

小马加编信息学教案(十)

C++数组(一)

- 一. 课程内容
- 二. 知识讲解
 - 1. 一维数组的定义
 - * 1.1 数组的含义
 - * 1.2 一维数组的定义
 - 2. 一维数组的使用
 - * 2.1 一维数组的元素引用
 - * 2.2 一维数组的存储结构
 - * 2.3 一维数组的初始化
 - 3. 一维数组的输入与输出
 - * 3.1 一维数组的下标输入输出法
 - * 3.2 给一维数组整体赋值的方法
- 三. 经典例题
- 四. 提高巩固

一. 课程内容

1. 一维数组的定义
2. 一维数组的使用
3. 一维数组的输入输出

二. 知识讲解

1. 一维数组的定义

1.1 数组的含义

数组就是一组相同类型的变量，它们往往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。

数组可以是一维的，也可以是二维或多维的。

1.2 一维数组的定义

定义一维数组的格式如下：

类型标识符 数组名 [常量表达式]；

其中，类型标识符可以是任何基本数据类型，也可以是结构体等构造类型，相同类型的数组可以一起定义。

数组名必须是合法的标识符。常量表达式的值即为数组元素的个数

ex:

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int a[1000];
    double b[10000];
    char c[1000000];
}
```

2. 一维数组的使用

2.1 一维数组的元素引用

数组定义好后，就可以引用（调用）其中的任意一个元素。

引用格式为：

数组名[下标]

如：h[5]、h[i*2+1] 等。

数组中的每个元素都有一个序号，这个序号从0开始，而不是从我们熟悉的1开始，称为下标（*Index*）。

其中，下标只能为整型常量或整型表达式，值必须在数组定义的下标范围内，否则会出现“下标越界错误”。

需要注意的是，不能一次引用整个数组，只能逐个引用数组的单个元素。

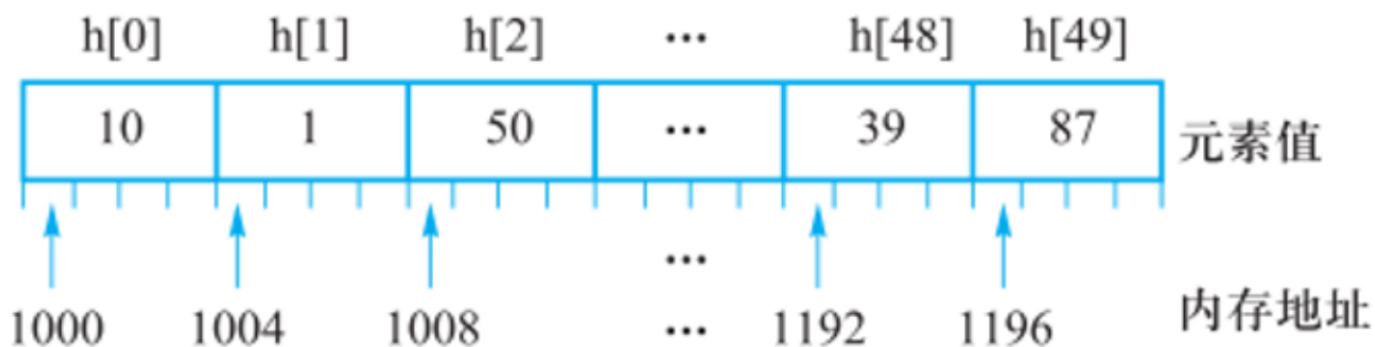
- 注意，c++ 中数组的下标是从0开始的！这一点非常重要
- 即访问数组元素时，下标的取值范围为 $0 \leq index < length$ ，过大或过小都会越界，导致数组溢出，发生不可预测的情况，

所以 a[100] 这个数组中的元素下标范围是 a[0 ~ 99] ,没有 a[100] !

2.2 一维数组的存储结构

数组在计算机内存单元中是连续存储的。

程序一旦执行到数组的定义语句，就会开辟出若干字节的内存单元。



2.3 一维数组的初始化

我们可以在定义数组的时候给数组赋值，称为数组的初始化。例如：

```
int a[4] = {20, 345, 700, 22};
```

数组元素的值由{ }包围，各个值之间以,分隔。

- 注意：

可以只给部分元素赋值。当{ }中值的个数少于元素个数时，只给前面部分元素赋值。例如如下代码表示只给 `a[0] ~ a[4]` 5个元素赋值，而后面 5 个元素自动初始化为0。

```
int a[10]={12, 19, 22 , 993, 344};
```

只能给元素逐个赋值，不能给数组整体赋值。

如给全部元素赋值，那么在定义数组时可以不给出数组长度。

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

等价于

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

3. 一维数组的输入与输出

3.1 一维数组的下标输入输出法

- 一维数组的输入、输出等操作，都是采用循环语句结合下标变化，逐个元素进行。

ex:

```
#include <bits/stdc++.h>
int main() {
    int nums[10];

    //将1~10放入数组中
    for(int i = 0; i < 10; i++){
        nums[i] = (i + 1);
    }

    //依次输出数组元素
    for(i = 0; i < 10; i++) {
        cout << nums[i] << ' ';
    }

    return 0;
}
```

运行结果

```
1 2 3 4 5 6 7 8 9 10
```

代码分析

变量 i 既是数组下标，也是循环条件；

将数组下标作为循环条件，达到最后一个元素时就结束循环。

数组 `nums` 的最大下标是 9，也就是不能超过 10，所以我们规定循环的条件是 $i < 10$ ，一旦 i 达到 10 就得结束循环。

3.2 给一维数组整体赋值的方法

- `memset` 函数

`memset` 函数是给数组“按字节”进行赋值，一般用在 `char` 型数组中。如果是 `int` 类型的数组，一般赋值为 0 和 -1 。

注意，一般情况下我们只能用 `memset` 来将数组初始化为 0 或 -1

使用方法：

