二维数组（矩阵）复习

1、二维的输入输出：

注意i,j 的先后顺序与输出的效果

2、矩阵的行列关系、斜线的特点

3、旋转

String 容器的几个注意点：

1、string a[10] 是一个二维数组

2、string a,b;

a=b;　　a+b;　　if(a>b)

string a[10],b[10];　a=b;

3、memcpy(目标，源，n)

memcpy(b,a,sizeof(a))

4、涉及到明确的n行,m列，最好用char a[n][m]　而不要用string a[n]

1、《最好的草》

```
3   int n,m,k,i,j,x[2]= {1,0},y[4]= {0,1},cnt;
4   char a[101][101];
5   int main() {
6       cin>>n>>m;
7       for(i=0; i<n; i++)
8           for(j=0; j<m; j++) cin>>a[i][j];
9       for(i=0; i<n; i++)
10          for(j=0; j<m; j++)
11              if(a[i][j]=='#') {//枚举每一个点,
12                  cnt++;
13                  for(k=0; k<2; k++) //及下和右方是不是'#'
14                      if(a[i+x[k]][j+y[k]]=='#') {
15                          a[i+x[k]][j+y[k]]='.';
16                      }
17              }
18      cout<<cnt<<endl;
```

作业问题：

3、《反反复复》

```
t o i o y
h p k n n
e l e a i
r a h s g
e c o n h
s e m o t
n l e w x
```

```
5
toioynnkpheleaigshareconhtomesnlewx
```

样例输出

```
theresnoplacelikehomeonasnowynightx
```

作业问题：

3、《反反复复》

```
using namespace std;
string s;
char a[20][30];
int main() {
    int i,j,m,n,len,k=0;
    cin>>n>>s; //读入 注意写在一行
    len=s.size();//字符数
    m=len/n; //求行数
    for(i=0; i<m; i++) //构造密码矩阵
        if(i%2==0)
            for(j=0; j<n; j++)
                a[i][j]=s[k++];
        else for(j=n-1; j>=0; j--)
                a[i][j]=s[k++];
    for( i=0; i<n; i++)  //按列优先输出
        for(j=0; j<m; j++)
            cout<<a[j][i];
    return 0;
}
```

错误探测

```
 4  int sumx(int x) { //统计行
 5      int s=0;
 6      for(int j=1; j<=n; j++)
 7          if(a[x][j]==1)s++;
 8      return s%2;
 9  }
10  int sumy(int y) { //统计列
11      int s=0;
12      for(int i=1; i<=n; i++)
13          if(a[i][y]==1)s++;
14      return s%2;
15  }
```

```
17  int main() {
18      cin>>n;
19      for(int i=1; i<=n; i++) //输入
20          for(int j=1; j<=n; j++)cin>>a[i][j];
21
22      int k=0,k2=0,k3=0,x,y;
23      for( int i=1; i<=n; i++)
24          for(int j=1; j<=n; j++) {//枚举每个位置
25              if(sumx(i)==0 && sumy(j)==0)k++;
26              else if( sumx(i)==1 && sumy(j)==1){k2++;x=i;y=j;}
27          }
28          //判断并输出
29      if(k==n*n)cout<<"OK"<<endl;
30      else if(k2==1)cout<<x<<" "<<y<<endl;
31      else cout<<"Corrupt"<<endl;
32      return 0;
```

〈古风排版〉

输出格式：

按古风格式排版给定的字符串，每列

输入样例：

```
4
This is a test case
```

输出样例：

```
asa T
st ih
e tsi
 ce s
```

1、注意读入格式：

```
int n;  string a;
cin>>n;
getline(cin,a);
getline(cin,a);
```

2、构造 二维数组时注意行列

〈古风排版〉

```
3   int n,m,len,i,k,j;
4   string a[1005],s;
5   int main() {
6       cin>>n; //矩阵的行数
7       getline(cin,s);
8       getline(cin,s);
9       len=s.size();
10      m=len/n; //计算出列数
11      if(len%n) {
12          m++;
13      }
14      for(i=m-1; i>=0; i--)  //构造 二维数组
15          for(j=0; j<n; j++) {//先列
16              if(k<len) {
17                  a[j][i]=s[k++];
18              } else {
19                  a[j][i]=' ';
20              }
21          }
22      for(i=0; i<n; i++) {//输出 先行后列
23          for(j=0; j<m; j++)
24              cout<<a[i][j];
25          cout<<endl;
26      }
27      return 0;
```

实验舱蛟龙三班

递归（1）

zlj

2022

什么是递归？什么是递归调用？

《从前有座山。。。。》

函数是如何调用的？电话求助

```
2   using namespace std;
3   int C(int x){
4       return x;
5   }
6   int B(int x){
7       return C(x-1)+x;
8   }
9   int A(int x){
10      return B(x-1)+x;
11  }
12  int main() {
13      int x;
14      cin>>x;//x=5 输出？
15      cout<<A(x)<<endl;
16      return 0;
```

