二叉树

1. scanf("%s",temp);//可以这样存储字符串吗？？方便后续操作吗？？

这里的temp不用写成temp[4].，只用写变量名就可以了。

2、inorder[in++]=s.top();

s.pop();

在进行pop()之前一定要注意先看看是否空，并且先top（），再pop（）

1. if(strcmp(temp,"Push")==0){//这样使用strcmp对吗？？

strcmp如果字符一样则返回0；注意使用strcmo比较的方式

若str1=str2，则返回零；若str1<str2，则返回负数；若str1>str2，则返回正数。

4、void post(int root){

if(root==-1){

return;

}

post(node[root].lchild);

post(node[root].rchild);

printf("%d",root);//这里有没有好的办法？？？是写一个输出函数吗？？这样写是没有空格的

num++;

if(num<n){

printf(" ");

}

}

在递归中如果要实现没有多余空格的输出，可以定义一个num计数器，然后一样的判断。

对于二叉树，树的问题，一定要注意必须知道根节点，才能进行操作。比如递归等。

字符地址也是要取地址的，scanf("%c %c",&node[i].lchild,&node[i].rchild);只有字符数组不需要

当题目的输入数据有多种类型时，比如有整型，字符型，可以统一成字符型，再写一个取值函数，

Int strchange(){

If(c==’-’)return -1;

Else{

Return c-’0’;

}

在树的问题中由于多存在递归，循环等，但是又要保证有空格限制，因此可以预先写一个输出函数。

void print(int id){

printf("%d",id);

num++;

if(num<n)printf(" ");

