**K. 子串统计**

#include <bits/stdc++.h>

using namespace std;

/\*

长度大于等于L的出现次数最多的子串

如果有多个，输出最长的，如果仍然有多个，输出第一次出现最早的

4

bbaabbaaaaa

枚举：

枚举子串的长度，枚举子串的起点

bbaa 2

baab 1

aabbaabb

\*/

string s,t;

string ans;//结果

int ma;//最多子串出现的次数

int len,c,p;

int main() {

    cin>>len>>s;

    int n = s.size();

    //枚举子串的长度

    for(int i = len;i <= n;i++){

        //枚举子串的起点：倒数第i个字母

        for(int j = 0;j <= n - i;j++){

            t = s.substr(j,i);//子串

            //求子串在字符串中出现的次数

            c = 0;

            p = s.find(t);

            while(p != -1){

                c++;

                p = s.find(t,p+1);

            }

            //打擂台：长度为len的子串t，在字符串中出现次c次

            if(c > ma || c == ma && t.size() > ans.size()){

                ma = c;

                ans = t;

            }

        }

    }

    cout<<ans;

    return 0;

}

**P. 时钟旋转（2）**

求时间差的分钟数 \* 0.5。

**R. 跑马灯匹配**

有2个字符串s1和s2。

将第1个字符串所有将第1个字母移动到最后的可能都取出来，分别和s2比较一下相同位置值相等的字符的个数。

**W. 字符串编辑**

#include <bits/stdc++.h>

using namespace std;

string s;

char op,c1,c2;

int main() {

    getline(cin,s);

    cin>>op>>c1;

    //特判特殊情况

    if(s.find(c1) == -1){

        cout<<"Not exist";

        return 0;

    }

    //依次处理各个情况

    if(op == 'D'){

    }else if(op == 'I'){

        cin>>c2;

    }else if(op == 'R'){

        cin>>c2;

    }

    cout<<s;

    return 0;

}

**一、string和数值转换**

为了使用sto?函数，在devcpp中做一个设置：

工具->编译选项->编译时加入如下指令：-std=c++11，才能支持。

**去年10月之后，CCF相关的竞赛（CSP、NOIP等）支持C++14。**

**1、相关函数**

**需要C++14支持的函数：stoi、stod、stoll、to\_string**

    string s1 = "123";

    int a = stoi(s1);

    cout<<a<<endl;

    string s2 = "3.14";

    double b = stod(s2);

    cout<<b<<endl;

    string s3 = "12345678912345";

    long long c = stoll(s3);

    cout<<c<<endl;

    int x = 12345;

    string s4 = to\_string(x);

    cout<<s4<<endl;

**不需要C++14支持的函数：atoi(s.c\_str())**

    string s = "123";

    //s.c\_str()：将字符串s转换为字符数组

    //atoi()：将字符数组转换为对应的整数

    int x = atoi(s.c\_str());

    cout<<x + 10;

注意：将字符转换为对应的整数，要减掉字符’0’。

    char c = '0';

    int x = c - '0';

**将整数转换为字符串：**

#include <bits/stdc++.h>

using namespace std;

//将整数转换为字符串

string tostr(int n){

    string r = "";

    while(n != 0){

         r = char(n%10+'0') + r;

         n = n / 10;

    }

    return r;

}

int main()

{

    string s = tostr(123);

    cout<<s;

    return 0;

}

**在devcpp中的设置：**

**工具->编译选项->编译时加入以下命令：-std=c++11**

**2、例题讲解**

**1130 - 简单a+b**

#include <bits/stdc++.h>

using namespace std;

string s;

int main(){

    cin>>s;

    //123+45=

    int p = s.find("+");

    string s1 = s.substr(0,p);

    string s2 = s.substr(p+1,s.size()-1-p-1);

//    cout<<s1<<endl<<s2;

    cout<<stoi(s1) + stoi(s2);

    return 0;

}

一、什么是vector

向量（vector）是一个顺序容器（Sequence Container），它能够存放各种类型的对象。可以简单的认为，向量是一个能够存放任意类型的动态数组（元素个数可变）。