1、四个函数中的x是同一个量变吗？他们的值分别是多少？

2、请说出 A(5) 的计算过程？

下面的代码如何执行？

```
3  int B(int a){
4      return A(a-1);
5  }
6  int A(int a){
7      return B(a-1);
8  }
9  int main() {   //函数调用
10     int x;
11     cin>>x; // x=3
12     cout<<A(x);
13     return 0;
14 }
```

函数能自己调用自己吗？

```
int A(int a){
    return A(a-1);
}
```

```
3  int A(int a){
4      return A(a-1);
5  }
6  int main() {  //函数能自己调用自己
7      int x;
8      cin>>x; // x=3 结果是多少？
9      cout<<A(x);
10     return 0;
11 }
```

下面的代码如何执行？

```
13  int A(int x){
14      if(x<=3)return 3;
15      return A(x-1)+x;
16  }
17
18  int main() {
19      int x;
20      cin>>x;//x=5 输出？
21      cout<<A(x)<<endl;
22      return 0;
23  }
```

什么是递归调用

- 什么是递归调用？
  1、函数A里调用B，或B里调用A
  2、在函数A中直接或间接地调用自己（函数A）

下面程序的结果是：

```
int a(int x){
 if(x==3）return x;
 else  return x+a(x-1);
   }
int main(){
  int x=5;
  cout<<a(x)<<endl;
return 0;
 }
```

1、程序调用了自已多少次？

2、第次调用产生一个新的函数，每个函数中的变量x 的值分别是多少？

3、请说明a(5) 的值是如何计算出来的，分析其计算过程。

练习、写出下列程序的运行结果。

```
void   A( ){
       char c;
       c=getchar();
       if (c==’!’) return;
        A( );
        cout<<c;
  }
int main(){
   A( );
  return 0;}
```

输入:ABCDE!

输出:（　　）

一、递归概念

当函数的定义中，其内部操作又直接或间接地出现对自身的调用，则称递归调用。

二、递归求解问题算法特点：

递归通常把一个大型复杂的问题层层转化为一个与原问题相似的规模较小的问题来求解，递归策略只需少量的程序就可描述出解题过程所需要的多次重复计算，大大地减少了程序的代码量。递归的能力在于用**有限的语句**来**定义对象的无限集合**。用递归思想写出的程序往往十分简洁易懂。

递归解题实现机制

- 递归算法通过函数来实现，通常是“分解问题，转化为子问题”做法，
- 采用以下两种机制：

1、函数调用自身的减少重复代码

2、用参数传递或返回值来传送相关数据

三、递归是如何传递数据的？

- 输入一个整数n，求1+2+3+….+n的和。 s(n)=s(n-1)+n

```
3  int main() {
4      int n,s=0;
5      cin>>n;
6      for(int i=1;i<=n;i++){
7          s+=i;  //循环迭代
8      }
9      cout<<s<<endl;
10     return 0;
```

```
3  int s(int n){
4      if(n==1)return 1;
5      return s(n-1)+n; //递归调用
6  }
7  int main() {
8      int n;
9      cin>>n;
10     cout<<s(n)<<endl;
11     return 0;
12 }
```

递归算法

S(5)=?

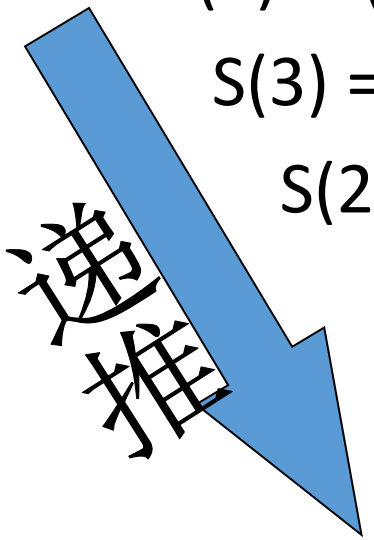
S(5) =s(4) +5

S(4) =s(3) +4

S(3) =s(2) +3

S(2) =s(1) +2

S(1) = 1



s(5)=(10)+5=15

s(4)=(6)+4=10

s(3)=(3)+3=6

s(2)=1+2=3

边界

S（n)=?

