

第 3 章

累 加 法

3.1 算法设计思想

累加是程序设计中最常遇见的问题,例如求某班级学生考试的总平均分;求某单位支出的总薪资等。累加是指在一个值的基础上重复加上其他值,典型的应用有求和、计数(统计出现的次数)等。

累加法的一般格式为 $S = S + T$, 其中, 变量 S 是累加器, 一般初值取 0; T 为每次的累加项, 通过累加项 T 的不断变化, 将所有的 T 值都累加到累加器 S 中。

注意: 格式中的两个 S 是不同的。“=”后面的 S 代表的是原来的值, 而“=”前面的 S 是 S 的原来值与累加项 T 的和。也就是说, S 保存累加过程中的中间结果和最后结果; 通常, 累加项 T 的变化是有规律的, 在设计算法时就是要找到其变化规律。

累加法一般由循环结构来实现, 在设计循环算法时, 需要确定以下 3 方面的内容。

- (1) 循环控制变量的初值、终值和步长。
- (2) 循环体的内容(一般包括语句: $S = S + T$;)。
- (3) 循环结束条件(一般与 T 有关)。

3.2 典型例题

3.2.1 自然数求和

1. 问题描述

求 $1 + 2 + 3 + \cdots + n$ 并输出结果。

2. 问题分析

这是对自然数进行累加的问题, 需要重复进行 n 次(或 $n-1$ 次)加法运算, 显然可以用循环结构来实现。重复执行循环体 n 次(或 $n-1$ 次), 每次加一个数; 可以看出每次相加的数是上一次的数加 1。使用一个变量 s 作为累加器, 用于记录累加的中间结果和最终结果, 初始值为 0; 使用一个

变量 t 作为循环控制变量, t 初始值为 1; 将 t 与 s 相加, 结果保存到 s 中, 然后 t 增加 1, 再与 s 相加, 不断重复, 直到 t 的值超过 n 为止, 结束循环。最后 s 保存的即为最终的累加和。

3. 算法说明

算法说明参见表 3-1。

表 3-1

类 型	名 称	代表的含义
算法	add(int n)	求自然数的累加和
形参变量	n	自然数, 进行累加的最大数
变量	s	累加器(和), 存放累加结果
变量	t	循环控制变量

4. 算法设计

```
#include "stdio.h"
int add(int n)
{
    int s=0;
    int t;                /* 用于保存逐步增加自然数的变量 t 赋初值为 1 */
    for(t=1;t<=n;t++)      /* t 从 1 到 n 执行循环体语句实现求和 */
    {
        s=s+t;
    }
    return s;              /* 返回 1 到 n 的自然数之和 */
}
void main()
{
    int s=0;                /* 用于保存累加结果, 初始值为 0 */
    int n;                  /* 用来存储用户输入的最大加数 */
    printf("input a number:\n");
    scanf("%d", &n);        /* 获取最大加数, 保存到变量 n 中 */
    s=add(n);
    printf("1 到 %d 的和为: %d\n", n, s);
}
```

5. 运行结果

```
input a number:
100
1到100的和为:5050
```

6. 算法优化

1) 优化说明

利用数学知识进行算法优化。通过对自然数累加的分析可以看出, 每次相加的数比上一次的数大 1, 即各项加数形成一个差为 1 的等差数列。根据等差数列求和公式: (首

项+末项)×项数/2,可得自然数的累加和。

2) 算法说明

算法说明参见表 3-2。

表 3-2

类 型	名 称	代表的含义
算法	add(int n)	求自然数的累加和
形参变量	n	自然数,进行累加的最大数
变量	s	累加器(和),存放累加结果

3) 算法设计

```
#include "stdio.h"
int add(int n)
{
    int s;
    s=(1+n)*n/2;          /* 利用数学公式(首项+末项)×项数/2 来替代 for 循环 */
    return s;
}
```

3.2.2 自然数倒数求和

1. 问题描述

求 $1/1+1/2+1/3+\cdots+1/n$,要求累加项精确到 10^{-6} ,并输出结果。

2. 问题分析

求解本问题可使用 do-while 循环,循环体中实现累加求和。加数 t 为 $1/n$, n 的值从 1 开始,每次循环 n 的值增加 1,并在每次求和后判定循环条件。根据题意,要求精度到 10^{-6} ,则循环终止条件为 $1/n<10^{-6}$ 。

