
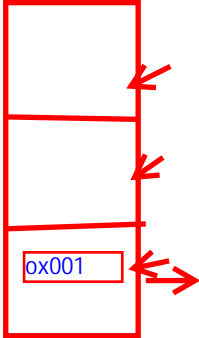


(/apps/redi
utm_sourc
banner-clic

C语言进阶课程笔记之指针

 Jameslong (/u/0da19aebc4e6) [+ 关注](#)
2017.04.25 11:15* 字数 1649 阅读 73 评论 0 喜欢 1
(/u/0da19aebc4e6)



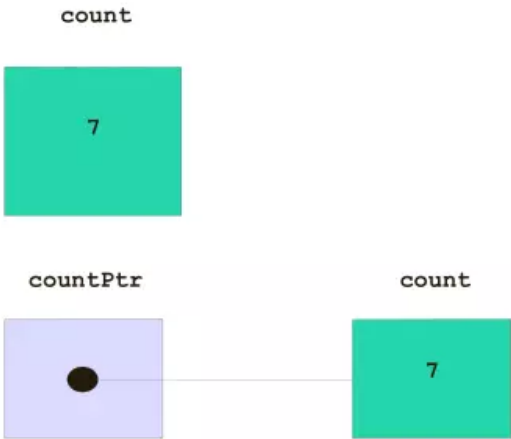
指针

指针就是保存地址的变量，想一下，无论什么类型的指针，因为地址的所占用的空间是一样的，所以指针所占用的字节应该是一样的。后面不妨给出验证。我们常用的写法无外乎向下面这样

```
int i;  
int* p = &i;  
int* p,q;  
int *p,q;
```

指针变量

普通变量的值是实际的值
指针变量的值是具有实际值的变量的地址



Paste_Image.png

我们这里不妨就验证一下指针所占的空间到底是不是一样的。

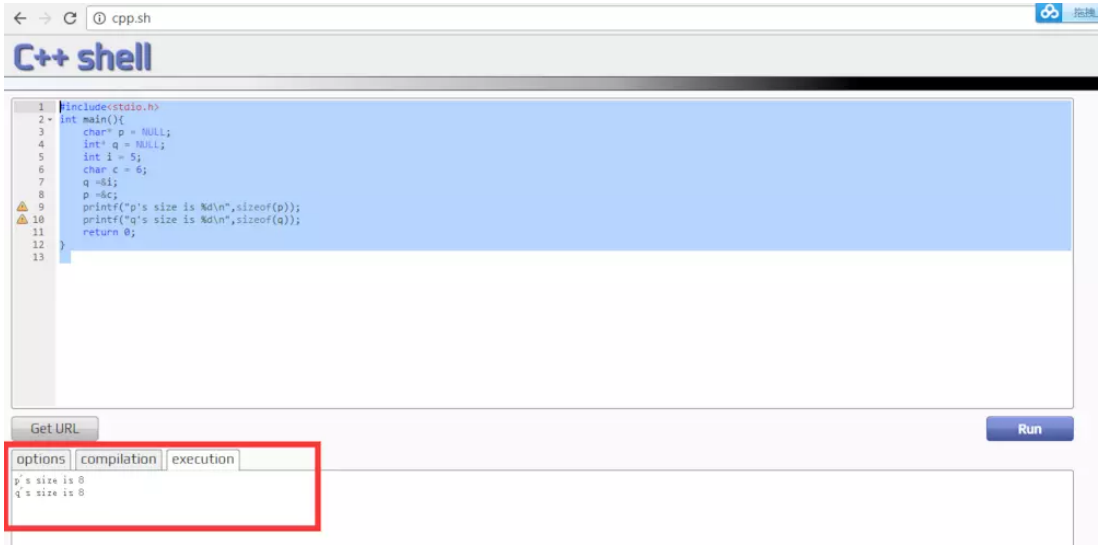
(https://log
yex.youda
slot=30edc
8e69-4995
4e3eddc26
XUy7mYH
click.youda
8e69-4995
4e3eddc26
361156253



```
#include<stdio.h>
int main(){
    char* p = NULL;
    int* q = NULL;
    int i = 5;
    char c = 6;
    q = &i;
    p = &c;
    printf("p's size is %d\n",sizeof(p));
    printf("q's size is %d\n",sizeof(q));
    return 0;
}
```