}

一旦程序中有大量重复的语句，我可以考虑写一个函数体。而且尽量这么做。

树

1、进行指数幂运算时，有两种方法：1）使用pow函数,并且是在cmath头文件中 2）使用循环

int ans=1;//注意这里一定是1

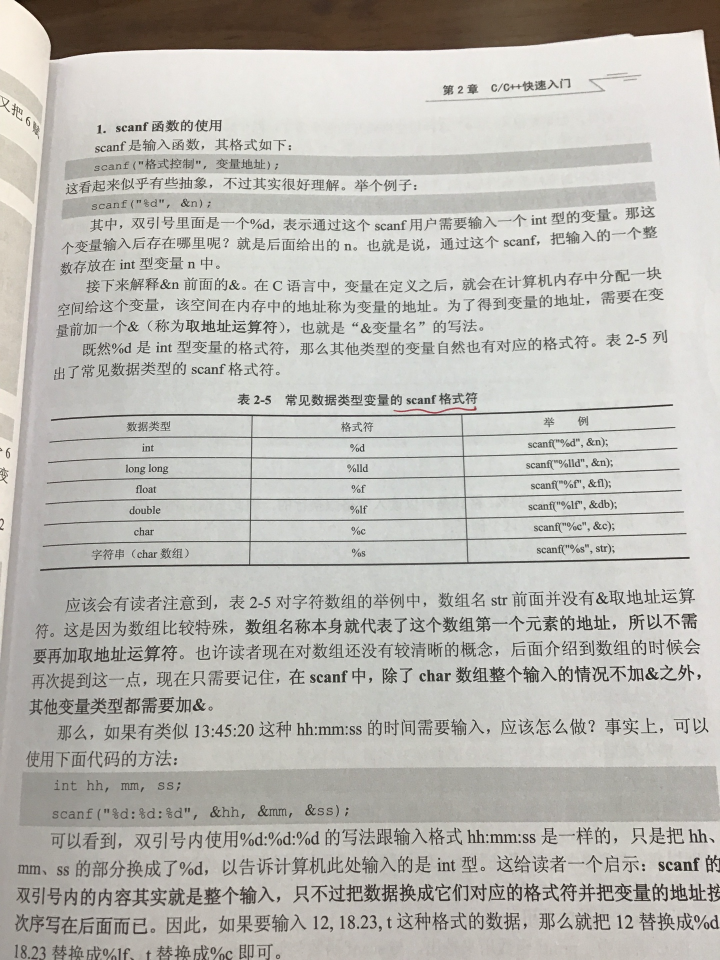
for(int i=0;i<n;i++){

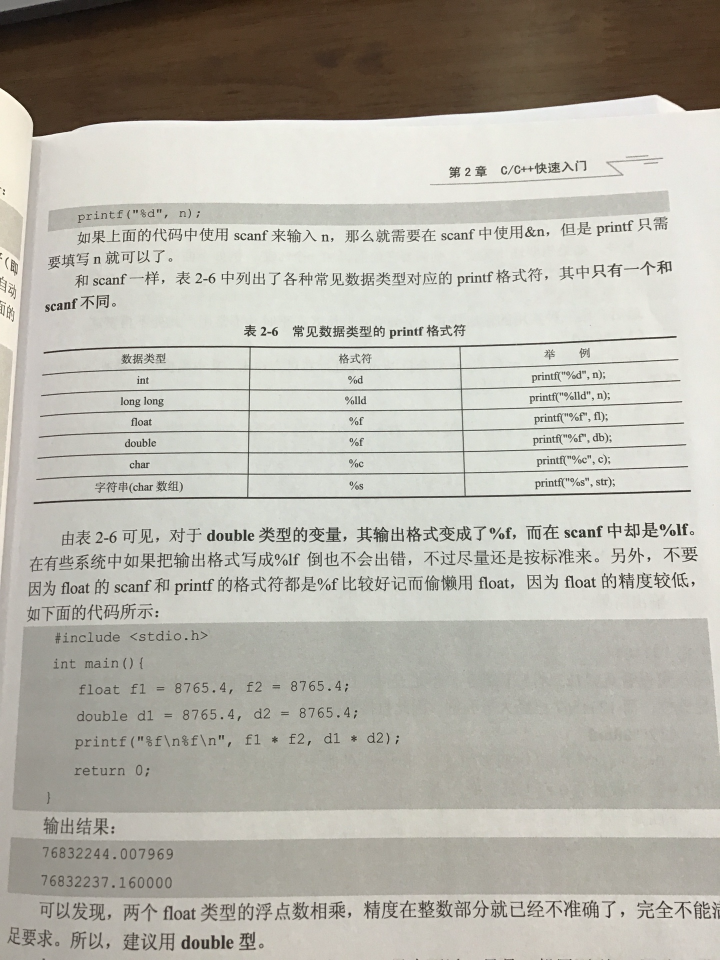
ans\*=5(底数);

}

2、树的叶子结点，递归边界是if(node[root].child.size()==0)

