1.4试仿照三元组的抽象数据类型分别写出抽象数据类型复数和有理数的定义（有理数是其分子、分母均为自然数且分母不为零的分数）。

复数

ADT Complex{

数据对象：D={r,i|r,i为实数}

数据关系：R={<r,i>}

基本操作：

InitComplex(&C,re,im)

//操作结果：构造一个复数C，其实部和虚部分别为re和im

DestroyCmoplex(&C)

//操作结果：销毁复数C

Get(C,k,&e)

//操作结果：用e返回复数C的第k元的值

Put(&C,k,e)

//操作结果：改变复数C的第k元的值为e

IsAscending(C)

//操作结果：如果复数C的两个元素按升序排列，则返回1，否则返回0

IsDescending(C)

//操作结果：如果复数C的两个元素按降序排列，则返回1，否则返回0

Max(C,&e)

//操作结果：用e返回复数C的两个元素中值较大的一个

Min(C,&e)

//操作结果：用e返回复数C的两个元素中值较小的一个

}

有理数

ADT Complex ADT RationalNumber{

数据对象：D={s,m|s,m为自然数，且m不为0}

数据关系：R={<s,m>}

基本操作：

InitRationalNumber(&R,s,m)

//操作结果：构造一个有理数R，其分子和分母分别为s和m

DestroyRationalNumber(&R)

//操作结果：销毁有理数R

Get(R,k,&e)

//操作结果：用e返回有理数R的第k元的值

Put(&R,k,e)

//操作结果：改变有理数R的第k元的值为e

IsAscending(R)

//操作结果：若有理数R的两个元素按升序排列，则返回1，否则返回0

IsDescending(R)

//操作结果：若有理数R的两个元素按降序排列，则返回1，否则返回0

Max(R,&e)

//操作结果：用e返回有理数R的两个元素中值较大的一个

Min(R,&e)

//操作结果：用e返回有理数R的两个元素中值较小的一个

}ADT RationalNumber

1.17

#include <stdio.h>

void a(int k,int m)

{

if(m<k)

{

printf("输入错误");

return;

}

int f[100]={0},s=(m==k)?1:0;

f[k-1]=1;

for(int i=k;i<m;++i)

{

for(int j=i-k;j<i;++j)

f[i]+=f[j];

if(i>=m-k) s+=f[i];

}

printf("%d\n",s);

}



int main()

{

int k,m;

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

a(k,m);

return 0;

}

1.18

void student(Data\* p) //传入数据的头指针

{

double boy\_s[5]={0.0},girl\_s[5]={0.0},sum[5]={0.0};

while(p[i]不为空) //计算各院男女总分

{

switch(p[i]->校名)

{

case "A":

if(p[i]->性别=="男") boy\_s[0]+=p[i]->得分;

else girl\_s[0]+=p[i]->得分; break;

case "B":

if(p[i]->性别=="男") boy\_s[1]+=p[i]->得分;

else girl\_s[0]+=p[i]->得分; break;

case "C":

if(p[i]->性别=="男") boy\_s[2]+=p[i]->得分;

else girl\_s[0]+=p[i]->得分; break;

case "D":

if(p[i]->性别=="男") boy\_s[3]+=p[i]->得分;

else girl\_s[0]+=p[i]->得分; break;

case "E":

if(p[i]->性别=="男") boy\_s[4]+=p[i]->得分;

else girl\_s[0]+=p[i]->得分; break;

}

}

for(int i=0;i<5;++i) sum[i]=boy\_s[i]+girl\_s[i]; //计算各院团队总分

for(int i=0;i<5;++i) //输出

printf("男：%lf 女：%lf 总：%lf\n",boy\_s[i],girl\_s[i],sum[i]);

}