展开了几个同名函数？

递归调用小结：

1、递归调用是用函数来实现的。

2、自己调自己，循环的是整个函数

3、要有边界，规模要减少

4、到了边界返回答案时，要将余下的任务完成。再依次返回到入口，输出答案。

递归解题的条件：

1. 需要解决的问题可以**转化**为一个或多个 **子问题**来求解，而这些子问题的求解方法与原来的问题完全相同，只是在数量规模上不同

2. 必须有结束递归的条件出口（**边界条件**）来终止递归。

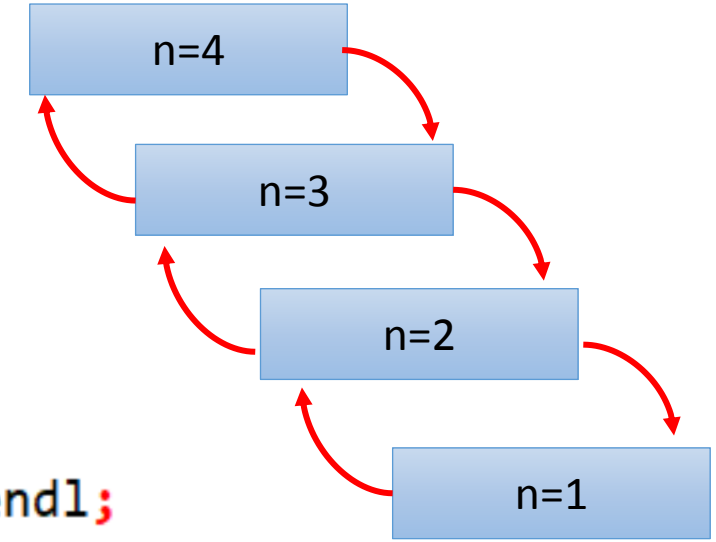
四、递归运行内存地址（掌握运行过程）

```
#include <bits/stdc++.h>
using namespace std;

void hs(int n)
{
    cout <<" Level "<<n<<":"<<"location "<<&n<<endl;
    if(n < 4) hs(n+1);
    cout <<" Level "<<n<<":"<<"location "<<&n<<endl;
}

int main()
{
    hs(1);
    return 0;
}
```

```
Level 1:location 0x70fe30
Level 2:location 0x70fdf0
Level 3:location 0x70fdb0
Level 4:location 0x70fd70
Level 4:location 0x70fd70
Level 3:location 0x70fdb0
Level 2:location 0x70fdf0
Level 1:location 0x70fe30
```

认识递归

```
#include <bits/stdc++.h>
using namespace std;

void hs(int n)
{
    cout <<" Level "<<n<<":"<<"location "<<&n<<endl;
    if(n < 4) hs(n+1);
    cout <<" Level "<<n<<":"<<"location "<<&n<<endl;
}

int main()
{
    hs(1);
    return 0;
}
```




```
Level 1:location 0x70fe30
Level 2:location 0x70fdf0
Level 3:location 0x70fdb0
Level 4:location 0x70fd70
Level 4:location 0x70fd70
Level 3:location 0x70fdb0
Level 2:location 0x70fdf0
Level 1:location 0x70fe30
```

递归函数通常都带有一些局部变量，只有当整个函数体执行完毕，局部变量才被收回失去意义。每递归调用一次，就生成一组“新的”变量，虽然与原来的变量名字相同，但分配的空间不同。

为什么要学递归？ 重复执行整个函数的语句，是循环的升华，代码很简化

想一想：这个程序的输出是什么？

```
#include <bits/stdc++.h>
using namespace std;

void hs(int n)
{
    for (int i=1;i<=n;i++) cout <<n;
    cout <<endl;
    if(n < 5) hs(n+1);
    for (int i=1;i<=n;i++) cout <<n;
    cout <<endl;
}

int main()
{
    hs(1);
    return 0;
}
```

例2 ：阶乘问题

n!=1*2*3*4……*(n-1)*n

函数：

f(n)=n!

f(n)=n*f(n-1)

f(n-1)=(n-1)*f(n-2)