3、常用的输出输出格式：

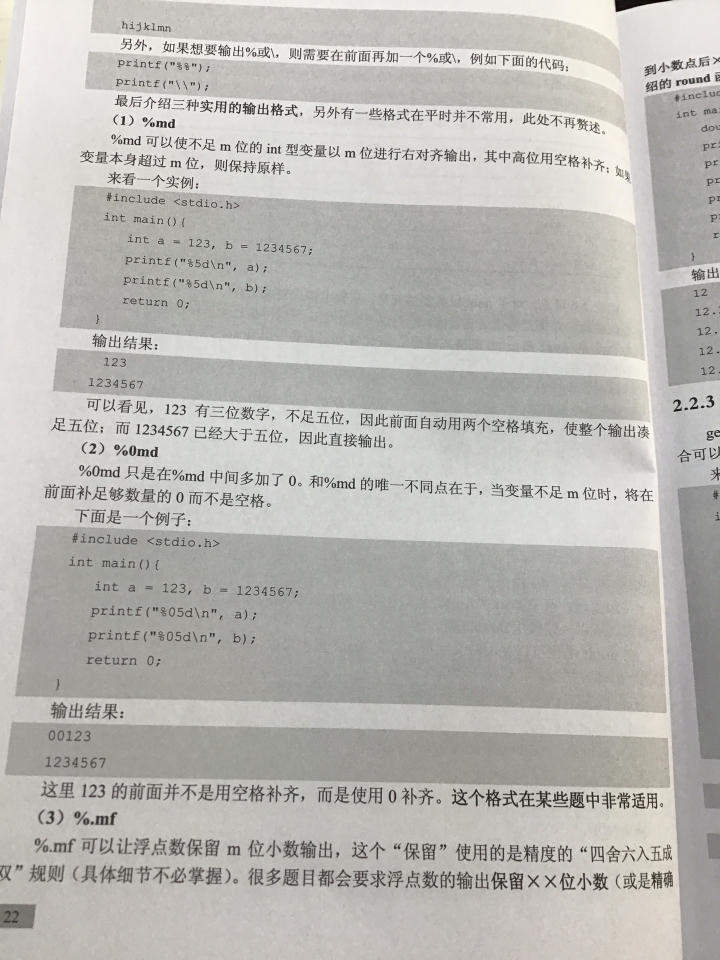




注意double类型的输入输出不一样，并且如果涉及浮点数尽量使用，double类型精度更高。

4、常用的输出格式

1. %md,总体m位，高位不足的补空格，如果数据超过了m位那么不变，那么这样可以实现居中对齐，比如：



1. %0md
2. %.mf
3. 在树的遍历中，如果有高度的话，他是没有把高度融进节点里。高度是随着递归深度在变化，相当于只定一了一个变量。//这样可以更节省空间。
4. 若需要统计每个节点的高度，那么可以单独开一个hash数组，存储。
5. 需要注意的是，第一层root的高度是为1还是0，root的编号是为1还是0；决定了是dfs（0，0）；还是dfs（1，1）；
6. 注意树的遍历dfs可能没有判空的边界条件，也就是没有递归边界。
7. 对于树的静态存储可以

1)使用结构体数组，然后每个节点放一个vector<int>,需要定义结构体。

2)建立一个二维数组；vector<int> g[n];比如；temp的孩子的多少就是，g【temp】.size();比一维的简单了很多。而且这样也不用定义树节点结构体了，只用写一个vector<int> g[n]；//存储树。

1. 题目中编结点号位01，但是可以使用1来进行dfs，dfs(1,1);
2. 如果在一个函数体内，不好实现输出的话，可以先将其放入数组，最后再遍历输出。

入门模拟

1. 循环五次两种写法：

1）for(int i=0;i<5;i++){

printf("%d",i);

}

2)int t=5;

while(t--){

printf("%d",t);

}

1. 对于数的范围，如果是32B的，那么为pow（2，-31）至pow（2，31）-1；

64B就是把31换成63即可，

int和float是32B，double和long long 是64B。所以在题目中应该注意数的范围。

1. strcmp是针对字符串比较的，如果是字符之间的比较直接使用==就可以了：

比如：if（a[i]==da）{},a[]是字符串数组，da是字符。

1. 获取一个数的每一位有两种方法：
2. 使用scanf字符串，存放在数组中。Scanf（“%s”,temp）;
3. 使用模除依次获得，

while(a!=0){

if(a%10==da) pa=pa\*10+da;//注意这里一定是pa\*10，不是temp\*10；

a/=10;

}

1. 时钟处理：printf("%d:%d:%d\n",dif/3600,dif%3600/60,dif%60);
2. int res=round(dif);//round函数，可以实现四舍五入的；

并且像这样，double dif=(end-begin)/100;dif就是一个带浮点的数。

1. 发生段错误还有一种可能就是，scanf()的时候没有&。

8、

int n;

int sum1=0;

int count2=0;

int sum2=0;

int count3=0;

int count4=0;

int sum4=0;

int max=0;

bool flag[5]={0};

float print[5]={0};

像这种会定义很多离散的值的时候，可以使用数组将他们存起来，ans[MAX],count[MAX]等。

1. 多分枝判断的时候有两种1）if-else if 2)switch但是通常都是使用if 比较方便一点。

10、如果不加getchar的话，会出现问题，所以第一个是%c时应该注意使用getchar（）吸收一个换行。

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

getchar();

scanf("%c %c",a,b);

}

1. 对于石头剪刀布等循环相生相克的的东西，可用如下来做到比较，由此衍生出，如果字符间的字典序不满足大小关系，我们将其赋值int，再比较。

char map[3]={'B','J','C'};

int change(char a){

if(a=='B') return 0;

else if(a=='J') return 1;

else return 2;

}

//判断输赢时一定要注意模除，因为是循环相克

if((k1+1)%3==k2)

1. 对于数据具有多分类的题目，一开的时候看看能不能尽量统一处理，比如统一编号，最后再分类，比如根据模除，整除等查找到类别。
2. 类似循环链表，边权值可以用数组存储，但是D【i】表示1到i+1的距离，所以，最后再处理一下dis[right+1]-dis[left-1];就是left到right的距离。并且最短距离是min[temp,sum-temp];所以对于sum要预处理。
3. 把数据定义成long long 之后还不够，就需要进行溢出判断了，并且要，先将res计算出来，溢出了是不知道的，还是会有数存在里面的。

long long res=a+b;

if(a>0&&b>0&&res<0)//正溢出

15、当题目没有说明有多少个数据时，可以使用如下方式，读入数据；

while(scanf("%s",temp)!=EOF){

......