(/apps/redi
utm_sourc
banner-clc

我们在cpp.sh上编译运行的结果是

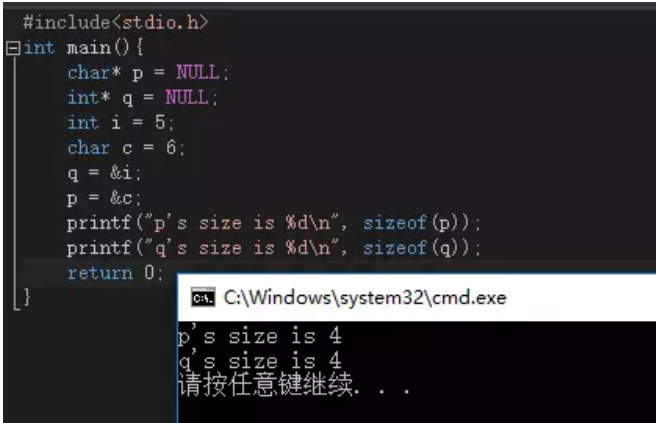


(https://log
yex.youda
slot=30edc
8e69-4995
4e3eddc26
XUy7mYH
click.youda
8e69-4995
4e3eddc26
361156253

Paste_Image.png

而在我的电脑上运行的结果是

不同的编译器造成的



Paste_Image.png

对呀，同样的代码，为什么有不同的结果呢？因为我们的运行环境不一样，在我电脑中指针只占用4的字节，而在cpp.sh服务器上每个指针占用8的字节。但是我们要强调的重点是我们char类型的指针和int类型的指针占用的字节数在同一机器架构下是一样的，也就验证了我们刚刚所言。



访问对应地址中的元素

*是解引用运算符，用来访问指针所指地址上的变量
可以作为右值也可以作为左值
int k = *p;
*p = k+1;



指针的应用场景

场景一

交换两个变量的值

我们都写过交换两个变量的swap函数，但是当我们从形参传进来两个值的时候，我们能在函数内交换，但是一旦离开了函数，原本的变量根本没有变化，因而我们要传入地址进来

```
void swap(int *pa, int *pb){
    int temp =*pa;
    *pa = *pb;
    *pb = temp;
}
```

这样我们就能交换两个变量的值了

(https://log
yex.youda
slot=30edc
8e69-4995
4e3eddc26
XUy7mYH
click.youda
8e69-4995
4e3eddc26
361156253

