

11 一维数组



巴蜀中學
BASHU SECONDARY SCHOOL

11 一维数组

主教练：党东



【数据量的大小问题】



巴蜀中學
BASHU SECONDARY SCHOOL

问题1： 输入3名学生的身高，先输出平均身高，并将3名学生的身高倒序输出。

【🙋我能做啊！】 定义3个变量，分别储存3名学生的身高，输出平均身高后，将3个变量依次倒序输出即可。

```
int a1,a2,a3;
```

```
cin>>a1>>a2>>a3;
```

```
cout<<(a1+a2+a3)/3.0<<a3<<a2<<a1;
```

问题2：输入30个学生的身高，先输出平均身高，并将30个学生的身高倒序输出。

【🙋我能做吧！】 在解决这个问题时，虽然我们仍然可以通过一个变量来累加读入的30个身高求出学生的总身高，进而求出平均身高。但因为只有读入最后一个学生的身高后才能求得平均身高，并且由于题目要求将30个学生的身高倒序输出，故必须把30个学生的身高都保留起来，然后逐个将这30个变量倒序输出。如果，用简单变量a1, a2,..., a30存储这些数据，要用30个变量保存输入的数据，程序片断如下：

.....

```
cin>>a1>>a2>>...>>a15>>...>>a30;
```


【数据量的大小问题】



巴蜀中學
BASHU SECONDARY SCHOOL

问题3： 输入300个学生的身高，先输出平均身高，并将300个学生的身高倒序输出。

【我能做吧？】

【数据量的大小问题】



巴蜀中學
BASHU SECONDARY SCHOOL

问题5： 输入30000个学生的身高，先输出平均身高，并将30000个学生的身高倒序输出。

【🙋我能做吗？】

【数据量的大小问题】



巴蜀中學
BASHU SECONDARY SCHOOL

问题n：输入300...0个学生的身高，先输出平均身高，并将300...0个学生的身高倒序输出。

【😞我做不出来】随着数据规模的逐渐增大，暂且不论具体计算过程，单单读入过程都会异常复杂，需要编写大量繁琐的输入语句。如果真正要像上面这样编写程序，电脑的优势根本就没有得到体现。

从以上的问题可以看出，如果只使用简单变量处理大量数据，就必须使用大量只能单独处理的变量，即使是**简单问题也需要编写冗长的程序。**

【数据量的大小问题】



巴蜀中學
BASHU SECONDARY SCHOOL



针对大量数据的利器 - - 数组



【针对大量数据的利器】



巴蜀中學
BASHU SECONDARY SCHOOL

从上一个问题中我们可知，储存 n 个学生(a_1, a_2, \dots, a_n)身高的变量，它们的作用和数据类型都是相同的。那么，我们能不能我们把它们集中到一起定义、储存和操作呢？

C++为我们提供了“**数组**”这一工具。**数组就是一组相同类型的变量**，他们都是为了表示同一批对象的统一属性，如一个班级所有同学的身高、全球所有国家的人口数等。

数组的定义



- 数组是指具有同一数据类型一类变量，它们**在内存中依次顺序存放**。整个数组共用一个名字，而其中的每一项又称为一个元素。
- 好比我们有50名学生，分别定义为： g_0, g_1, \dots, g_{49} ；如果用数组则可以定义为 $g[50]$ ，编号从 $g[0]$ — $g[49]$ 。



一、一维数组的定义和引用



一维数组的定义方式:

类型说明符 **数组名[常量表达式];**

- 其中，类型说明符是任意一种基本数据类型；数组名是用户定义的数组标识符；方括号中的常量表达式表示数据元素的个数，也称为数组的大小。
- 例如：
`int a[10];` //说明整型数组a，有10个元素。
`double b[10];` //说明实型数组b，有10个元素；
`char ch[20];` //说明字符数组ch，有20个元素。

对于数组类型说明应注意以下几点：

- (1)数组的类型说明符实际上是指数组元素的取值类型，对于同一个数组，其所有元素的数据类型都是相同的
- (2)数组名的书写规则应符合标识符的书写规定
- (3)数组名不能与其它变量名相同，例如：
 int **a**;
 double **a**[10]; 是错误的
- (4)方括号中常量表达式表示数组元素的个数，如a[5]表示数组a有5个元素。但是其下标从0开始计算。因此5个元素分别为a[0],a[1],a[2],a[3],a[4]。

对于数组类型说明应注意以下几点:

- (5)不能在方括号中用**变量**来表示元素的个数，但是可以是符号常数或常量表达式。例如：

`const int maxn=1000;`

`int a[3+2],b[maxn+1];`是合法的。

但是下述说明方式是错误的：`int a[n];`

- (6)允许在同一个类型说明中，说明多个数组和多个变量。

例如：`int a,b,c,d,k1[10],k2[20];`

数组的存放



必须是常数

`int a[4];` // 表明a数组由4个int型元素组成

其元素分别为: `a[0]`, `a[1]`, `a[2]`, `a[3]`

其序号从0开始。 若存放首地址为2000H, 则在内存中顺序存放的:

2000H	2004H	2008H	200CH	2010H
a[0]	a[1]	a[2]	a[3]	

二、一维数组的引用



- **每个数组元素都是一个变量**，数组元素可以表示为：

数组名[下标]

如数组 $a[10]$ ，用 $a[2]$ 可以调用第3个数组元素

- 其中，下标可以是任何值为整型的表达式，该表达式里可以包含变量和函数调用。引用时，下标值应在数组定义的下标值范围内。
- 例如：若 i 、 j 都是 int 型变量，则 $a[5]$ ， $a[i+j]$ ， $a[i++]$ 都是合法的元素。
- 数组的**精妙**在于下标可以是变量，通过对下标变量值的灵活控制，达到灵活处理数组元素的目的。

三、一维数组的基本操作



(1) **赋值**——可以边定义边赋值，也可以在中间赋值
对数组中的一部分元素列举初值，**未赋值的部分是0。**

```
int a[10]= {0,1, 2, 3, 4, 5};
```

即是 `int a[10]= {0,1, 2, 3, 4, 5, 0, 0 , 0 , 0};`

```
int a[10]= {0,1,2,.....,9};
```

这样赋值是错误的。因为不能给数组整体赋值，只能一个一个地赋值。

■ **可以给数组全部清0，有以下三种表示：**

```
int a[105]={0};
```

```
memset(a,0,sizeof(a));
```

```
for(i=0;i<105;i++)a[i]=0;
```

三、一维数组的基本操作



(2)一维数组的读入： `for(i=1;i<=100;i++)cin>>a[i];`

(3)一维数组的输出： `for(i=1;i<=100;i++)cout<<a[i];`

【思考】从键盘输入 $n(n \leq 100)$ 个整数，然后反向输出 --1082

【样例输入】

4

1 2 3 4

【样例输出】

4 3 2 1

```
int a[105],i,n;  
cin>>n;  
for(i=1;i<=n;i++)cin>>a[i];  
for(i=n;i>=1;i--)cout<<a[i]<<" ";
```


【例1】 Fibonacci数列 --1062



巴蜀中學
BASHU SECONDARY SCHOOL

【问题描述】 已知faibonacci数列的前几个数分别为0, 1, 1, 2, 3, 5,编程求出此数列的第n项。 (n由键盘输入)
 $n \leq 60$

【输入格式】 输入仅一个整数n ($n \leq 60$)

【输出格式】 输出faibonacci数列中的第n个数

【样例输入】 5

【样例输出】 3

【例1】 Fibonacci数列 --1062



题目分析

$f[1]$	$f[2]$	$f[3]$	$f[4]$	$f[5]$	$f[6]$	$f[7]$	$f[8]$	$f[9]$
0	1	1	2	3	5	8	13	21	

■ 设 $f[i]$ 表示第 i 项的值, 则: $f[i]=f[i-1]+f[i-2]$

```
long long n,i,f[65]={0};  
f[1]=0;f[2]=1;  
cin>>n;  
for(i=3;i<=n;i++) $f[i]=f[i-1]+f[i-2]$ ;  
cout<<f[n]<<endl;
```

注意事项

- ①比较大的数组应尽量定义为全局变量;
- ②数组之间赋值问题
 - `int a[maxn],b[maxn];`
 - 数组之间不能**`b=a`**这样赋值;
 - 若要将数组a全部复制到数组b中, 可以写成:

`memcpy(b,a,sizeof(a))`

或用循环**`for(i=0;i<maxn;i++)b[i]=a[i];`**

【例2】兔子繁殖 --1063



Description

有一对小兔,过一个月之后长成大兔,到第四个月就可以生下一对小兔,并且以后每个月都生下一对小兔。而所生的一对小兔也同样到一个月之后长成大兔,到第四个月就可以生下一对小兔,并且以后也每个月都生下一对小兔.假设所有的兔子均不死亡,问第n个月后共有多少对兔子? 请设计一个程序, 解决这一问题。

Input:一个整数n ($n \leq 50$)

Output:第n个月后共有多少对兔子

Sample Input:5

Sample Output:3

【例2】兔子繁殖 --1063



巴蜀中學
BASHU SECONDARY SCHOOL

参考代码

```
#include<iostream>
using namespace std;
int main()
{   int a[55],i,n;
    a[1]=1;a[2]=1;a[3]=1;
    cin>>n;
    for(i=4;i<=n;i++)a[i]=a[i-1]+a[i-3];
    cout<<a[n]<<endl;
    return 0;
}
```

四、活用数组下标



- 数组的**精妙**在于下标可以是变量，通过对下标变量值的灵活控制，达到灵活处理数组元素的目的。
- 在例1中通过循环控制数组下标值从1变化到n，将读入的n个数逐个存入 $a[1] \sim a[n]$ 中，反向输出只要控制数组下标值从n变化到1，逐个输出即达到目标。