当n=1时，f(1)=1

$$f(n)=\begin{cases}1 & (n=1)\\ n*f(n-1) & (n>1)\end{cases}$$

阶乘问题

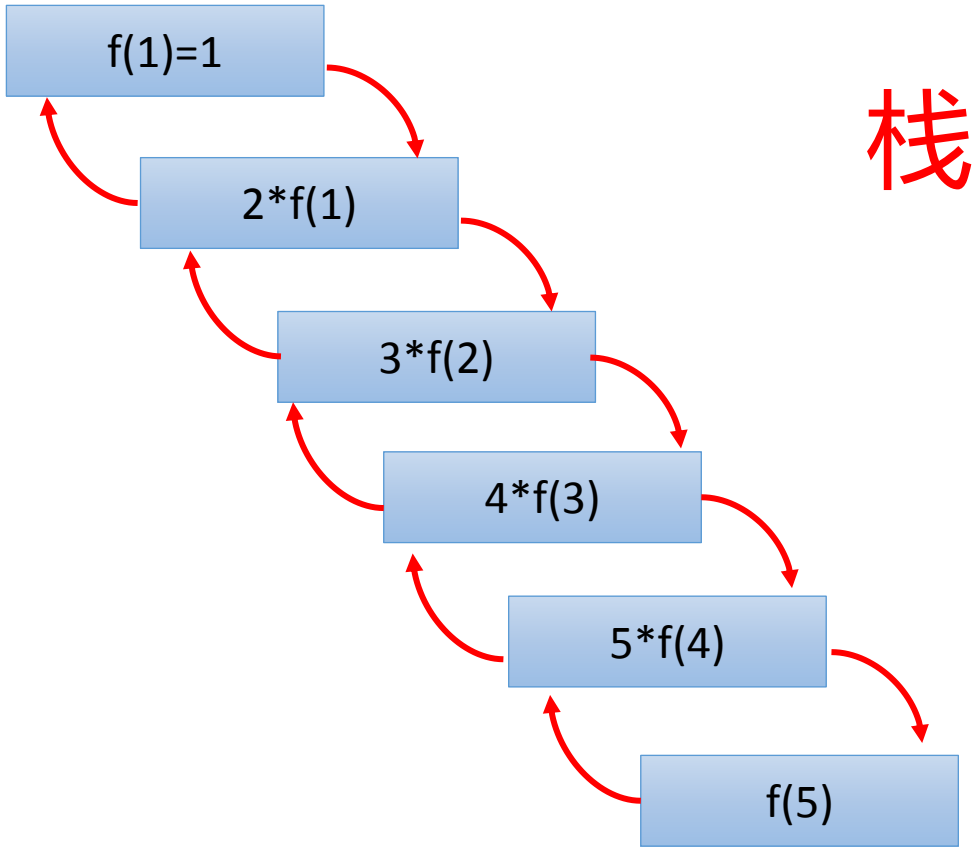
```
#include <iostream>
using namespace std;
int f(int n)
{
    if (n==1) return 1;
    else return n*f(n-1);
}
int main()
{
    int n;
    cin >>n;
    cout <<f(n)<<endl;
    return 0;
}
```

写递归函数时必须遵循下列条件：

1. 存在递归关系，问题可以划归为一个子问题。
2. 递归有终止的条件（边界条件）。

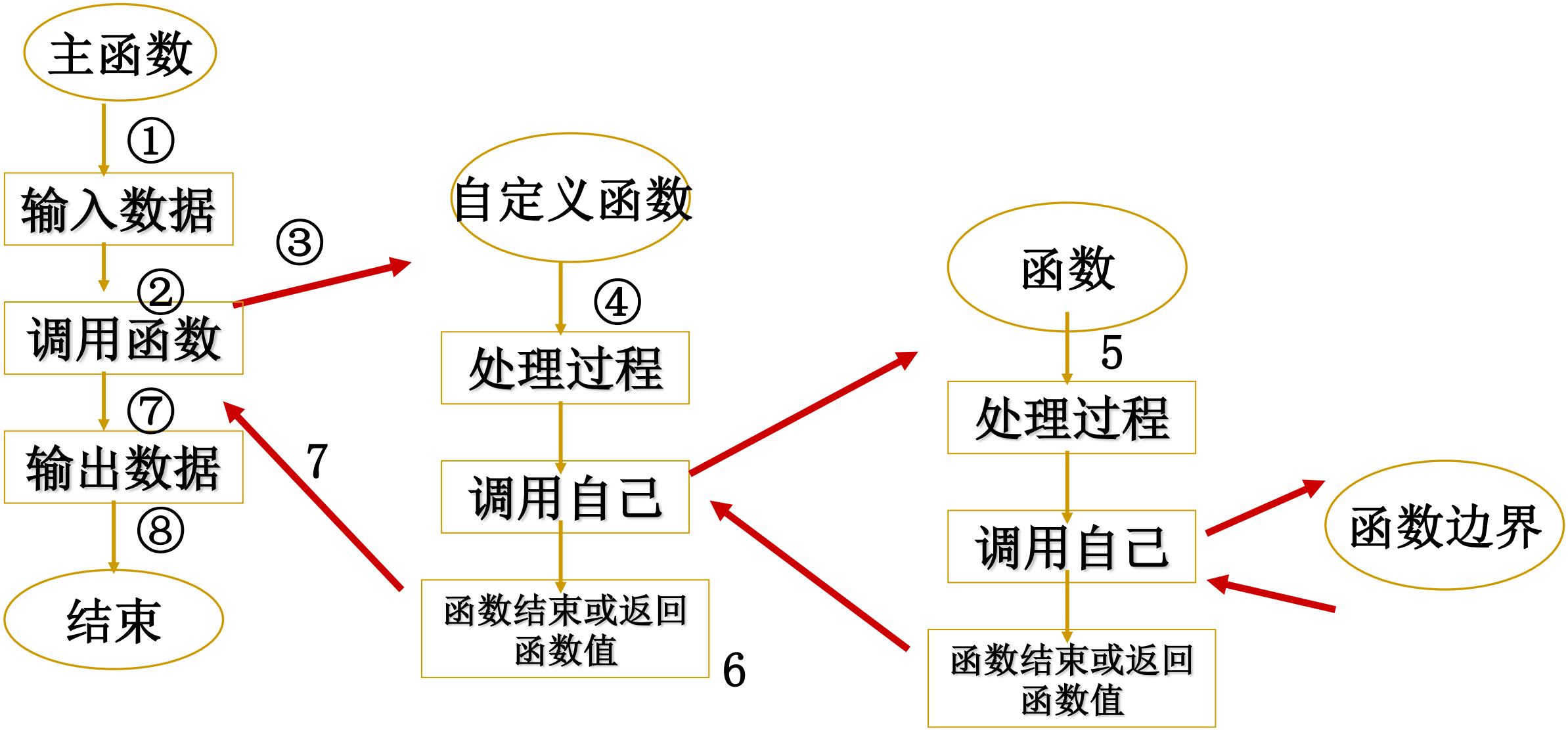
阶乘问题

```
#include <iostream>
using namespace std;
int f(int n)
{
    if (n==1) return 1;
    else return n*f(n-1);
}
int main()
{
    int n;
    cin >>n;
    cout <<f(n)<<endl;
    return 0;
}
```