3. 算法说明

算法说明参见表 3-3。

表 3-3

类 型	名 称	代表的含义
算法	fun()	求自然数倒数之和
变量	s	累加器(和),存放累加结果
变量	t	累加项
变量	n	循环控制变量

4. 算法设计

```
#include "stdio.h"
double fun()
```

```
{
    double s=0,t;
    int n=1;
    do
    {
        t=1.0/n;
        s=s+t;
        n++;
    }while(t>=0.000001);           /* t 的值大于精确度时执行循环体 */
    return s;
}
void main()
{
    double s=0;
    s=fun();
    printf(" 1/1+1/2+1/3+...+1/n=%f\n",s);
}
```

5. 运行结果

```
1/1+1/2+1/3+...+1/n=14.392728
```

6. 算法优化

优化说明：

- (1) 题中要求累加项精确到 10^{-6} ，可以直接将它表示成小数形式 0.000 001，但其中 0 比较多时，容易出错；这时建议用指数形式表示为 $1e-6$ 。
- (2) 算法中“ $t=1.0/n$ ；”不能写成“ $t=1/n$ ；”，这是因为 C 语言有整除和实除之分。
- (3) 算法中“ $t=1.0/n$ ； $n++$ ；”可简写为“ $t=1.0/(n++)$ ；”，在 C 语言中，“ $++i$ ；”是使用 i 之前，先使 i 的值加 1。“ $i++$ ；”在使用 i 之后，对 i 的值加 1，因此等价。

3.2.3 统计及格人数

1. 问题描述

一次考试共考了语文、代数和外语三科。某小组共有 9 人，考后各科及格人数名单如表 3-4 所示，请编写算法找出三科全及格的学生的学号，并统计全部及格的人数。

表 3-4

科目	及格学生的学号	科目	及格学生的学号
语文	1,9,6,8,4,3,7	外语	8,1,6,7,3,5,4,9
代数	5,2,9,1,3,7		

2. 问题分析

从语文名单中逐一抽出及格学生学号，先在代数名单中查找，若有该学号，说明代数

也及格了,再在外语名单中继续查找,看该学号学生是否外语也及格了,若仍在,说明该学号学生三科全及格,否则至少有一科不及格。语文名单中没有的学号,不可能三科全及格,因此,语文名单处理完算法就可以结束了。

3. 算法说明

算法说明参见表 3-5。

表 3-5

类 型	名 称	代表的含义
算法	check()	判断三门课程是否都及格
数组	a,b,c	分别存储语文、代数、外语及格的学号
变量	count	统计三门都及格的人数
变量	flag	标志变量

4. 算法设计

```
#include "stdio.h"
int  a[7],b[6],c[8],count=0;
int check()
{
    int i,j,k,temp;
    for(i=0;i<7;i++)
    {
        temp=a[i];
        for(j=0;j<6;j++)
            if(temp==b[j])
                for(k=0;k<8;k++)
                    if(temp==c[k])
                    {
                        count++;
                        printf("%-4d",temp);
                        break;
                    }
    }
    printf("\n");
    return count;
}
int main()
{
    int i;
    printf("请输入语文及格的学号:");
    for(i=0;i<7;i++)
        scanf("%d",&a[i]);
```

```
printf("请输入代数及格的学号:");
for(i=0;i<6;i++)
    scanf("%d",&b[i]);
printf("请输入外语及格的学号");
for(i=0;i<8;i++)
    scanf("%d",&c[i]);
check();
if(count==0)
    printf("no.\n");
else
    printf("全部及格的人数为:%d\n",count);
}
```

5. 运行结果

```
请输入语文及格的学号:1 9 6 8 4 3 7
请输入代数及格的学号:5 2 9 1 3 7
请输入外语及格的学号:8 1 6 7 3 5 4 9
1 9 3 7
全部及格的人数为:4人
```