场景二

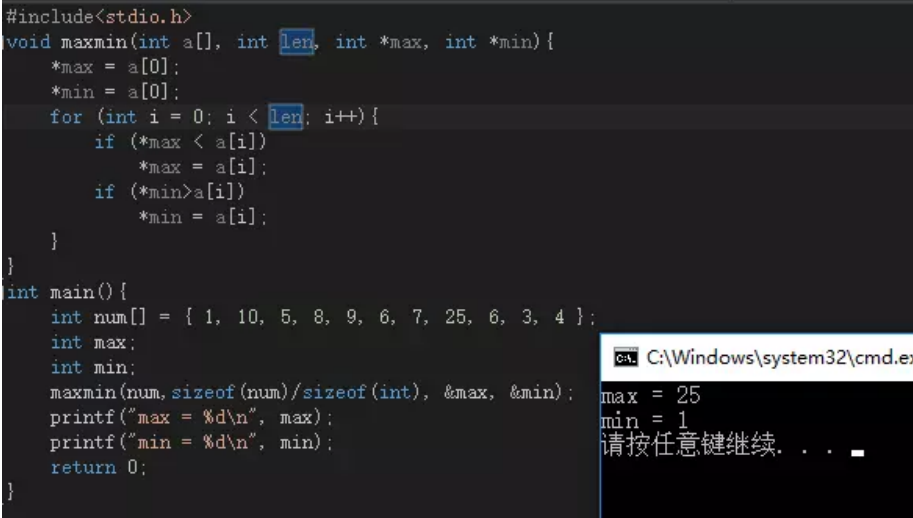
函数要同时返回多个值

一般情况我我们通过函数的return语句返回值的时候只能返回一个，但是如果我们要同时返回多个值该怎么办呢？这时候我们通过参数传入地址，然后通过指针把要返回的值带回到函数外。举个例子：我们写个函数求数组中的最大最下值。



```
#include<stdio.h>
void maxmin(int a[], int len, int *max, int *min){
    *max = a[0];
    *min = a[0];
    for (int i = 0; i < len; i++){
        if (*max < a[i])
            *max = a[i];
        if (*min > a[i])
            *min = a[i];
    }
}
int main(){
    int num[] = { 1, 10, 5, 8, 9, 6, 7, 25, 6, 3, 4 };
    int max;
    int min;
    maxmin(num, sizeof(num)/sizeof(int), &max, &min);
    printf("max = %d\n", max);
    printf("min = %d\n", min);
    return 0;
}
```

(/apps/redi
utm_sourc
banner-clip



```
#include<stdio.h>
void maxmin(int a[], int len, int *max, int *min){
    *max = a[0];
    *min = a[0];
    for (int i = 0; i < len; i++){
        if (*max < a[i])
            *max = a[i];
        if (*min > a[i])
            *min = a[i];
    }
}
int main() {
    int num[] = { 1, 10, 5, 8, 9, 6, 7, 25, 6, 3, 4 };
    int max;
    int min;
    maxmin(num, sizeof(num)/sizeof(int), &max, &min);
    printf("max = %d\n", max);
    printf("min = %d\n", min);
    return 0;
}
```

C:\Windows\system32\cmd.exe
max = 25
min = 1
请按任意键继续. . .

Paste_Image.png

(https://log
yex.youda
slot=30edc
8e69-4995
4e3eddc26
XUy7mYH
click.youda
8e69-4995
4e3eddc26
361156253

在maxmin函数中，我们返回值类型是void，也就是不通过return返回值，但是我们确实得到了max和min，如何做到的？那就是我们穿进去两个地址，在函数内部把地址上的内容改了，所以函数结束以后，我们能得到我们想要的。我们在进一步探索一下，maxmin函数的参数列表中除了我们要传入的数组int a[]，两个指针变量int *max和int *min，还多了一个长度int len 为什么我们不在函数内部做sizeof(a)/sizeof(int) 来得到数组的长度呢？我们来看一个例子

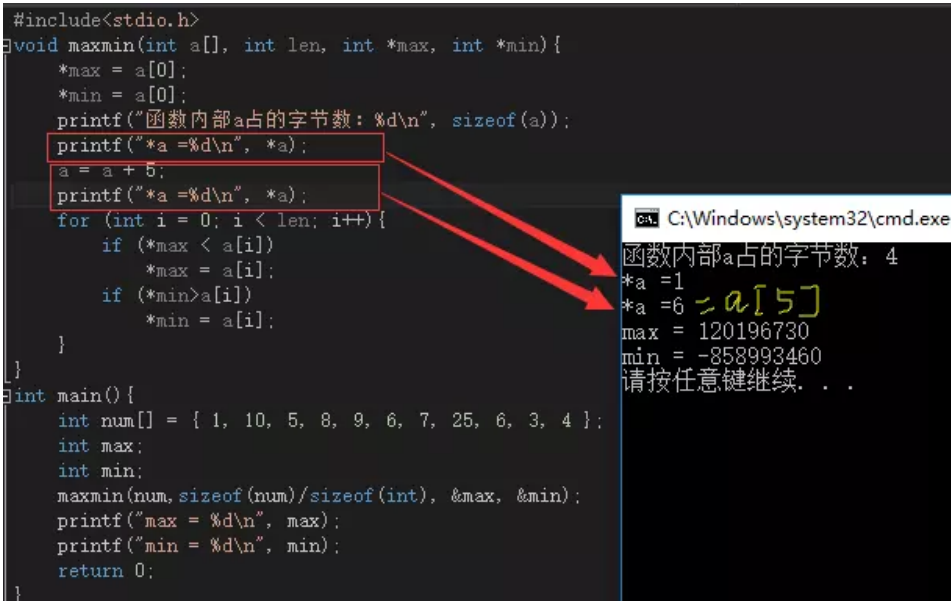


```
#include<stdio.h>
void maxmin(int a[], int len, int *max, int *min){
    *max = a[0];
```

