目录

[动态数组二维数组的行和列坐标 1](#_Toc82019190)

[string 1](#_Toc82019191)

[vector 动态数组 1](#_Toc82019192)

[queue队列 2](#_Toc82019193)

[优先队列(priorify\_queue) 2](#_Toc82019194)

按“F6”，这将使用设置中的默认参数编译和运行文件。或按“F7”，这将使用您为程序指定的参数。

//读入缓冲区

freopen("1.txt", "r", stdin);//读入

freopen("1.txt", "w", stdin);//输入输出

bool is = true;//->1 is = false; //->0

int n = s.length();// 数组长度

s.substr(0, 1);//返回截取数组某一段

#include <math.h>//pow(x,y);

递归 和 递推

# 动态数组二维数组的行和列坐标

bool Find(int target, vector<vector<int> > array) {

int row=array.size();//行

int col=array[0].size();//列

# string

 string str="hello";

 string s="Hahah";

 str.insert(1,s);//在原串下标为1的字符e前插入字符串s

 cout<<str<<endl;

 string str1="hello";

 char c='w';

 str1.insert(4,5,c);//在原串下标为4的字符o前插入5个字符c

 cout<<str1<<endl;

 string str2="hello";

 string s2="weakhaha";

 str2.insert(0,s2,1,3);//将字符串s2从下标为1的e开始数3个字符，分别是eak，插入原串的下标为0的字符h前

 cout<<str2<<endl;

# vector 动态数组

std::reverse(arr.begin(), arr.end());

arr为动态数组 倒转元素

用到迭代器

创建：

1.vector<int>v1{1,1};**//创建存int类型，类型可改,初始化可省略**

2.vector<string>v1(20,”zzuecm”);**//创建int类型，长度为20，初始化可省**

vector<vector<int>> vec(row, vector<int> (col,1));//二维初始化

**//略,省略后默认值为0;**

3.vector <int>v[3]**//存int类型二维数组，列可变**

4.vector<vector<int>>v{{1,2},{1},{1,2,3}};**//存int类型二维数组，行列都//可变**

应用：

v.size()//返回数组长度o(1)

vec[i].resize(col)//调整长度为col

# queue队列

释：只能在容器的末尾添加新元素，只能从头部移除元素

创建：

queue<type>q;**//建立一个存放数据类型为type的队列q**

应用：

q.push(item):**//在q的最后添加一个type类型元素item o(1)**

q.pop():**//使q最前面的元素出队o(1)**

q.front():**//获取q最前面的元素o(1)**

q.size():**//获取q中元素个数o(1)**

q.empty():**//判断q是否为空，空放回1，反之0 o(1)**

# 优先队列(priorify\_queue)

释：该容器中顺序与插入顺序无关；与数据优先级有关，本质是一个堆

创建：

//有关string 数组的传入和修改试验

#include<iostream>

#include"string"

using namespace std;

//\* 一下是string 数组的传入与传出发现的 小问题\*/

class Solution {

public:

string stets(string &a) {

a[4] = 'a';

a = "badasd";

a[6] = 'p';

return a;//传进来为4个，传出来也为4个？？？？

}

};

int main() {

Solution p;

string a = "nida";

a[1] = 'r';

cout << p.stets(a);

for (int i = 0; i < 7; i++) {

cout << " " << a[i];

}

return 0;

}