一：快速排序

#include<stdio.h>

int QKpass(int A[],int low,int high)

{

int x ;

x = A[low];

while(low<high)

{

while(low<high&&A[high]>=x)

high--;

if(low<high)

{

A[low] = A[high];

low++;

}

while(low<high&&A[low]<x)

low++;

if(low<high)

{

A[high] = A[low];

high--;

}

}

A[low] = x;

return low;

}

void QKsort(int A[],int low,int high)

{

int pos;

if(low<high){

pos = QKpass(A,low,high);

QKsort(A,low,pos-1);

QKsort(A,pos+1,high);

}

}

int main()

{

int N,i;

printf("输入数组长度：");

scanf("%d",&N);

int A[N];

printf("输入数组值：");

for(i = 0;i<N;i++){

scanf("%d",&A[i]);

}

QKsort(A,0,N-1);

for(i = 0;i<N;i++){

printf("%d",A[i]);

printf("\t");

}

return 0;

}

二：分别用递推和递归解决猴子吃桃问题

①递推实现：

#include <stdio.h>

#define N 100

int main()

{

int a[N],i;

a[10]=1;

for(i=10;i>=2;i--)

{

a[i-1]=2\*a[i]+2;

}

printf("%d",a[1]);

return 0;

}

②递归实现：

#include <stdio.h>

int chitao(int t)

{

if (t==10)

{

return 1;

}

else

三;四位数乘二等于五位数，九个数字互不相同

#include<stdio.h>

int main()

{

int x,y,i;

int m = 0;

int n = 0;int temp =0;

int A[4];

int B[5]; int f;

for(i = 5000;i<=9999;i++)

{

x = i;y = 2\*i;

if(y>=10000&&y<=99999)

{

if(x!=0){

A[m] = x%10;

x = x/10;

m++;

}

if(y!=0){

B[n] = y%10;

y = y/10;

n++;

}

}

int judge[10] = {0};

for(f = 0;f<4;f++){

judge[A[f]]++;

}

for(f = 0;f<5;f++){

judge[B[f]]++;

}

for(f = 0;f<10;f++){

if(judge[f]>1)

{

temp = 1;

break;

}

}

if(temp == 0)

printf("%d\t%d\n",i,i\*2);

}

return 0;

}

四;用分治法求数组中两个最大数和最小数

int a[100],n;

void quicksort(int left, int right) {

int i, j, t, temp;

if(left > right)

return;

temp = a[left];

i = left;

j = right;

while(i != j) {

while(a[j] >= temp && i < j)

j--;

while(a[i] <= temp && i < j)

i++;

if(i < j)

{

t = a[i];

a[i] = a[j];

a[j] = t;

}

}

a[left] = a[i];

a[i] = temp;

quicksort(left, i-1);

quicksort(i+1, right);

}

int main()

{

int i,n;

scanf("%d", &n);

for(i = 1; i <= n; i++)

scanf("%d", &a[i]);

quicksort(1, n);

printf("最小的数为：%d ", a[1]);

printf("较小的数为：%d ", a[2]);

printf("较大的数为：%d ", a[n-1]);

printf("最大的数为：%d ", a[n]);

return 0;

}