**vector：动态数组。**

STL：标准模板库！

char s[]：长度固定；

string：可以理解为长度可变的字符数组；

STL是可以比较字典码。

二、vector的常见函数

|  |  |
| --- | --- |
| **函数名** | **函数说明** |
| push\_back(元素) | 增加一个元素到向量后面 |
| pop\_back() | 弹出（删除）向量的最后一个元素 |
| insert(**位置**,元素) | 插入元素到向量的指定**位置（位置指针，不是下标）** |
| insert(**位置**,个数n,元素) | 插入n个相同的元素到指定位置 |
| erase(**位置**) | 删除指定位置的元素 |
| erase(开始位置**,结束位置**) | 删除向量中[first,last)中元素 |
| clear() | 清除向量所有元素，size()变为0 |
| 运算符[i] | 取向量下标为i的元素 |
| begin() | 返回向量头指针（迭代器），指向第一个元素 |
| end() | 返回向量尾指针，指向向量最后一个元素的下一个位置 |
| rbegin() | 反向迭代器，指向最后一个元素 |
| rend() | 反向迭代器，指向第一个元素之前的位置 |
| size() | 返回向量中实际元素的个数。 |
| empty() | 判断向量是否为空，等价于size()为0。 |
| swap() | 交换两个同类型向量的数据。 |

对应于数组，要注意：向量的大小是可变的，开始时向量为空，随着不断插入元素，向量自动申请空间，容量变大。

注意学会使用：sort()、reverse()等函数对vector进行排序、逆序等操作。

例子：vector的存储和遍历

注意掌握vector构造的4个常见方法：

**（1）vector<> v:创建一个空vector**

（2）vector<> v(n):创建一个元素个数为n的vector

（3）vector<> v(n,t):创建一个元素个数为n且值为t的vector

（4）vector(begin,end):复制[begin,end)区间内另一个数组的元素到vector中

三、vector的应用

1211 - 数组元素的插入

#include <bits/stdc++.h>

using namespace std;

vector<int> v;//动态数组

int n,x,y;//x读入的每个数

int main(){

    cin>>n;

    for(int i = 1;i <= n;i++){

        cin>>x;

        v.push\_back(x);//查到vector末尾

    }

    cin>>x>>y;//在第x个数的位置（下标：x-1）

    //v.insert(地址,值);

    v.insert(v.begin()+x-1,y);

    //输出

    for(int i = 0;i < v.size();i++){

        cout<<v[i]<<" ";

    }

    return 0;

}

1162 - 数组元素的删除

#include <bits/stdc++.h>

using namespace std;

vector<int> v;//动态数组

int n,x;//x读入的每个数

int main(){

    cin>>n;

    for(int i = 1;i <= n;i++){

        cin>>x;

        v.push\_back(x);//查到vector末尾

    }

    cin>>x;//删除第x个数的位置（下标：x-1）

    v.erase(v.begin()+x-1);

    for(int i = 0;i < v.size();i++){

        cout<<v[i]<<" ";

    }

    return 0;

}

1453 - 橘子排队

#include <bits/stdc++.h>

using namespace std;

/\*

1. 有n个数，删掉最小、最大

2. 剩余算平均，并输出

\*/

vector<int> v;

int n,x,s = 0;

int main(){

    cin>>n;

    for(int i = 1;i <= n;i++){

        cin>>x;

        v.push\_back(x);

        s += x;

    }

    //排序(v.end()：最后一个数的后一个数的位置)

    sort(v.begin(),v.end());

    printf("%.1f\n",(s - v[0] - v[v.size()-1]) \* 1.0 / (n-2));

    //删除最小最大

    v.erase(v.begin());

    v.erase(v.begin() + v.size() - 1);

    //输出

    for(int i = 0;i < v.size();i++){

        cout<<v[i]<<" ";

    }

    return 0;

}

注意：

    vector<int> v;

    v.push\_back(10);

    /\*

    xx.size()：返回值为unsigned long long

    因此：xx.size()-t如果是负数，实际打印出来会是一个无穷大的数

    \*/

    cout<<v.size()<<endl;//1

    cout<<v.size()-1<<endl;//0

    cout<<v.size()-2<<endl;//无穷大

1502 - 数字翻转

注意：学会在vector中使用reverse函数。

#include <bits/stdc++.h>

using namespace std;

vector<int> v;

int n,m,x,y;

int main(){

    cin>>n>>m;

    //读入n个数

    for(int i = 1;i <= n;i++){

        cin>>x;

        v.push\_back(x);

    }

    //读入m个操作

    for(int i = 1;i <= m;i++){

        cin>>x>>y;

        //提供2个地址：开始地址,结束地址

        //注意：不包含结束地址

        reverse(v.begin()+x-1,v.begin()+y);

    }

    for(int i = 0;i < v.size();i++){

        cout<<v[i]<<" ";

    }

    return 0;

}