Paste_Image.png

(/apps/redi
utm_sourc
banner-clc

为什么我们的数组才占了4个字节呢?显然，此时a并不表示数组了，那么我们传进来的到底是什么呢？4个字节刚好是一个指针所占用的字节数，没错此时的a就是一个指针。那么我们能否像操作指针一样操作a呢？我们来试一下：



Paste_Image.png

(https://log
yex.youda
slot=30edc
8e69-4995
4e3eddc26
XUy7mYH
click.youda
8e69-4995
4e3eddc26
361156253

没错我们可以像操作指针一样操作a，所以a其实就是一个指针，所以我们在函数内部是不能通过sizeof(a)/sizeof(int)来求得数组的个数的，那么我们只能再传进来一个参数len。

指针最常见的错误

定义了指针变量，还没有指向任何变量就开始使用指针

当你定义了一个指针变量的时候，由于没有指向任何变量，所以它里面的内容是杂乱无章的，可能是个可以访问的地址，也可能是不可访问的地址，无论哪种都不是我们想要的。

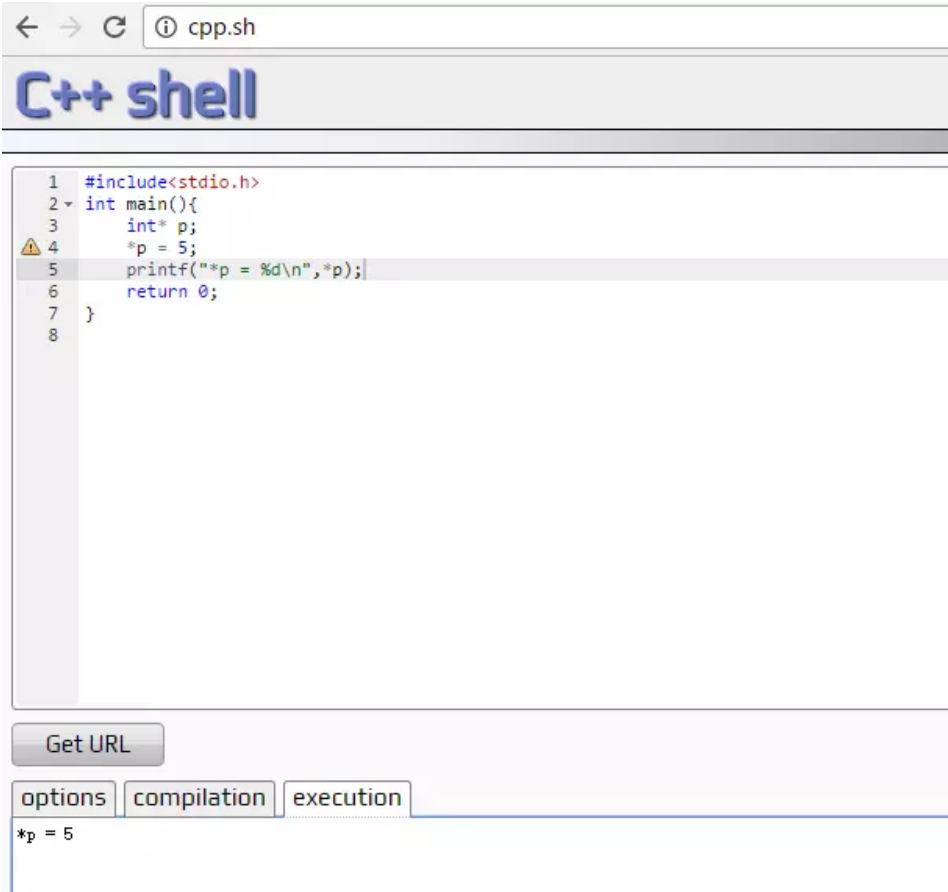




Paste_Image.png

(/apps/redi
utm_sourc
banner-lic

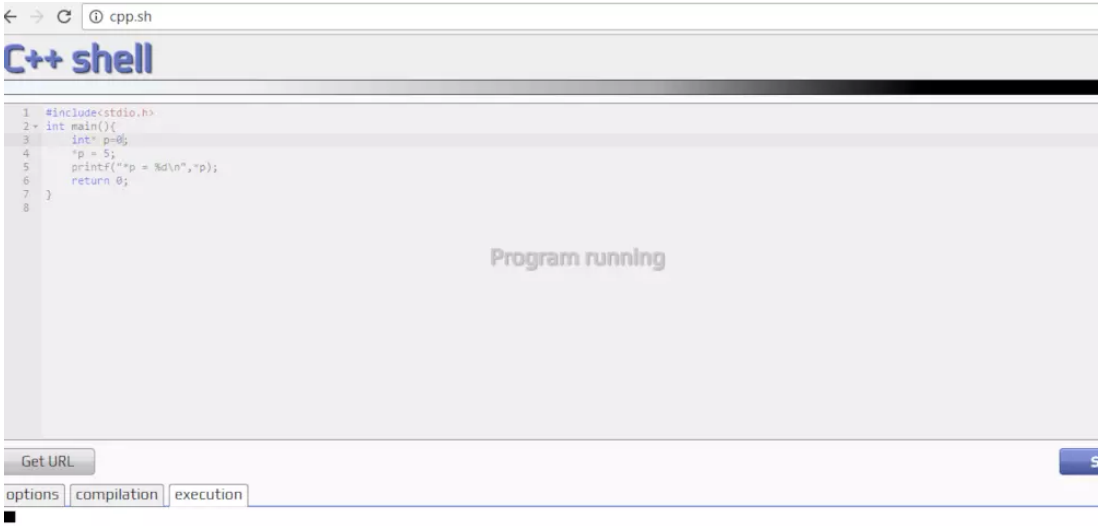
vs提示我们是使用了未初始化的变量，所以连编译都过不去



Paste_Image.png

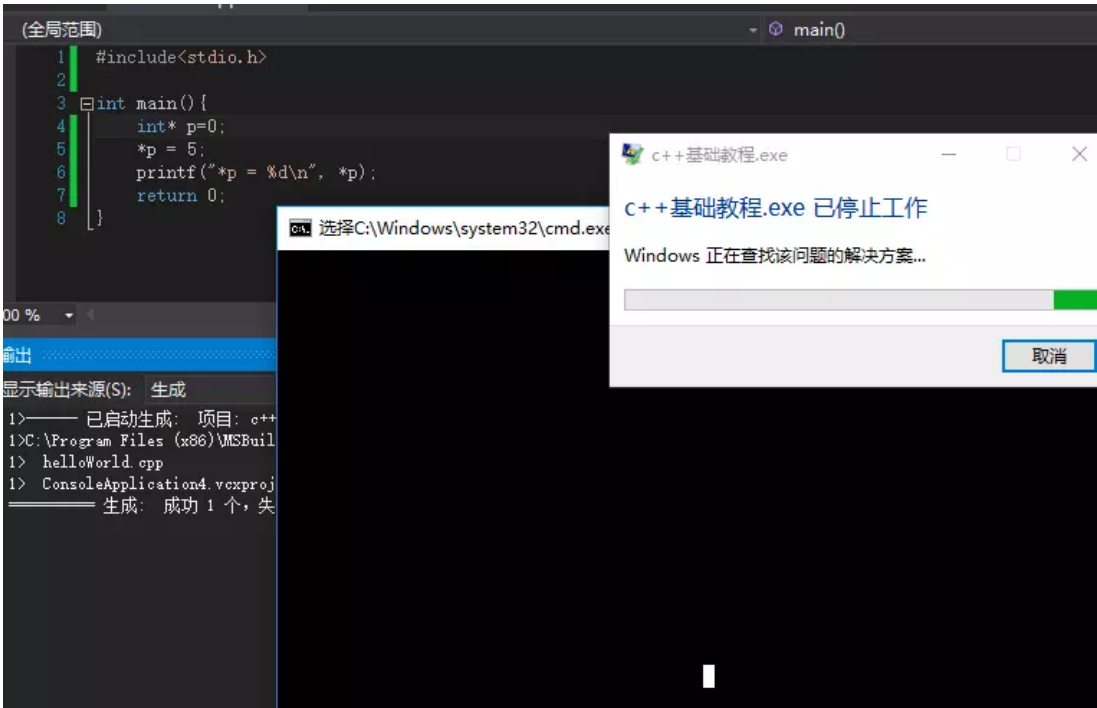
(https://log
yex.youda
slot=30edc
8e69-4995
4e3eddc26
XUy7mYH
click.youda
8e69-4995
4e3eddc26
361156253

