给数组赋初值的方法

1、直接初始化：int arr[3]={1,2,3}；  
2、遍历访问初始化：  
for(i = 0;i< 3;i++)  
arr[i]=i;  
3、内存操作函数。  
memset(arr,3,abs);//abs为另一个已知的数组。  
4、字符串[赋值函数](https://www.baidu.com/s?wd=%E8%B5%8B%E5%80%BC%E5%87%BD%E6%95%B0&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)，仅限于char型数组。  
strcpy(arr,abs) ;abs为一字符串或者char型数组。

# fill和memset 数组填充跟数组初始化

2017年04月27日 16:53:30

阅读数：410

一般我们在使用memset来初始化数组，但是昨天在做一道题时，死活出错，原因在于我想把一个int数组初始化为最大整型数，下面是memset函数原型

void \*memset(void \*s, int ch, [size\_t](http://baike.baidu.com/item/size_t) n)

功能是：（直接从百度百科抄过来）

将s所指向的某一块内存中的前n个 [字节](http://baike.baidu.com/item/%E5%AD%97%E8%8A%82)的内容全部设置为ch指定的[ASCII](http://baike.baidu.com/item/ASCII)值， 第一个值为指定的内存地址，块的大小由第三个[参数](http://baike.baidu.com/item/%E5%8F%82%E6%95%B0)指定，

这个函数通常为新申请的内存做初始化工作， 其返回值为指向s的[指针](http://baike.baidu.com/item/%E6%8C%87%E9%92%88)。

注意：这边是对前n个字节进行赋值，而且还是ASCII的形式，那就是说，对于字符数组，可以将其初始化为任意一个字符，那对于其他基础类型的数组呢？

那只能说一声抱歉了，因为memset按字节对内存块进行初始化,所以他无法对int数组初始化除了0和-1以后的其他数

1. #include<iostream>
2. #include<bits/stdc++.h>
3. using namespace std;
4. int main(){
5. int a[20];
6. int b[20];
7. int c[20];
8. int d[20];
9. memset(a,0,sizeof(a)); *//right 初始化为0*
10. memset(b,-1,sizeof(b)); *//right 初始化为-1*
11. memset(c,1,sizeof(c)); *//wrong*
12. memset(d,2,sizeof(d)); *//wrong*
13. for(int i=0;i<10;i++) cout<<a[i]<<" "; cout<<endl;
14. for(int i=0;i<10;i++) cout<<b[i]<<" "; cout<<endl;
15. for(int i=0;i<10;i++) cout<<c[i]<<" "; cout<<endl;
16. for(int i=0;i<10;i++) cout<<d[i]<<" "; cout<<endl;
17. return 0;
18. }

这个例子输出结果为：

所以小伙伴们在使用memset函数要注意这些

接下来介绍 fill函数

函数参数 fill(first,last,val)

first,last 对应数组的范围，val为元素的值

fill 顾名思义 这个函数是用来填充的，例如我们想对一个数组赋值，那么

1. #include<iostream>
2. #include<bits/stdc++.h>
3. using namespace std;
4. int main(){
5. int a[20];
6. int b[20];
7. char c[20];
8. float d[20];
9. fill(a,a+11,1); *//赋值1*
10. fill(b,b+11,2147483647); *//赋值最大整型 2147483647*
11. fill(c,c+11,'\*'); *//赋值 \**
12. fill(d,d+11,0.1); *//赋值 0.1*
13. for(int i=0;i<10;i++) cout<<a[i]<<" "; cout<<endl;
14. for(int i=0;i<8;i++) cout<<b[i]<<" "; cout<<endl;
15. for(int i=0;i<10;i++) cout<<c[i]<<" "; cout<<endl;
16. for(int i=0;i<10;i++) cout<<d[i]<<" "; cout<<endl;
17. return 0;
18. }

运行结果为：

所以当我们想对数组赋值时，使用fill 函数还是比较方便的

[**c++ 批量初始化数组 fill和fill\_n函数的应用**](https://www.cnblogs.com/kimsimple/p/6661320.html)

转自[c++ 如何批量初始化数组 fill和fill\_n函数的应用](http://blog.csdn.net/sunquana/article/details/9153213)

std::fill(a+2,a+10,0x3f3f3f3f);///从下标2到下标10 前闭后开 共8个

std::fill\_n(a+2,10,0x3f3f3f3f);///从下标2 开始 填充10个

memset(a,0x7f7f7f7f,sizeof(a));

一、 fill和fill\_n函数的应用：

    fill函数的作用是：将一个区间的元素都赋予val值。

     函数参数：fill(first,last,val);//first为容器的首迭代器，last为容器的尾迭代器，

替换元素的区间为**[first,last)**,val为将要替换的值。

     eg：  
             vector <int> V;  
             fill(V.begin(),V.end(),val);

二、fill\_n函数的作用是：

给你一个起始点，然后再给你一个数值count和val。把从起始点开始依次赋予 count个元素val的值。

**注意： 不能在没有元素的空容器上调用fill\_n函数。**

三、关于memset（）函数：

  这个 函数是按字节覆盖，批量初始化内存空间

max\_int= 2147483647，也就是：(1<<31)-1

 而0x7f7f7f=2139062143