栈

函数递归调用和返回的过程



猴子吃桃问题

```
15  int main(){
16      cin>>n;
17      int sum[10]={0};
18      sum[7]=n;      //循环解决问题
19      for(int i=6;i>=1;i--){
20          sum[i]=(sum[i+1]+2)*2;
21      }
22      cout<<sum[1]<<endl;
23      return 0;
```

```
 3  int n;
 4  int sum(int k){ //递归思想
 5      if(k==7)return n;
 6      return (sum(k+1) + 2 )*2;
 7  }
 8  int main() {
 9      cin>>n;
10      cout<<sum(1)<<endl;
11      return 0;
12  }
```

三、递归例题应用

例 3、《人口增长问题》

今年2020人口11.2亿，每年增长1%，问2029年后我国人口将达到多少亿？

输入： 2020 11.2 2029 1

输出：12.2493

分析：

```
int  sum(int x){

}   //求 第x年的人口数
```

填空：

```
2  using namespace std;//人口增长
3  int y;double s;
4  double p(int x,double r) {
5      if(x==y)return  (1) ;
6      return  (2) *(1+r/100.0);
7  }
8  int main() {
9      int n;
10     double r;
11     cin>>y>>s>>n>>r;
12     cout<<p(n,r)<<endl;
13     return 0;
14 }
```

阿克曼函数

- Ackmann函数Akm（m,n）函数定义为:
- Akm(m,n)=n+1; (m=0)
- Akm(m,n)=akm(m-1,1); (m>0,n=0)
- Akm(m,n)=akm(m-1,akm(m,n-1));　(m,n>0)
- **【输入格式】**
- 输入两个整数m、n，　m<=3,n<=10。
- **【输出格式】**
- 一个整数，表示函数的值。
- **【输入样例】**
- 2　3
- **【输出样例】**

阿克曼函数

分析：

按照函数的定义模拟即可

```
using namespace std;
int ACK(int m,int n) {
    if(m==0&&n>0)return 1+n;
    if(m>0&&n==0)return ACK(m-1,1);
    if(m>0&&n>0)return ACK(m-1,ACK(m,n-1));
}
int main() {
    int x,y;
    cin>>x>>y;
    cout<<ACK(x,y)<<endl;
    return 0;
}
```

例 5、《二进制》

输入一个整数n,用递归求它的二进制。

```
void er2(int x){
    if(x==0)return;
    er2(x/2);
    cout<<x%2;
}
int main(){
er2(5);
return 0;
}
```