6. 算法优化

1) 优化说明

本题统计三科及格学生名单,因为有 9 名学生,可以开辟 9 个元素的数组 $a[]$,作为各学号考生及格科目的计数器。将三科及格名单通过键盘输入,无须用数组存储,只要同时用数组 a 累加对应学号的及格科目个数即可。最后,凡计数器的值等于 3,就是全及格的学生,否则,至少有一科不及格。

2) 算法说明

算法说明参见表 3-6。

表 3-6

类 型	名 称	代表的含义
算法	check()	判断三门课程是否都及格
数组	a[10]	下标为学号,元素值为该考生及格科目的计数器
变量	count	统计三门都及格的人数
变量	flag	标志变量

3) 算法设计

```
#include "stdio.h"
int count=0,xh, a[10]={ 0 };
int check()
{
    for(xh=1;xh<=9;xh++)
```

```
        if(a[xh]==3)
        {
            count++;
            printf("%-4d",xh);
        }
        printf("\n");
        return count;
    }
int main()
{
    int i;
    printf("请依次输入及格的学号:");
    for(i=1;i<=21;i++)
    {
        scanf("%d",&xh);
        a[xh]++;
    }
    check();
    if(count==0)
        printf("no.\n");
    else
        printf("及格人数为:%d人\n",count);
}
```

3.2.4 计算 π 值

1. 问题描述

利用公式 $\pi/4=1-1/3+1/5-1/7+\cdots$ 计算 π 的近似值,要求项的绝对值小于 10^{-6} ,输出结果保留 6 位小数。

2. 问题分析

根据已知公式,关键是求出多项式的值。经过观察发现多项式的各项是有规律的:一是各项符号先正后负依次交替;二是每项的分子都是 1;三是后一项的分母是前一项分母加 2。

找到这些规律就可以用累加法设计算法了。首先设置一个符号变量 sign,用来标记各项的符号,使用“sign=-sign;”语句进行符号交替变换;其次使用一个循环变量 i 表示分母,i 的初值为 1,其变化规律是语句“i=i+2;”,因此通项即可表示为 $t=\text{sign} * 1.0/i$ 。再次设累加器变量 pi,用来表示 π 值,计算方法是语句“pi=pi+t;”,最后循环的终止条件是“fabs(t)<1e-6”,最后求得 π 值为 pi*=4。

注意: fabs(x)是求实数 x 绝对值的库函数,使用时要包含“math.h”头文件。

3. 算法说明

算法说明参见表 3-7。

表 3-7

类 型	名 称	代表的含义
算法	fun()	求 π 的近似值
变量	pi	累加器,保存 π 值
变量	i	循环变量,保存每项的分母
变量	sign	标记符号,用来记录多项式的符号
变量	t	表示多项式的项

4. 算法设计

```
#include "stdio.h"
#include "math.h"
double fun()
{
    long i;
    int sign=1;
    double pi=0,t=1;
    i=1;
    do
    {
        pi=pi+t;
        i+=2;
        sign=-sign;
        t=sign*1.0/i;
    }while(fabs(t)>=1e-6);
    pi*=4;
    return pi;
}
void main()
{
    double pi;
    pi=fun();
    printf("π=%.6f\n",pi);
}
```

5. 运行结果

π=3.141591

3.2.5 数位求和

1. 问题描述

给定一个 int 类型的整数 n ,求这个整数的各位数之和。例如: 123 表示计算 $1+2+3$ 的值,这个整数的各位数之和就是 6。

2. 问题分析

要想得到整数的各位数,可以利用数组将整数的各位数存入数组中,最后再遍历数组求和。

接下来的关键是如何获得这个整数的各位数。

假如这个整数是 $10^2 \times a + 10 \times b + c$, 它的各位数就是 a, b, c , 我们只需要将 $10^2 \times a + 10 \times b + c$ 模除 10 就可以得到 c , 并且 $10^2 \times a + 10 \times b + c$ 变为新的整数 $10 \times a + b$, 我们再将 $10 \times a + b$ 模除 10 就可以得到 b , 并且 $10 \times a + b$ 变为新的整数 a , 即得到整数 $10^2 \times a + 10 \times b + c$ 的各位数 a, b, c 。

