实验七

（1）

#include <stdio.h>

#define SIZE 20

int len=5;

//选择法排序，由大到小次序

void sort(double a[],int n ){

int i,j,maxLoc;

double temp;

for (i=0;i<n-1;i++){

maxLoc=i;

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

if (a[j]>a[i])

maxLoc=j;

if (maxLoc!=i) {

temp=a[j] ; a[j]=a[i] ;a[i]=temp ;

}

}

}//向有序数组中插入一个元素

void insertElement(double \*a,int n,double element)

{

int i=0,j;

//寻找插入位置i

while (a[i++]>element && i<n);

if (i==n)

{ //插入在数组的最后

a[i+1]=element;

len++;

}

else

{

i--;

//从第i个元素起往后移一个位置，使第i个位置空出来

for (j=n-1;j>=i;j--)

a[j+1]=a[j];

a[i]=element; len++;//第i个位置放入element

}

}

//显示数组的所有元素

void printList(double a[],int n)

{

int i;

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

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

printf("\n");

}

void main()

{

double arr[SIZE] ;

int i;

double ele;

printf("Input %d elements:\n",len);

for (i = 0; i < len; i++ )

scanf("%lf",&arr[i]);

printf("排序前的数组:\n");

printList(arr,len);

sort(arr,len); //排序数组arr

printf("降序排序后的数组：" );

printList(arr,len);

printf("\nInput a element to be insered:\n");

scanf("%lf",&ele);

insertElement(&arr,len,ele); //插入ele到输入arr中

printList(arr,len);

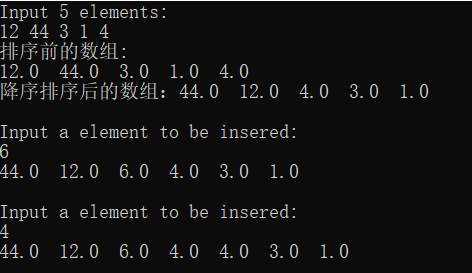
printf("\nInput a element to be insered:\n");

scanf("%lf",&ele);

insertElement(&arr,len,ele);//入元素到有序数组中

printList(arr,len);

}



(2)

#include <stdio.h>

#include<math.h>

int f(int a)

{//返回1是素数，返回0不是素数；

int i,sqrta;

sqrta=(int)sqrt(a);

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

if(a%i==0)

break;

if(i>sqrta)

return 1;

else return 0;

}

int fun(int lim,int \*aa)

{

int m,i,j=0,x;

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

{

x=f(i);

if(x)

{

aa[j]=i;

j++;

}

}

for(m=0;m<=j;m++)

printf("%-5d",aa[m]);

printf("\n");

}

void main()

{

int lim;

scanf("%d",&lim);

int a[lim];

fun(lim,a);

}



(3)

#include<stdio.h>

#define M 4

#define N 4

int fun(int a[][N])

{ int i,j,s=0;

//求第10行和最后一行的元素之和

for(j=0;j<N;j++)

{ s+=a[0][j];

s+=a[M-1][j];

}

//求其他边角元素之和

for(i=1;i<=M-2;i++)

{ s+=a[i][0];

s+=a[i][N-1];

}

return s;

}

void main()

{

int a[M][N]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16};

int i,j,s;

s=fun(a);

printf("二维数组为\n",s);

for (i=0;i<M;i++)

{

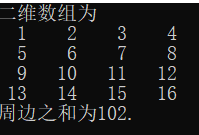
for (j=0;j<N;j++)

printf("%3d ",a[i][j]);

printf("\n");

}

printf("周边之和为%d.\n",s);

}

(4)

#include<stdio.h>

#include<stdlib.h>

void fun(int m, int \*a, int \*n)

{

int i,j=0; \*n=0;

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

if (i%7==0 || i%11 ==0)

{ a[j]=i;

j++;

}

\*n=j;

}

main()

{

int m,n,i;

int\* p;

scanf("%d",&m);

p=malloc(m); //给P动态分配m个int型数据空间

fun(m,p,&n);

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

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

printf("\n");

free(p);

}

6.1

（4）

A．

#include <stdio.h>

#include <stdlib.h>

#include<math.h>

#define N 100

int f(int a)

{

int i,sqrta;

sqrta=(int)sqrt(a);

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

{

if(a%i==0)

break;

}

if(i>sqrta)

return 1;

else return 0;

}

void main()

{

int i,count=0,x,array[50];

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

{

x=f(2\*i+1);

if(x)

{

printf("(%-4d,%-4d)",i,i+1);

count+=1;

if(count%6==0&&count!=0)

printf("\n");

}

}

}