例6：斐波那契数列

输入一个整数N，求斐波那契数列第 N项是多少？

例6：斐波那契数列

0，1，1，2，3，5，8，13，21，34， 55……

- 递归关系

fib(n)=fib(n-1)+fib(n-2)　　n>=2

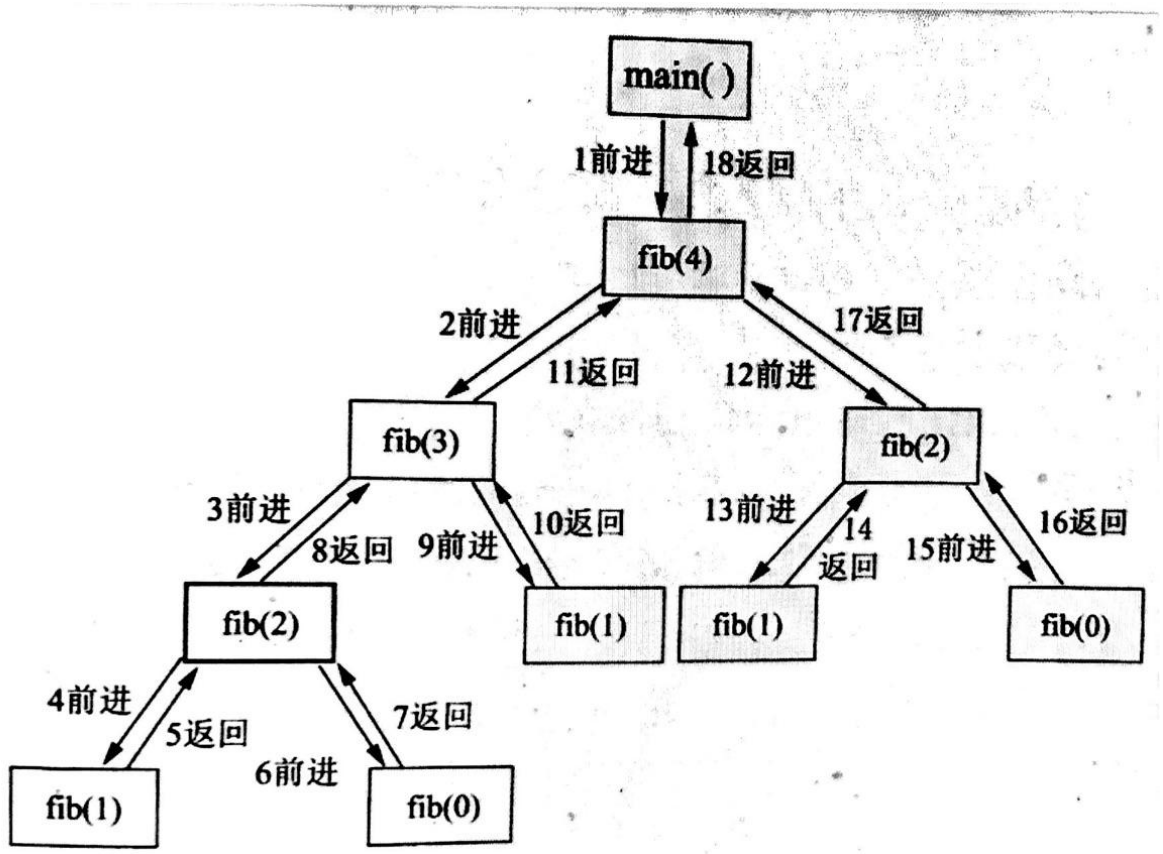
- 递归终止条件

fib(0)=0　　n=0

fib(1)=1　　n=1

斐波那契数列

```
#include <iostream>
using namespace std;
int fib(int n)
{
    if (n<=1) return n;
    return fib(n-1)+fib(n-2);
}
int main()
{
    int n;
    cin >>n;
    cout <<fib(n)<<endl;
    return 0;
}
```



最大公约数和最小公倍数

1、给你两个正整数x,y； 如何求它们的最大公约数？

2、最小公倍数呢？

最大公约数和最小公倍数

```
2  using namespace std;//最大公约数
3  int x,y;
4  int Gcd(int a,int b) {
5      if(a%b==0)return b;
6      return Gcd(b,a%b);
7  }
8  int main() {
9      cin>>x>>y;
10     cout<<Gcd(x,y)<<endl;
11     return 0;
12 }
```

最小公倍数＝(x*y)/Gcd(x,y)

例7、《上楼梯》

从楼下到楼上有N级台阶，小明上楼可以一次走1阶，也可以一次走2阶，请问小明一共有多少种不同的走台阶方式到达楼上的家？

输入一个整数n（n<20)，表示台阶总数，

输出一个整数x,表示不同的走台阶方式的总和。

A——C 共有几种走法？

例8、《汉诺塔》

递归思想（为降低题目难度，公开递归思想）：

> 将 X 杆上的 $n-1$ 个圆盘都移到空闲的 Z 杆上，并且满足上面的所有条件
> 将 X 杆上的第 n 个圆盘移到 Y 上
> 剩下问题就是将Z杆上的 $n-1$ 个圆盘移动到Y上了

