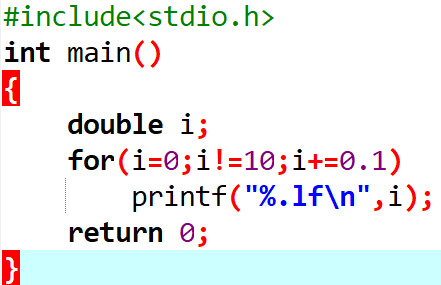
**Tips 1；**

#include<stdio.h>

int main()

{

double i;

for(i=0;i!=10;i+=0.1)

printf("%.lf\n",i);

return 0;

}

注意，该代码中i!=10;并不是说到10就不会执行了，此循环是无限循环。

**Tips 2；**

C++中，想输出的浮点型保留几位小数时用setprecision(5)，使用时要记得加头文件#include<iomanip>

**Tips 3；**

C语言中，不能直接将数组a，b赋值写成a=b；将数组a复制K个元素到数组b；可以这样写：memcpy（b,a,sizeof(int)\*k）;如果数组a，b都是浮点型的，可以将memcpy（b,a,sizeof(double)\*k）,注意使用时要加头文件<string.h>,将aquanbu复制到b是可以这样写memcpy(b,a,sizeoof(a))。

**Tips 4；**

memset(a,0,sizeof(a));作用是把数组a清零；

**Tips 5；**

l+r>>1 *右移除以2，左移乘2；*

l+r的值右移1位，相当l+r的值除以2取整。

Int是将一个数值向下取整为最接近的整数的函数。Int是数据库中常用函数中的取整函数，常用来判别一个数能否被另一个数整除

是比特操作,可以看做是除2，如

15的二进制表示是00001111,15>>1将00001111右移一位，变为00000111，即7.

另外<<就是左移，相当于乘以2.

**Tips6:**

C++ vector用法(原文链接：https://blog.csdn.net/msdnwolaile/article/details/52708144)

１，简述一下vector的基本操作，它的ｓｉｚｅ，capacity（），clear，reverse，reserve

push\_back等！！！

２，说说，vector的存储特性，是顺序存储还是如同链表般，如果是顺序存储的话，那么是如何执行aerse，insert等函数，？？？（假如后面的空间不够的话，我们需要合理的算法来重新找出一块相应的空间吗？？？拷贝，回收吗？？？是不是特别麻烦），如果是链式存储的话，那么它又是如何做到快速的访问的（通过下标来的）！！！

1 基本操作

(1)头文件#include<vector>.

(2)创建vector对象，vector<int> vec;

(3)尾部插入数字：vec.push\_back(a);

(4)使用下标访问元素，cout<<vec[0]<<endl;记住下标是从0开始的。

(5)使用迭代器访问元素.

1. <span style="font-size:18px;">vector<**int**>::iterator it;
2. **for**(it=vec.begin();it!=vec.end();it++)
3. cout<<\*it<<endl;</span>

(6)插入元素： vec.insert(vec.begin()+i,a);在第i个元素后面插入a;

(7)删除元素： vec.erase(vec.begin()+2);删除第3个元素

　　　　　　　 vec.erase(vec.begin()+i,vec.end()+j);删除区间[i,j-1];区间从0开始

(8)向量大小:vec.size();

(9)清空:vec.clear()　　　//清空之后，vec.size()为０

一个简单的程序：

1. <span style="font-size:18px;">#include<stdio.h>
2. #include<vector>
3. #include<iostream>
4. **using** **namespace** std;
5. **int** main()
6. {
7. **int** i=0;
8. vector<**int**> vec;
9. **for**(i=0; i<10; i++)
10. {
11. vec.push\_back(i);　　　//10个元素依次进入，结果为10
12. }
14. **for**(unsigned **int** i=0; i<vec.size(); i++)
15. {
16. cout<<"初始化遍历："<<vec[i]<<endl;
17. }
18. //结果为：０，１，２，３，４，５，６，７，８，９
19. vector<**int**>::iterator it;
21. **for**(it = vec.begin(); it!=vec.end(); it++)
22. {
23. cout<<"迭代遍历："<<\*it<<endl;
24. }
25. //结果为：０，１，２，３，４，５，６，７，８，９
26. vec.insert(vec.begin()+4,0);
27. //结果为:11
28. **for**(unsigned **int** i=0; i<vec.size(); i++)
29. {
30. cout<<"插入遍历："<<vec[i]<<endl;
31. }
32. //结果为：０，１，２，３，０，４，５，６，７，８，９
33. vec.erase(vec.begin()+2);
34. **for**(unsigned **int** i=0; i<vec.size(); i++)
35. {
36. cout<<"擦除遍历："<<vec[i]<<endl;
37. }
38. //结果为：０，１，３，０，４，５，６，７，８，９
39. vec.erase(vec.begin()+3,vec.begin()+5);
41. **for**(vector<**int**>::iterator it = vec.begin(); it!=vec.end(); it++)
42. {
43. cout<<"迭代遍历："<<\*it<<endl;
44. }
45. **return** 0;
46. }

49. </span>

其他相关知识原文链接：<https://blog.csdn.net/msdnwolaile/article/details/52708144>

Tips:7

测试时间:

#include<ctime>

int main()

{

begin = clock();

end = clock();

return 0;

}