}

while(gets(str)!=NULL){

}

1. 对于多项式的题目，
2. 只需要开一个hash数组即可a[e],表示指数为e的系数，很方便，如果要求指数从大到小输出，那么就hash后往前遍历即可。这种适用于加减，因为只需要让对应的同指数加减就可。
3. 使用结构体存储，这种适用于乘除法：因为乘法要求每个项都要相互乘。

struct Poly{

int exp;

int cof;

}poly[MAX];

17、

printf("%d",count);

for(int i=1000;i>=0;i--){

if(a[i]!=0){

printf(" %d ",i);

printf("%.1f",a[i]);

}

为了保证格式没错，数据总数与具体数据应该分开，处理。比如上图不能写成下图。

printf("%d ",count);//这个空格尽量放在具体数据里，分开。

for(int i=1000;i>=0;i--){

if(a[i]!=0){

printf("%d ",i);

printf("%.1f",a[i]);

}

1. 两个多项式在进行相加，相乘的时候，不用将两个都存入数组后再进行乘，加；

可以先存入一个多项式，再对第二个多项式读入时，边读边操作。

1. 对于一个浮点数在判断大小的时候，应该保留小数。EG：if（ans[i]!=0.0）
2. 对于学生编号之类的虽然是数字，但是可能会产生溢出，
3. 所以应该使用char数组进行存储。
4. 或者采用long long进行存储。
5. 定义宏是可以进行计算的，

const int flag=1814\*365+9\*30+6;

const int now=2014\*365+9\*30+6;//定义宏是不能做计算的吗？？？

1. 在以后进行计数的时候不要再使用count了，用cnt；
2. bool cmp(game a,game b){//注意这里是类型，应该是game而非ga，在排序时，一定得知道你是在排什么类型。
3. C语言规定scanf实型数据不能规定小数点后面位数，你可以修改为scanf("%4f",&f);
4. 在进行四舍五入时，可以先使用round函数，如果不行，再自己写判断函数。（这个就得根据题意了）
5. 以打印沙漏为例，好像所有的图形输出都需要预先处理，比需要知道第一行要打多少个，然后每行依次减2。所有图形输出都是找规律，找到对应输出的公式，一定求公式。
6. 向下取整，可以使用（int）强制转换，当使用sqrt()函数时就是这样。

28、注意while的条件时sum！=0；而非sum%k!=0||sum/k!=0;

while(sum!=0){//注意这是倒着存储的

res[count++]=sum%k;//进制转换就是这样吗？？注意while的一个条件设置

sum/=k;

}

1. 如果输出结果中有负号，那么需要提前判断并输出输出“-”。
2. 如果在程序中不好控制输出格式，呢吗可以先实现功能，再设置一个标志位，最后再根据标志位进行输出，比如回文数判断，可以先判断是否是回文数，再设置标志位，将功能程序运行完了以后再根据标志位来进行输出。
3. 转换成任意进制：

char change(int x){

if(x<=9) return '0'+x;

else return x-10+'A';

}

1. 如果一个字符不是数字怎么写？？

If（！(id[j]>'0'&&id[j]<'9')）

1. 字符数组只能每一个下标存储一个字符，
2. 指针数组可以，一个下标存储一个字符串
3. 也可以使用二维数组，a[i]就是下标。
4. 当在控制台使用EOF时，要么ctrl+c要么ctrl+d;
5. 在实际编写程序的时候注意考虑，数据的边界，比如上界，下界，一般很多时候数据为零的情况会单独写出来，注意注意啊！！！！
6. 在处理字符串的时候，如果正方向不好做，可以考虑反方向做，也就是将其翻转后做。比如求公共后缀的题目。可以使用algorithm头文件下的reverse函数来反转。

37、当有scanf()时注意观察前后有没有%c读入的，如果有就要使用getchar()吸收换行符。

数学

1. 针对拿到题目没有思路的题目，可以先用人脑实现一遍，再考虑使用什么算法，以及使用什么数据结构，最后再形成代码。
2. scanf是以空格或者换行符为结尾的。Gets（）是以换行符为结尾。所以如下：

scanf("%d",&temp);

getchar();//这里一定要有一个getchar（）来吸收换行字符。

int i=9;

while(i--){

gets(str);