【输入说明】

一行，盘子数量

【输出说明】

移动方案，见输出样例

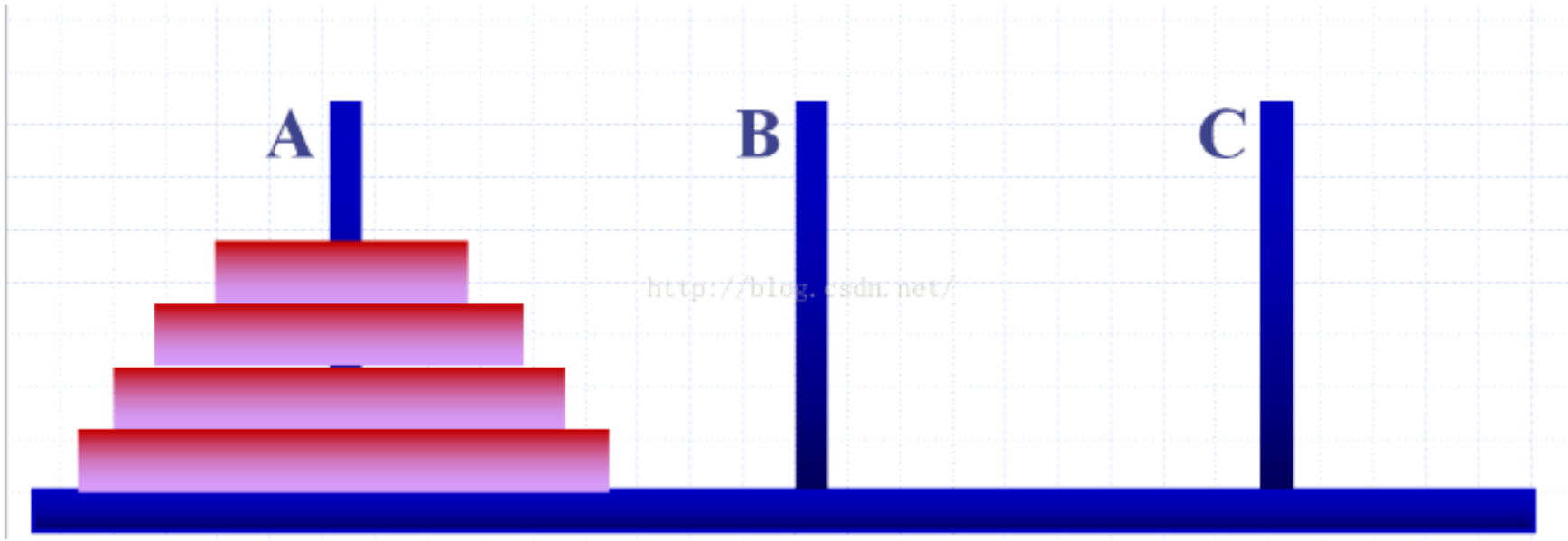
【输入样例】

```
3
```

【输出样例】

```
第1步，将1从X杆移动到Y杆
第2步，将2从X杆移动到Z杆
第3步，将1从Y杆移动到Z杆
```

例8、《汉诺塔》



分析：

1、什么情况下结束？

2、如何进行第一步？将n-1个盘子看做一个整体。

显然是先将n -1 个在a柱子上的盘子通过b柱移动到c柱上，

再将在a柱子上的编号为n的盘子移动到b柱上，

再将c柱子上的n-1个盘子通过a柱移动到b柱上，over

```
2  using namespace std;
3  int cnt=1;
4  void out(int k,int n,char a,char b) {
5      printf("第%d步，将%d从%c杆移动到%c杆",k,n,a,b);
6      cout<<endl;
7  }
8  void hnt(int n,char a,char b,char c) {
9      if(n==1) out(cnt++,n,a,c);
10     else {
11         hnt(n-1,a,c,b); //分为3步，2个子问题
12         out(cnt++,n,a,c);
13         hnt(n-1,b,a,c);
14     }
15 }
16 int main() {
17     int n;
18     cin>>n;
19     hnt(n,'X','Z','Y') ;
20     return 0;
```

