1. 一定记得先写

Int mian（）

{

Return 0；

}

因为你的编译器不会有问题，但是ccf测试结果成绩为0。

2.一定反复审题，审题后自己写出一到两个样例，而且还有注意Corner case。

3. cout << sizeof(long int);

cout << sizeof(long long);

Long int和int在32位时一样长的，只有4个字节，32位

Long long才是8个字节，64 位

4.遇到无输出的情况可能是：

声明数组超过上限2147483647个字节；

5.字符串的处理

strcat(s1, s2);  
连接字符串 s2 到字符串 s1 的末尾。 注意直接声明的数组a[100]，里面为，必须写入’\0’之后才能使用strcat；

调用函数对字符串进行处理后，若返回值为一个声明的局部数组的地址，那么返回的结果将会什么都没有，因为作用域的原因。

6.atoi 和itoa

头文件：Include<stdlib.h>

int atoi(const char \*nptr);

返回值为int型，可为正为负，若不能转化为数字，则返回0；

？？处不处理前导零啊（好像处理的）

头文件：<cstdlib>

string itoa(int x,char \*string,int jz)；

写入到数组中。

|  |
| --- |
| #include<iostream>  #include<string>  #include<vector>  #include<map>  #include<set>  #include<sstream>  #include<cctype>  #include<algorithm>  int toi(const string &str)  {  int ans;  stringstream ss(str);  ss >> ans;  return ans;  }  string tos(int i)  {  stringstream ss;  ss << i;  string s = ss.str();  if (s.size() == 1)s = "0" + s;  return s;  } |

7.对于string类型，scanf和printf均不能使用：

全部使用scanf读入，但是如果读string，由于scanf不会改变size所以读不进去，采用如下：

char tmp[20];

scanf("%s",tmp);

**px.assign(tmp);**

对于输出应该使用：

printf("%s\n",**a.c\_str()**);

8.使用sscanf进行字符串的切割

int main()

{

char s[100];

int a;

sscanf("ab 123pc", "%s %dpc", s, &a);

cout << s << ' ' << a << endl; //ab 123

}

注意对于string px要使用

sscanf(px.c\_str(), "%d ", &ipn[i].num[0]);

9.结构体的声明：

struct ru{

string name;

vector <string> wos;

int flag;

};

10.传参和返回值为&的含义？

11.在判断条件千万不要写 “res=1” 和 “0<=res<=1”,

正确表示**res==1；和0<=res&&res<=1;**

12.字符串划分函数：

void split(string s, string c, vector<string> &res) {//注意此处的传参方式

int pos1, pos2;

pos1 = 0;

pos2 = s.find(c,0);

while (pos2 != string::npos) {

res.push\_back(s.substr(pos1, pos2 - pos1));

pos1 = pos2 + c.size();

pos2 = s.find(c, pos1);

}

if (pos1 != s.size())

{

res.push\_back(s.substr(pos1));

}

}

13.读取一行的输入以“\n” 作为分割，则使用getline；

需要注意getline前一个输入是否残留换行符“\r\n”或者“\n”;

char d[100];

cin.getline(d, 100);

14.不要仅使用vs进行编译，因为会自动添加头文件

15.对于**char \*型变量，不能使用“==”**，（但不会报错所以易错），要使用strcmp，需要加入string.h头文件。

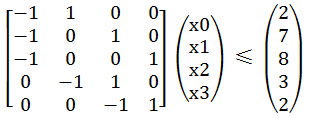
# 图算法

1. 最短生成树

181204，审题困难，最好假设它是，然后反推

2.查分约束

对于如若一个系统由n个变量和m个不等式组成，并且这m个不等式对应的系数矩阵中每一行有且仅有一个1和-1，其它的都为0，这样的系统称为**差分约束( difference constraints )**系统。引例中的不等式组可以表示成如图三-1-1的系数矩阵。



图三-1-1

特点：全是整数，可以将＜转化为≤

常用方法：

1. 对于想用查分约束的情况，若发现项数不满足上述情况，可以尝试前缀和的形式。

我们可以经b1+b2 b1+b2+b3 等转化为 s2-s0 s3-s0  这种前缀和的形式 然后再将答案转化为各个数字

2.注意不要漏掉特殊情况，比如某个变量等于确定的值，即

s[23] = T，它并不是一个不等式，我们需要将它也转化成不等式，由于设定s[-1] = 0，所以 s[23] - s[-1] = T，它可以转化成两个不等式：

      s[23] - s[-1] >= T

      s[-1] - s[23] >= -T

3.对于原本不是整数，后来变为整数的舍入等于，可以改写为≤

4. 如果给出的不等式有"<="也有">="，又该如何解决呢？很明显，首先需要关注最后的问题是什么，如果需要求的是两个变量差的最大值，那么需要将所有不等式转变成"<="的形式，建图后求最短路；相反，如果需要求的是两个变量差的最小值，那么需要将所有不等式转化成">="，建图后求最长路。怎么求最长路？