

memset函数用法

基本用法

```
void *memset(void *str, int c, size_t n)
```

功能：将指针变量 str 所指向的前 n 字节的内存单元用一个“整数” c 替换，注意 c 是 int 型。s 是 void* 型的指针变量，所以它可以为任何类型的数据进行初始化。

memset 一般使用“0”初始化内存单元，而且通常是给数组或结构体进行初始化。一般的变量如 char、int、float、double 等类型的变量直接初始化即可，没有必要用 memset。如果用 memset 的话反而显得麻烦。

赋值原理

初学时，很多同学不理解为什么memset只对初始化 0 和 -1 时有效。这是因为 memset 是按字节赋值，即使给它int类型数，它也是只取该数最低位的一个字节来赋值。

所以 memset 传int型参数来初始化只适用于各个字节都相同的数，这样才能表现出期望的效果。

```
例如： 0x3f3f3f3f = 00111111 00111111 00111111 00111111 = 1,061,109,567
        0xffffffff = 11111111 11111111 11111111 11111111 = -1
        0x00000000 = 00000000 00000000 00000000 00000000 = 0
        0x7f7f7f7f = 01111111 01111111 01111111 01111111 = 2139062143
```

可以使用如下代码进行测试

```
#include<bits/stdc++.h>

using namespace std;

const int LIMIT = 5;
int a[LIMIT];

int main() {
    //以下两个代码是一样的 63就是0011 1111
    memset(a, 0x3f3f3f3f, sizeof(a));
    memset(a, 63, sizeof(a));
    for (int i = 0; i < LIMIT; ++i) {
        cout << a[i] << endl;
    }
    return 0;
}
```

根据上面的规律我们可以测试一下，制造出一个各个字节都相同的数字，然后赋值给数组进行测试。

例如：00000001 00000001 00000001 00000001 = 16843009 = 0x1010101

```
#include<bits/stdc++.h>

using namespace std;

const int LIMIT = 5;
int a[LIMIT];

int main() {
    //以下两种写法结果一样
    memset(a, 16843009, sizeof(a));
    memset(a, 0x1010101, sizeof(a));
    for (int i = 0; i < LIMIT; ++i) {
        cout << a[i] << endl;
    }
    return 0;
}
```

上面的代码运行结果就是16843009

浮点数数组

我们在使用上述方法对double数组赋值的时候，会发现结果并不是我们想要的，那是因为浮点数含有阶码运算所导致的。

经过测试，如果我们想要给数组在初始化时赋值一个最大值，建议使用127，如下所示：

```
#include<bits/stdc++.h>

using namespace std;

const int LIMIT = 5;
double a[LIMIT];

int main() {
    //以下两个代码是一样的 63就是0011 1111
    memset(a, 1.38242e+306, sizeof(a));
    for (int i = 0; i < LIMIT; ++i) {
        cout << a[i] << endl;
    }
    return 0;
}
```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