3. 算法说明

算法说明参见表 3-8。

表 3-8

类 型	名 称	代表的含义
算法	getNum(int m)	求数位之和
形参变量	m	输入的 int 类型的数
变量	len	int 类型数的长度
变量	sum	各位数之和
一维数组	num	存储 int 类型数的各位

4. 算法设计

```
int getNum(int m)
{
    int len=0, num[20], i, sum=0;
    while (m)
    {
        num[len]=m%10;           /* 存储 m 的最后一位 */
        len++;
        m/=10;                   /* 除 10, 最后一位进行更新 */
    }
    for (i=0; i<len; i++)
    {
        sum+=num[i];             /* 利用 sum 进行累加求和 */
    }
    return sum;
}

int main()
{
    int n;
    scanf("%d", &n);
    printf("%d", getNum(n));
}
```

```
        return 0;
    }
}
```

5. 运行结果

```
123456
21
```

6. 算法优化

1) 优化说明

根据题意,只需要得到 int 类型的数的各位数之和就可以,前一个算法利用数组进行求和,结果虽然正确,但是算法效率低。在此可以不设置数组存储整数的各位数,直接利用 sum 对取出来的数字进行累加求和。

2) 算法说明

算法说明参见表 3-9。

表 3-9

类 型	名 称	代表的含义
算法	getNum(int m)	求数位之和
形参变量	m	输入的 int 类型的数
变量	n	输入的 int 类型的数
变量	sum	各位数之和

3) 算法设计

```
#include "stdio.h"
int getNum(int m)
{
    int sum=0;
    while (m)
    {
        sum +=m %10;          /* 对 sum 进行累加求和 */
        m /=10;
    }
    return sum;
}
int main()
{
    int n;
    scanf("%d", &n);
    printf("%d", getNum(n));
}
```

```
        return 0;
    }
}
```

3.2.6 小鱼游泳问题

1. 问题描述

有一条小鱼,它平日每天游泳 250 千米,周末休息(实行双休日),假设从周 n 开始算起,过了 k 天以后,小鱼一共累计游泳了多少千米呢?

2. 问题分析

本题表明除了周末每天游泳 250 千米,那么问题的关键在于判断周末,假设小鱼从周 n 开始,一共游泳 k 天,用 i 控制循环次数,每游一天 n 就加上 1,累加器 s 加上 250,当 $n=6$ 或者 $n=7$,就证明是周末,只需改变 n 。

3. 算法说明

算法说明参见表 3-10。

表 3-10

类 型	名 称	代表的含义
算法	fun(int n,int k)	求解小鱼游泳问题
形参变量	n	周几开始游
形参变量	k	游几天
变量	s	累加器(和),存放累加结果

4. 算法设计

```
#include<stdio.h>
int fun(int n,int k)
{
    int s=0;                                /* 游了 s 千米 */
    for (int i=1; i <=k; i++)                /* 要游 k 天 */
    {
        if (n != 6 && n != 7)                /* 如果不是周末,加 250 */
            s +=250;
        if (n ==7)
            n =1;                            /* 如果是周日,n 赋值 1 */
        else
            n++;                             /* 否则 n++ */
    }
    return s;
}
int main()
{

```

```

int n, k;                                /* 周 n 开始游,过了 k 天 */
scanf("%d%d", &n, &k);
printf("游了%d千米", fun(n,k));          /* 输出游的千米数 */
return 0;
}

```

5. 运行结果

```

3 10
游了2000千米

```

6. 算法优化

1) 优化说明

不难想到,当游到周六(即 $n=6$)和周日(即 $n=7$)时,小鱼游的千米数(累加器 s)没有变化,因此可以通过适当的改变,跳过周日的情况。

2) 算法说明

算法说明同上。

3) 算法设计

```

#include<stdio.h>
int fun(int n,int k)
{
    int s=0;                                /* 游了 s 千米 */
    for (int i=1; i<=k; i++)                /* 要游 k 天 */
    {
        if (n!=6 && n!=7)                  /* 如果不是周末,加 250 */
            s+=250;
        if (n==6)                          /* 如果是周六 */
        {
            n=1;                            /* 跳过周日到周一 */
            i++;                             /* 在 for 循环 i++的基础上再 i++,跳过周日 */
        }

        else
            n++;                             /* 否则 n++ */
    }
    return s;
}

int main()
{
    int n, k;                                /* 周 n 开始游,过了 k 天 */
    scanf("%d%d", &n, &k);
}

