# 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!+......+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. . . |

1. **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，否则将会限入死循环；
* 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;  } |

1. 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;  } |

1. 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--);  } |

1. **选择题**
2. 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>

1. 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

1. **程序设计的三种基本结构是（）**
2. 顺序结构、分支结构、选择结构
3. 递归结构、循环结构、选择结构
4. 顺序结构、循环结构、逻辑结构
5. 顺序结构、循环结构、选择结构
6. 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

1. 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

1. 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# # # #

1. 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

1. 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

1. 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

1. 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。

1. **综合题**
2. 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。
3. 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。
4. 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;  } |

1. 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;  } |

1. 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

1. 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．

B．

C．

D．

1. 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# # # #

1. 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

1. 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

1. 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);  } |

1. 1,2,0 B. 2,1,0 C. 1,2,1 D. 2,1,1
2. 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);  } |

1. 20,7 B. 6,12 C. 20,8 D. 8,20
2. 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

1. 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

1. **综合题**
2. 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”。
3. 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;  } |

1. 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.
2. 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。
3. 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

1. **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<=0(即4<=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。

1. 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。
2. D。分析：本题考查逗号表达式及++元素符的使用。
3. B。分析：注意运算法的优先级，在 表达式!x!=0中，!的优先级高于!=。
4. 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++<=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++<=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。 |

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

流程图：

|  |
| --- |
|  |

1. 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;  } |

1. 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;  } |

1. 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;  } |