我们在cpp.sh上运行了一下，通过了，能正常运行，但这不总是那么幸运，万一我们的p内的地址是0，这个特殊的地址，我们的程序就会崩掉。



Paste_Image.png

我们把p的值改为了0，程序一直得不到结果，在vs下呢



Paste_Image.png

一样的结果，因为我们的0地址是无法访问的。
一般我们用NULL来表示0地址。

指针与const

在使用指针的时候我们经常会看到const，一脸懵逼，你能正确分辨下面的几种写法么？
int i;
int *const p1=&i;
const int *p2=&i;
int const *p3=&i;

三种写法有两种意思，const在*之前，表示指针是const，也就是说指针只能指向这个变量，不能改变，所以一般声明和初始化一起。const在*之后表示不能通过该指针来改变变量的值，并不是说变量不可改变，或者指针不可改变，而是通过该指针改变该变量的值是不允许的。

强制类型转换

void* 这种类型的指针，表示某个地址，但是不明确指出所存储是数据类型。一般用在比较低层的操作中，例如动态分配内存的时候，我们只要告诉系统，请求分配多少字节的内存，并不需要告诉它用来存什么数据，所以malloc返回值类型为void*
必要的时候我们可以通过强制类型转换来改变指针的类型

```
void* p = malloc(128);  
int *q = (int*) p;
```

(/apps/red
utm_sourc
banner-lic

(https://log
yex.youda
slot=30edc
8e69-4995
4e3eddc26
XUy7mYH
click.youda
8e69-4995
4e3eddc26
361156253



