我想说:**只需记住下面三种语句、一个函数**，我们就可以使用它们构造算法及描述算法；要你们记的知识点是不是很少很少呢？

1. 首先我要让大家统一思想,把你们学过的程序语句归为下面三种伪代码语句,以后**作业要求首先用伪代码语句描述算法,然后用c语言把伪代码描述的算法改写成程序。**

1.1 顺序语句

input(x,a[5]); //输入语句; 这些伪代码语句大家一看就明白含义

print("hello",x,a[5]); //输出语句

x=5; //赋值语句

y=(x+4)\*6%7; //赋值语句

1.2 条件语句

if (条件) then 语句1 ; //简单条件语句

if (条件) then 语句1;

else 语句2; //复杂的条件语句 ; 注意 以后无论在哪碰到"语句1"、"语句2"、.....,要记住他们可以是复合语句；

{语句1；

语句2；

.....

语句n；} //表示复合语句

1.3 循环语句

1.3.1 for 循环语句

for i=1 to 15 do

print(i) ; //print语句是循环体，表示打印出：1 2 3 。。。。。15

1.3.2 while 循环语句：

i=1；

while (i<=15) do

{print(i);

i=i+1;} //功能同上面for循环，循环体用复合语句且采用缩进格式，含义明确

1.4 基本的操作

数学函数，关系运算，逻辑运算，算术运算在伪代码描述算法中可直接使用

理论上，有了上述三种形式的语句，可以描述一切算法；但为了可重用、易阅读及符合我们自顶向下逐步求精的思维习惯，我们的伪代码还使用函数。

2. 函数 //**尽量多用函数:** 函数是解决一类问题的抽象方法

类型 函数名（形参表）

{语句系列；}

3. 举例说明：

例1：

3.1 题目：任给两个正整数m,n,求其最大公约数

3.2 思路: 从2开始到min(m,n)为止的每个整数i,如i是m,n的公约数,则这些公约数之积为所求; 特别注意:每个i可能试多次,如对40,84 2试了两次都算！

3.3 伪代码描述的算法如下： //重在描述思路：忽略细节，聚焦算法；

Int gcd(m,n) //无需变量说明等细节

{k=min(m,n);

gys=1;

for i=2 to k do

{while (m ,n 均/都整除 i) do

{gys=gys\*i;

m整除i的结果-->m; //类似赋值语句的格式也可，语义不言自明吧

n整除i的结果-->n;} //当然要是基本操作或不重要的复杂的操作

}

Return(gys);

}

3.4 c程序： //把上面伪代码描述的算法用c语言实现了

#include "stdio.h"

int gcd(int m,int n)

{ int gys=1;int k,i;

if (m<n) k=m ; else k=n;

for (i=2; i<=k ; i++)

while ((m % i)==0 && (n%i==0))

{gys=gys\*i;

m=m/i;

n=n/i;}

return gys ;

}

void main()

{int m,n;

printf("pls input two zhengzhengshu ");

scanf("%d%d",&m,&n);

printf("%d is maximal gys \n",gcd(m,n));

}

例2：

3.1 题目：给定数组中的元素值，求从下标l开始到h为止的一系列元素中最小的两个值。

3.2 递归思路:如果只有一个或两个元素，结果显而易见；否则分为两部分，分别找出两个最小的，然后从两个最小的之中选。

3.3 伪代码描述的算法如下：

//忽略细节，聚焦算法； 是不是感觉算法更容易理解一些呢？

selectmymins(l,h,min1 ,min2)

//给定全局数组a，找出其中两个最小的且通过min1 ,min2返回给调用者

{if (l==h) {min1=a[l] ;min2=a[l];return(0);}

if (l==h-1) {if (a[l]<a[h]) {min1=a[l];min2=a[h];} else {min1=a[h];min2=a[l];}}

else {m=(l+h)/2;

selectmymins(l,m,min1,min2);lmin1=min1;lmin2=min2;

selectmymins(m+1,h,min1,min2);rmin1=min1;rmin2=min2;

mymin[1]=lmin1;mymin[2]=lmin2;mymin[3]=rmin1;mymin[4]=rmin2;

将mymin[1..4]按从小到大排序 //不重要，故一句话描述完毕(这里最好不要也不需要展开具体描述)

min1 = mymin[1];

min2 = mymin[2];

}

}

3.4 c程序

#include <stdio.h>

#include <stdlib.h>

int a[12]={3,9,23,10,1,6,11,2,15,19,17,4}; int mymin[5];

//这里直接给定了一个数组,可改写成运行时生成;但这不重要;重要的是找两个最小的.

selectmymins(int l,int h,int \* min1 ,int \* min2)

{ int m;int j;int lmin1,lmin2, rmin1,rmin2;

if (l==h) {\* min1=a[l] ;\* min1=a[l];return(0);} //is return not needed ?

if (l==h-1) {if (a[l]<a[h]) {\* min1=a[l];\* min2=a[h];} else {\* min1=a[h];\* min2=a[l];}}

else {m=(l+h)/2;

selectmymins(l,m,min1,min2);lmin1=\* min1;lmin2=\* min2;

selectmymins(m+1,h,min1,min2);rmin1=\* min1;rmin2=\* min2;

mymin[1]=lmin1;mymin[2]=lmin2;mymin[3]=rmin1;mymin[4]=rmin2;

for (int i=2;i<=4;i++)

{mymin[0]=mymin[i];j=i-1;

while (mymin[0]<mymin[j]) {mymin[j+1]=mymin[j];j=j-1;}

mymin[j+1]=mymin[0];}

\* min1 = mymin[1];

\* min2 = mymin[2];

}

}

int main()

{ int a,b; selectmymins(0,11,&a,&b);

printf("Hello world! %d %d \n",a,b);

return 0;

}