}

1. 思路：如果一个字符串我们只允许有P、A、T那么如何知道有没有其他字符？？可以只关注P、A、T的数量，其他的字符我们预先也不会知道是啥，所以就统计其他的数量就行，如果数量不为零则，肯定有其他字符。
2. 不知道要循环多少次可以使用如下循环：

while(1){

if(,,,,)break;//设置条件进行跳出。

}

1. 传递数组的方式：void int\_to(int x,int temp[])

int\_to(n，temp);下面就不用写temp[]了。

1. 数列片段每个片段出现次数，i\*（n+1-i）
2. 对于每个题目一定要试试边界的数据运行的咋样，这样也是有分数的。可以骗分。
3. 在写函数的时候一定要关注几个方面：1）函数传值参数2）函数是否有return；
4. 在进行分数运算的时候一定要先输出化sum的up,down值。
5. 进行分数的定义时，尽量使用long long 数据类型，避免溢出。
6. 针对于对一个分数输出后带括号的问题，可以再输出函数中，前后夹上括号：

if(a,up<0)printf("(");

//中间是输出函数体；

if(a.up<0)printf(")");

1. 发生了浮点错误，可能是因为1）浮点数定义有问题，2）除法出现了除以0的情况。

for(int i=2;i<=sqr;i++){

if(x%i==0){//如果上面的i从0开始，就可能发生浮点错误。

return false;

}

}

1. 注意在实现素数时，检查是否是素数，有sqr，但是在n的范围内打印素数就是没有sqr的，范围是n。
2. 输出格式，每10个为一行，每行最后无空格，但中间有空格。

if(cnt%10!=0&&cnt<Pnum)printf(" ");

else printf("\n");

1. 在不知道输出多少数据时可用while,,EOF,但如果题目中说了以负数，结尾的话，就是如下：

while(scanf("%d",&n)!=EOF){

if(n<0)break;

}

如果时如下:好像也是可以的。

while(1){

scanf("%d",&n);

if(n<0)break;

}

1. 发生了题目出错前，先检查比较简单的一些错误（常见的，检查一下，浮点错误、段错误等等），再看题目的意思是不是读懂了，然后再看看，在进行仔细的调试。不要一上来就是调试或者读懂意思。
2. 对于质因子分解成连续的数，可以类似的双指针，第一个指针记录能够整除的数，然后再往后探测，直到不能整除为止（因为往后不知道会循环多少次，所以使用while循环。）
3. 在分解质因数的那里，一定得注意是<=sqr,还是<sqr。
4. 这里注意：

find\_prime();

if(n==1)printf("1=1");//一定要先进行，特判。

1. 发生了段错误，很可能是越界了：
2. 数组大小增大；
3. 循环的边界看看有没有越界，

for(int i=a.len-1;i>=0;i--){//如果这里写成i++，那么就会段错误。

1. scanf（）没有取地址。
2. 将数组倒置的方法，有两种，
3. 自己写一个倒置函数，主要就是将第一个与最后一个交换。
4. 使用algorithm下的reverse函数。
5. 在题目中有循环出现时，一定要分清楚是不是已知循环次数
6. 已知的话，使用for循环。
7. 不已知使用while,再发现结束循环的条件。

1. 一定一定要注意整个程序之间的数据范围，很可能就是给出的数据没有溢出，但是在运算过程中会发生溢出，这个是需要提前知道的。

STL专题

1. strcmp()函数不一定都是返回-1，0，1的，不同编译器有不同的输出方式，因此在写的时候只能写<0,>0等。
2. 在map专题中，遇到如果不是很好模拟的话，可以采用先将全部的情况枚举出来再查询即可。
3. map和string是一起用的，map可以直接使用字符串作为索引，作为字符串到数字的转化，string作为数字到字符串的转化，所以对于一个特定的字符串，我们应该初始化两个：1）map 2）string
4. 如果不符合题意，采用跳过的方式，怎么实现？？循环（不知道次数，所以是while()）

while(check(str[i])==false){

i++;//直接让下标加加，不做任何操作，就可以实现跳过。

}

1. 在字符串中分割单词的时候使用string会方便很多。

5.20左右重打代码要点

1. 如果要通过DFS进行变量更新，一般都是直接把变量放在DFS（参数里面的），比如DFS（node \*root,int depth）;
2. 只有Dijkstra的时候寻找最短路径的序列是只有vector<int> pre；而非vector<int> pre[MAX],后者是在有DFS的时候进行的。
3. void函数类型不需要return。