阅读程序题：

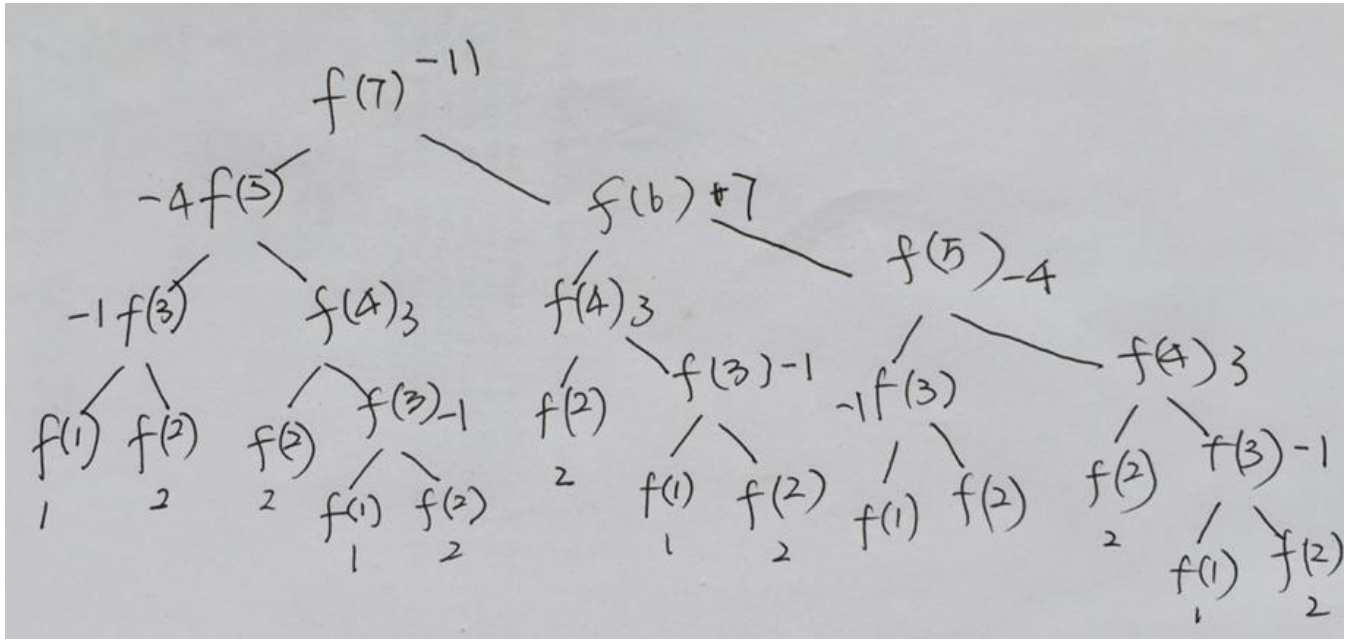
2014年普及组试题

2.
```
#include <iostream>
using namespace std;

int fun(int n) {
    if (n == 1)
        return 1;
    if (n == 2)
        return 2;
    return fun(n - 2) - fun(n - 1);
}

int main() {
    int n;
    cin >> n;
    cout << fun(n) << endl;
    return 0;
}
```

输入：7

输出：_____

2014年普及组试题



2.
```
#include <iostream>
using namespace std;

int fun(int n) {
    if (n == 1)
        return 1;
    if (n == 2)
        return 2;
    return fun(n - 2) - fun(n - 1);
}

int main() {
    int n;
    cin >> n;
    cout << fun(n) << endl;
    return 0;
}
```

输入：7

输出：_____

练习1：写出下列程序的运行结果。

```
int fun(int x){
    if((x==0)||(x==1))return 3;
     else
    return x-fun(x-2);
int main(){
  cout<<fun(15)<<endl;
end.
```

输出（　　）

练习2、写出下列程序的运行结果。

```
int f(int m,int n){
      if (m*n==0) return m+n+1
        else
      return f(m-1 , f(m,n-1));
       }
int main(){
   cin>>m>>n;
   cout<<f(m,n)<<endl;
   }
```

输入：2 1　　输出（　　）

练习3、写出下列程序的运行结果。

```
3  int  n=5;
4  char a[10];
5  void print(int  i) {
6      if(i==n)cout<<a[i];
7      else {
8          print(i+1);
9          cout<<a[i];
10     }
11 }
12 int main() {
13     gets(a); //abcdefghijklmn
14     print(0);
15     return 0;
16 }
```

9、第k个数

【描述】
输入一个正整数n，输出从右边数第k个数字。
【输入】
2个整数，分别是n和k。(n<10^9)
【输出】
一个一位数。
【输入样例】
394829 3
【输出样例】
8

第k个数

```
int n,m;
void datasin()
{
    cin >>n>>m;
}
int shu(int t,int k)
{
    k--;
    if (k==0) return t%10;
    else return shu(t/10,k);
}
int main()
{
    datasin();
    cout <<shu(n,m)<<endl;
    return 0;
}
```

```
using namespace std;
int n,k;
void findk(int x) {//方法2
    if(x==k) {
        cout<<n%10<<endl;
        return;
    }
    n=n/10;
    findk(x+1);
}
int main() {
    cin>>n>>k;
    findk(1);
    return 0;
}
```

Sin函数的嵌套

题目描述

Mas 在纸上写下了一个奇怪的函数:

$$f(n) = \sin(1 + \sin(2 + \sin(3 \ldots \sin(n) \ldots)))$$

输入格式

一个正整数 $n(1 \leq 1000)$

输出格式

输出这个函数

输出样例

```
5
```

输出样例

```
sin(1+sin(2+sin(3+sin(4+sin(5)))))
```

Sin函数的嵌套

```
 2  #define ll long long
 3  using namespace std;
 4  int n;
 5  void sin(int deep) {
 6      if(deep==n) {
 7          cout<<"sin("<<n<<")";
 8          return;
 9      }
10      cout<<"sin("<<deep<<"+";
11      sin(deep+1);
12      cout<<")";
13  }
14  int main() {
15      cin>>n;
16      sin(1);
17      return 0;
18  }
```

11、化简分数

- 将真分数化简为最简分数，例如：24/64化简为3/8。
- **【输入格式】**
- 输入两个整数n、m，分别是分子和分母，输入的数据保证为分子<分母，且n、m均小于等于30000。
- **【输出格式】**
- 输出化简后的分数。
- **【输入样例】**
- 24 64
- **【输出样例】**
- 3/8

化简分数

分析：

找到分子分母的最大公约数，化简即可