 后面的语句。

2. 功能：计算： $a+aa+aaa+\dots+aa\dots a$  (n 个 a) 的值。如 输入： $a=2, n=4$ , 则  $sum=2+22+222+2222=2468$ 。  
流程图：



3. 编程实现。折叠次数为 14 次。

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

```
int main(void){
```

```

int n=0;                /*存储折叠的次数*/
double thickness=0.1;   /*折叠后纸的厚度，初值为 0.1*/

do
{
    thickness *= 2;      /*每次折叠厚度翻倍*/
    n++;
}while(thickness <= 1000); /*厚度增大到大于 1 米时终止循环*/

printf("折叠次数为: %d\n",n);
system("pause");
return 0;
}

```

#### 4. 编程实现。

```

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

int main(void)
{
    //guess 代表用户输入的值，ra 代表随机数值

    int guess,ra;

    //maxTime 代表用户可以猜词的最大次数

    int maxTime =10;

    int i=1;

    //标识是否进行下一轮猜词活动

    char isContinue ='Y';

    srand((unsigned)time(NULL));

    do{

        printf("please guess a number in 0~100\n");

        //生成随机数值 ra

```

```

    ra = rand() % 100 + 1; //产生随机数
    for(i=1;i<=maxTime;i++)
    {
        printf(">");
        scanf("%d",&guess);
        if(guess == ra)
        {
            printf("Right, [猜测次数] time(s): %d\n",i);
            break;
        }
        else if(guess < ra)
            printf("Small\n");
        else
            printf("Big\n");
    }
    if(i > maxTime)
        printf("over");

    fflush(stdin);
    printf("是否进行下一轮游戏? 是 (Y) \n");
    printf(">");
    scanf("%c",&isContinue);
} while(isContinue == 'Y');

system("pause");
return 0;
}

```

## 5. 编程实现。

```

#include<stdio.h>

#include<stdlib.h>


int main(void)
{
    //hun 代表百位数， ten 代表十位数字， single 代表个位数字

    int hun,ten,single;

    int number;

    for(hun=1;hun<10;hun++)

        for(ten=0;ten<10;ten++)

            {

                for(single=0;single<10;single++)

                    {

                        if (hun!=ten&&ten!=single&&single!=hun) /*确保三位互不相同
*/

                            {

                                number = hun*100+ten*10+single;

                                printf("%d\t",number);

                            }

                    }

                printf("\n");

            }

    system("pause");

    return 0;

}

```