malloc动态分配内存

有时候我们不知道数组的大小，要传入一个变量n到数组中，在C99之前是不允许的，这时候就要动态分配内存了
具体做法

← → ↺ ⓘ cpp.sh

C++ shell

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 int main(){
4     int n;
5     scanf("%d", &n);
6     int *p = (int*)malloc(n*sizeof(int));
7     for (int i = 0; i < n; i++){
8         scanf("%d", &p[i]);
9     }
10    for (int i = 0; i < n; i++){
11        printf("%d ", p[i]);
12    }
13    return 0;
14 }
```

Get URL

options compilation execution

5
1 2 3 4 5
1 2 3 4 5

Paste_Image.png


记住不仅仅如此，我们借了内存，还要还回去，不然，有借无还，最终内存会用光的。
我们用free函数还回内存

(/apps/redi
utm_sourc
banner-clc

(https://log
yex.youda
slot=30edc
8e69-4995
4e3eddc26
XUy7mYH
click.youda
8e69-4995
4e3eddc26
361156253




被以下专题收入，发现更多相似内容

-  @IT·互联网 (/c/V2CqjW?utm_source=desktop&utm_medium=notes-included-collection)
-  算法编程 (/c/e9cfd7b50c28?utm_source=desktop&utm_medium=notes-included-collection)
-  程序员 (/c/NEt52a?utm_source=desktop&utm_medium=notes-included-collection)


C语言指针 (/p/627264427bd9?utm_campaign=maleskine&utm_content=...)
指针是C语言中广泛使用的一种数据类型。运用指针编程是C语言最主要的风格之一。利用指针变量可以表示各种数据结构；能很方便地使用数组和字符串；并能象汇编语言一样处理内存地址，从而编出精练而高...

 朱森 (/u/3eb2699ade08?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendatio
8e69-49954e3eddc26XUy7mYHclick.youda8e69-49954e3eddc26361156253

彻底搞定C语言指针——初学者的救赎 (/p/f5906d2f29f1?utm_campaign=...)
1.语言中变量的实质 要理解C指针，我认为一定要理解C中“变量”的存储实质，所以我就从“变量”这个东西开始讲起吧！先来理解理解内存空间吧！请看下图：内存地址→6 7 8 9 10 11 12 13 -----...

 金巴多 (/u/e71be29c578b?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendatio

《C++ Primer》读书笔记 (/p/7b9381f08452?utm_campaign=maleskine&...)
前言 把《C++ Primer》读薄系列笔记全集。目录 第I部分：C++基础 开始学习C++ 变量和基本类型 字符串、向量和数组 表达式 语句 函数 类 第II部分：C++标准库 IO库 顺序容器 范型算法 关联容器 动态内存 第...


 尤汐_Jennica (/u/f11b59cb7098?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendatio

(/p/82d017f8cbf4?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendatio
水滴 (/p/82d017f8cbf4?utm_campaign=maleskine&utm_content=note&...)

在三体世界派出的水滴不费吹灰之力地毁灭了人类的庞大舰队后，水滴终于要去执行它被交付的秘密任务。在三体世界强大的数据分析能力下，水滴来到了外形像是空间站的建筑上空。这个建筑群似乎很有特点，...

 凯恩Kane (/u/0842df1eff77?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendatio

(/p/43f6fcf9dfb7?)



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendatio
出来混，迟早要还的 (/p/43f6fcf9dfb7?utm_campaign=maleskine&utm_c...

(/apps/redi
utm_sourc
recommendatio
banner-clic

01、你有没有遇到过熊孩子？不是调皮好动那类的，而是没礼貌、没教养、坑爹那类的。还记得那个在地铁上直接从别人手里抢吃食的小男孩吗？有没有想拍他的冲动？有句话说的好：每个熊孩子背后都有一对熊...

镜中燃灯 (/u/f9655eb9d642?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendatio

生活有太多的不易，我们要如何学会接受现实 (/p/52e2f87f76c9?utm_cam...

曾经，我以为自己可以活成自己梦想了的样子，但是现实的残酷，逼得我不得不改变生活方式。有太多的事是自己不能左右的，所以一直在妥协，所以活成了自己最不愿活成的样子！你们呢，也是这样吗？

放肆的年华668 (/u/21d6ab7ab9a1?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendatio

爱非坚持/践行100天20170711，第31天【昨天完成情况】 (/p/0ac93933bfb...

早上好！我是滕滕，爱非坚持/践行100天 20170711，第31天【昨天完成情况】 健康：作息时间 早6:00—晚22:00 运动：步行5862步 休闲：祈祷文抄写 【易效能今日三只青蛙】 拆柜子 海外游费用结算 TA53创素...

凤栖梧桐917 (/u/217028e8f43b?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendatio

先调整心态，再调整技能 (/p/bf4006718e6b?utm_campaign=maleskine&...

工作的高效率，一方面跟专业度相关，一方面也跟心态有关，愿不愿意投入更多的有效的时间到工作中去，是给自己积极的规划成长道路，还是当一天的和尚撞一天的钟。团队管理中很多时候，部门的领导容易忽...

Cloudya云 (/u/1efd6640b645?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendatio

(https://log
yex.youda
slot=30edc
8e69-4995
4e3eddc26
XUy7mYH
click.youda
8e69-4995
4e3eddc26
361156253