```
int a[10005];

memset(a, 0, sizeof(a));
memset(a, -1, sizeof(a));
```

三. 经典例题

1. 反序输出数组

输入一个整数数组，反序输出这个数组

输入格式

第一行输入一个正整数 n ,接下来一行是 n 个正整数，以空格分开

输出格式

输出一行，将该数组反序输出

样例输入	样例输出
3 8 4 3	3 4 8

2. 陶陶摘苹果(*NOIP*2005普及组第一题)

陶陶家的院子里有一棵苹果树，每到秋天树上就会结出 10 个苹果。

苹果成熟的时候，陶陶就会跑去摘苹果。陶陶有一张 30 厘米高的板凳，当她不能直接用手摘到苹果的时候，就会踩到板凳上再试试。

现在已知 10 个苹果到地面的高度，以及陶陶把手伸直的时候能够达到的最大高度。

请帮陶陶算一下她能够摘到的苹果的数目。假设她碰到苹果，苹果就会掉下来。

输入格式

第一行包含 10 个 100 ~ 200 之间(包括 100 和 200)的整数（以厘米为单位）分别表示 10 个苹果到地面的高度，两个整数之间用一个空格隔开。

第二行只包括一个 100 120 之间（包含 100 和 120）的整数（以厘米为单位），表示陶陶把手伸直的时候能够达到的最大高度。

输出格式

一行一个整数，表示陶陶能够摘到的苹果的数目。

样例输入	样例输出
100 200 150 140 129 134 167 198 200 111 110	5

3. 报名签到

n 位同学（编号从1到 n ）同时来到体育馆报名签到，领取准考证和参赛资料。

为了有序报名，这 n 位同学需要按编号次序（编号为1 的同学站在最前面）从前往后排成一条直线。

然而每一位同学都不喜欢拥挤，对于第 i 位同学，如果有另外一位同学距离他（她）的距离小于 $a[i]$ ，那么就会发生冲突。

小可想知道如果要不发生任何冲突的情况下，这 n 位同学排队的队列最短长度是多少。

输入格式

输入有两行：

第一行一个整数 n ，表示报名签到的同学人数。

第二行有 n 个整数，第 i 个整数 $a[i]$ 表示第 i 个同学必须与其他同学保持的距离。

输出格式

输出一行，包括一个整数，表示这 n 位同学排队队列的最小长度。

注意： n 位同学要按 $1 \sim n$ 的次序从前往后排队。

样例输入	样例输出
3 3 1 2	5

数据范围

对于20%的数据满足： $1 \leq n \leq 20$ ；

对于70%的数据满足： $1 \leq n \leq 10000$ ；

对于100%的数据满足： $1 \leq n \leq 100000$ ， $1 \leq a[i] \leq 100000$ 。

四. 提高巩固

1. 小鱼的数字游戏

小鱼最近被要求参加一个数字游戏，要求它把看到的一串数字记住了然后反着念出来(表示结束的数字 0 就不要念出来了)。

(长度不一定，以0结束，最多不超过100个，数字不超过 $2^{32} - 1$)

这对小鱼的那点记忆力来说实在是太难了，你也不想想小鱼的整个脑袋才多大，其中一部分还是好吃的肉！所以请你帮小鱼编程解决这个问题。

输入格式

一行内输入一串整数，以0结束，以空格间隔。

输出格式

一行内倒着输出这一串整数，以空格间隔。

样例输入	样例输出
8 4 3 0	3 4 8
3 65 23 5 34 1 30 0	30 1 34 5 23 65 3

2. 不高兴的津津(NOIP2004普及组第一题)

津津上初中了。妈妈认为津津应该更加用功学习，所以津津除了上学之外，还要参加妈妈为她报名的各科复习班。

另外每周妈妈还会送她去学习朗诵、舞蹈和钢琴。

但是津津如果一天上课超过八个小时就会不高兴，而且上得越久就会越不高兴。

假设津津不会因为其它事不高兴，并且她的不高兴不会持续到第二天。

请你帮忙检查一下津津下周的日程安排，看看下周她会不会不高兴；如果会的话，哪天最不高兴。

输入格式

输入包括7行数据，分别表示周一到周日的日程安排。

每行包括两个小于10的非负整数，用空格隔开，分别表示津津在学校上课的时间和妈妈安排她上课的时间。

输出格式

一个数字。如果不会不高兴则输出0，如果会则输出最不高兴的是周几（用1, 2, 3, 4, 5, 6, 7分别表示周一，周二，周三，周四，周五，周六，周日）。

如果有两天或两天以上不高兴的程度相当，则输出时间最靠前的一天。

样例输入	样例输出
5 3 6 2 7 2 5 3 5 4 0 4 0 6	3

3. 最长连号

输入 n 个正整数,要求输出最长的连号的长度。（连号指从小到大连续自然数）

输入格式

第一行，一个数 n ；

第二行， n 个正整数，之间用空格隔开。

输出格式

一个数，最长连号的个数。

样例输入	样例输出
10 10 3 5 6 2 3 4 5 6 8 9	5

数据范围

$$1 \leq n \leq 10000$$