```

```
printf("游了%d千米", fun(n,k));      /* 输出游的千米数 */
return 0;

}
```

3.2.7 判断天数

1. 问题描述

输入一个年月日,格式如 2013/5/15,判断这一天是这一年的第几天。

2. 问题分析

首先对输入的年份进行判断,是闰年还是平年,闰年二月是 29 天,平年二月是 28 天。其次对输入的月份进行判断,假如输入的是 5 月,则把 1、2、3、4 四个月份的天数加在一起,最后再加上日(几号)的值,则可得到这个日期是该年的第几天。

3. 算法说明

算法说明参见表 3-11。

表 3-11

类 型	名 称	代表的含义
算法	day(int y,int m,int d)	判断天数
形参变量	y,m,d	年、月、日
变量	sum	统计天数

4. 算法设计

```
#include "stdio.h"
int day(int y,int m,int d)
{
    int sum=0;
    switch(m-1)
    {
        case 11:sum+=30;
        case 10:sum+=31;
        case 9:sum+=30;
        case 8:sum+=31;
        case 7:sum+=31;
        case 6:sum+=30;
        case 5:sum+=31;
        case 4:sum+=30;
        case 3:sum+=31;
        case 2:sum+=28;
        case 1:sum+=31;
    }
}
```

```
        if(m>2)
            if((y%4==0&& y%100!=0)|| (y%400==0))
                sum++;
            sum+=d;
            return(sum);
    }
    main()
    {
        int y,m,d;
        printf(" Input date:(eg. 2013/5/5): ");
        scanf("%d/%d/%d",&y,&m,&d);
        printf(" day=%d\n",day(y,m,d));
    }
```

5. 运行结果

```
Input date:(eg. 2013/5/5): 2013/5/5
day=125
```

6. 算法优化

1) 优化说明

首先将每个月份的天数保存到数组 *a* 中。对输入的月份进行判断,假如输入的是 5 月,则把 1、2、3、4 四个月份的天数加在一起,然后对输入的年份进行判断,若闰年二月是 29 天,则天数增加 1。最后再加上日(几号)的值,就可得到这个日期是该年的第几天。

2) 算法说明

算法说明参见表 3-12。

表 3-12

类 型	名 称	代表的含义
算法	day(int y,int m,int d)	判断天数
形参变量	y,m,d	年、月、日
一维数组	a	存储每月的天数
变量	sum	统计天数

3) 算法设计

```
#include "stdio.h"
int day(int y,int m,int d)
{
    int a[12]={31,28,31,30,31,30,31,31,30,31,30,31};
    int i,sum=0;
    for(i=0;i<m-1;i++)
```

```
        sum=sum+a[i];
    if(m>2)
        if((y%4==0&& y%100!=0)|| (y%400==0))
            sum++;
    sum=sum+d;
    return(sum);
}
```

3.3 小 结

本章讲解了算法设计方法中的累加法,并结合一些具体的问题来分析和设计算法。重点要理解累加法的设计思想,并能够运用它解决实际问题。

累加法的基本思想是:首先设置累加器 S , 根据情况将初值赋 0 或一个特定值;其次定义一个变量 T 存放累加项,针对具体问题找出 T 的变化规律;最后,在循环体中执行 $S=S+T$,直到满足循环结束条件为止。

如果循环次数确定,那么累加法设计的一般过程如下(假设累加次数为 n 次):

```
/* 其他语句 */
S=0;
for(i=1; i<=n; i++)
{
    计算累加项 T 的值
    S=S+T;
}
```

如果循环次数不确定,则累加法设计的一般过程如下:

```
/* 其他语句 */
S=0;
do
{
    计算累加项 T 的值
    S=S+T;
}
while(循环结束条件);
```

习 题

- 3-1 求 $1-2+3-4+5-6+\cdots+99-100$ 。
- 3-2 求 $1-1/2+1/3-1/4+\cdots-1/100$ 。
- 3-3 求 100 以内所有素数的和。

- 3-4 求 100 以内所有奇数的和。即 $1+3+5+\cdots+99$ 。
- 3-5 编程计算 $(1+2)+(2+3)+(3+4)+\cdots+(20+21)+(21+22)$ 的值。
- 3-6 编写一个程序,计算半径分别为 0.5mm、1.5mm、2.5mm、3.5mm、4.5mm、5.5mm 时圆的面积。
- 3-7 求数列 9,99,999...前 n 项的和。
- 3-8 计算 $(1+2)+(2+3)+(3+4)+\cdots+(n+(n+1))$ 的值。
- 3-9 输入一个数 n ,求 $1+2+3+\cdots+n+(n-1)+(n-2)+\cdots+3+2+1$ 。例如:输入 5 时,要求输出 $1+2+3+4+5+4+3+2+1$ 的值。
- 3-10 编程计算 $1!+2!+3!+\cdots+10!$ 的值。
- 3-11 编写程序,求下面数列的表达式前 40 项的和(结果取 4 位小数): $1, (1/2)^4, (1/3)^4, \cdots, (1/n)^4$ (其中, ^ 表示幂运算)。
- 3-12 求下面图形中直到第几行为止,所有的 * 数目和为 5 151。

```
      *
    *  *
  *   *   *
*  *  *  *  *
:

```

- 3-13 编程计算 $a+aa+aaa+\cdots+aa\cdots a$ (n 个 a) 的值, n 和 a 的值由键盘输入。
- 3-14 求 1 000 以内所有的完全数的和(完全数是指一个数除其本身外的因子之和等于该数。例如, $28=1+2+4+7+14$, 因此 28 为完全数)。
- 3-15 求 100 以内所有同时能被 3 和 5 整除的数的和。
- 3-16 小猴摘桃子,第一天摘一个,以后每天摘的桃子数均是前一天的 2 倍多一个,求第 10 天小猴子一共摘了多少个桃子。
- 3-17 找出 100 到 500 以内所有同时能被 3、5、7 整除的正整数,并用 N 记录有多少个。
- 3-18 计算 $S=1+2+3+4+\cdots+n$ 在累加的过程中,求当 S 的值首次大于 3 000 时的 n 值是多少。
- 3-19 求数列 1,10,100,1 000,...前 n 项的和, n 由键盘输入。
- 3-20 已知一个数列为 1,2,4,7,11,16,22,...,则数列的第 n 项为多少? (提示: $a_2-a_1=1$, $a_3-a_2=2$, $a_4-a_3=3$,...)。
- 3-21 输入 n 个百分制成绩,计算并输出平均成绩。要求输出结果精确到两位小数。
- 3-22 输入若干非 0 实数,以 0 为终止条件,统计其中正数的个数、负数的个数。
- 3-23 输入一行字符,统计其中的英文字母个数。提示:输入到字符 '\n' 时停止输入。
- 3-24 要求用户输入一个大于 1 的数,然后计算从 1 到这个数的累加和。
- 3-25 求 4 到 200 之间(包括 4 和 200)偶数的累加和,并求这些偶数的平均值。
- 3-26 请编写函数 fun,函数的功能是:根据以下公式计算 s ,计算结果作为函数值返回, n 通过形参传入。 $s=1+1/(1+2)+1/(1+2+3)+\cdots+1/(1+2+3+\cdots+n)$ 。
- 3-27 把 10 个整数存入一维数组中,求这 10 个整数的和、最大值、最小值。
- 3-28 键盘输入一行字符(以回车符表示结束),将其中每个数字字符所代表的数值累加

起来,输出结果。如输入 abc235,答案输出为 10。

- 3-29 求 e 的值,根据输入的 n 值,求前 n 项之和。 $e = 1 + 1/1! + 1/2! + 1/3! + \cdots + 1/n!$ 。
- 3-30 在唱歌等大奖赛评分时,一般要有若干名评委,记分规则是:去掉一个最高分和一个最低分,再算平均分。设按百分制记分,请设计一个算分的程序(提示:算法的基本思路为①输入评委人数 N ;②逐一输入每个评委的打分,同时累加求和 sum ,并记录下最高分 max 和最低分 min)。