班级：计算机173 姓名：张博翔 学号：201706060334

1. 程序改错题
2. 从键盘输入十个学生的成绩，统计最高分，最低分和平均分。max 代表最高 分，min 代表最低分，avg 代表平均分

#include <stdio.h>

int main()

{

int i;

float a[10],min,max,avg;

printf("input 10 score:");

for(i=0;i<=9;i++)

{

printf("input a score of student:");

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

}

max=min=avg=a[0];

for(i=1;i<=9;i++)

{

if(min>a[i])

min=a[i];

if(max<a[i])

max=a[i];

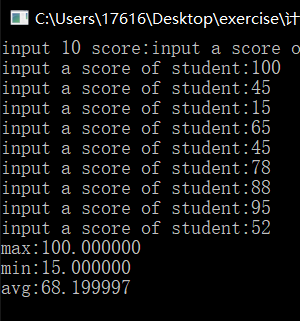
avg=avg+a[i];

}

avg=avg/10;

printf("max:%f\nmin:%f\navg:%f\n",max,min,avg);

}



1. 功能：在一个已按升序排列的数组中插入一个数，插入后，数组元素仍按升 序排列。

#include <stdio.h>

#define N 11

int main()

{

int i,number,a[N]={1,2,4,6,8,9,12,15,149,156};

printf("please enter an integer to insert in the array:\n");

scanf("%d",&number);

printf("The original array:\n");

for(i=0;i<N-1;i++)

printf("%5d",a[i]);

printf("\n");

for(i=N-2;i>=0;i--)

if(number<=a[i])

a[i + 1]=a[i];

else

{

a[i+1]=number;

break;

}

if(number<a[0])

a[0]=number;

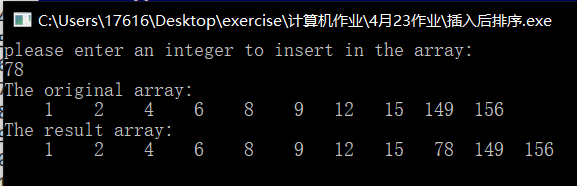
printf("The result array:\n");

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

printf("%5d",a[i]);

printf("\n");

}



1. 功能：给定 n 个实数，输出平均值，并统计在平均值以下（含平均值）的实 数个数。 例如：n=6时，输入23.5, 45.67, 12.1, 6.4，58.9, 98.4所得平均值为40.828335, 在平均值以下的实数个数应为 3。

#include <stdio.h>

int fun(double x[],int n)

{

int j,c=0;

float xa=0;

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

xa+=x[j];

xa=xa/n;

printf("ave=%f\n",xa);

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

if(x[j]<=xa)

c++;

return c;

}

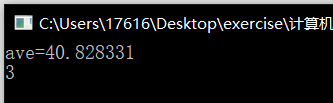
int main()

{

double x[]={23.5,45.67,12.1,6.4,58.9,98.4};

printf("%d\n",fun(x,6));

}



1. 下列给定程序中函数 fun 的功能是：用选择法对数组中的 n 个元素进行升序排 列。

#include <stdio.h>

#define N 20

void fun(int a[], int n)

{

int i, j, t, p;

for (j = 0 ;j < n-1;j++)

{

p = j;

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

if(a[i] < a[p])

p = i;

t = a[p];

a[p] = a[j];

a[j] = t;

}

}

int main()

{

int a[N]={9,6,8,3,-1},i, m = 5;

printf("排序前的数据:");

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

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

printf("\n");

fun(a,m);

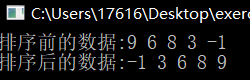
printf("排序后的数据:");

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

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

printf("\n");

}



1. 填空
2. 下面函数为二分法查找 key 值。数组中元素已递增排序，若找到 key 则返回 对应的下标，否则返回-1。

#include <stdio.h>

int fun(int a[],int n,int key)

{

int low,high,mid;

low=0;

high=n-1;

while(mid <= high)

{

mid=(low+high)/2;

if(key<a[mid])

high = mid - 1;

else if(key>a[mid])

low = mid + 1;

else

return mid - 1;

}

return -1;

}

int main()

{

int a[10]={1,2,3,4,5,6,7,8,9,10};

int b,c;

b=4;

c=fun(a,10,b);

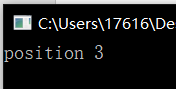
if(c==1)

printf("not found");

else

printf("position %d\n",c);

}



1. 下列给定程序中，函数 fun 的功能是：逆置数组元素中的值。 例如：若 a 所指数组中的数据为：1、2、3、4、5、6、7、8、9，则逆置后依 次为：9、8、7、6、5、4、3、2、1。形参 n 给出数组中数据的个数。

#include <stdio.h>

void fun(int a[], int n)

{

int i,t;

for (i=0; i<n / 2; i++)

{

t=a[i];

a[i] = a[n - 1- i];

a[n - 1 - i]= t;

}

}

int main()

{

int b[9]={1,2,3,4,5,6,7,8,9}, i;

printf("\nThe original data :\n");

for (i=0; i<9; i++)

printf("%4d ", b[i]);

printf("\n"); fun(b, 9);

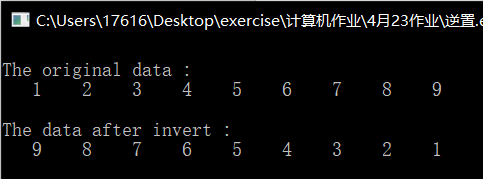
printf("\nThe data after invert :\n");

for (i=0; i<9; i++)

printf("%4d ", b[i]);

printf("\n");

}



3、.以下程序的功能是：采用二分法在给定的有序数组中查找用户输入的值，并显 示查找结果。请填写适当的符号或语句，使程序实现其功能。 如输入 5 输出为：找到位置为 6 如输入 10 输出为：10 未找到

#include <stdio.h>

#define N 10

int main()

{

int a[]={0,1,2,3,4,5,6,7,8,9},k;

int low=0,high=N-1,mid,find=0;

printf("请输入欲查找的值：\n");

scanf("%d",&k);

while (low<=high)

{

mid=(low + high) / 2;

if(a[mid]==k)

{

printf("找到位置为：%d\n",mid+1);

find=1;

}

if(a[mid] < k)

low = mid + 1;

else

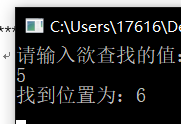
high = mid - 1;

}

if(!find)

printf("%d 未找到\n",k);

}



1. 程序设计

1、

#include <stdio.h>

double fun()

{

double min = 11, max = -1, num, sum = 0;

for(int i = 0; i < 10; i++)

{

do

{

if(num < 0 || num > 10)

{

printf("输入错误请重新输入：");

}

printf("请输入第%d个成绩！",i + 1);

scanf("%lf",&num);

}while(num < 0 || num > 10);

if(num > max)

{

max = num;

}

if(num < min)

{

min = num;

}

sum += num;

}

sum -= max + min;

return sum / 8;

}

int main()

{

printf("最终成绩为：%lf",fun());

}



3、输出斐波那契数列的前 40 项，一行输出 4 个数。

#include <stdio.h>

void fun()

{

int num1 = 1, num2 = 1,num3;

printf("%-20d%-20d",num1,num2);

for(int i = 3; i <= 40; i++)

{

num3 = num1 + num2;

printf("%-20d",num3);

num1 = num2;

num2 = num3;

if(i % 4 == 0)

printf("\n");

}

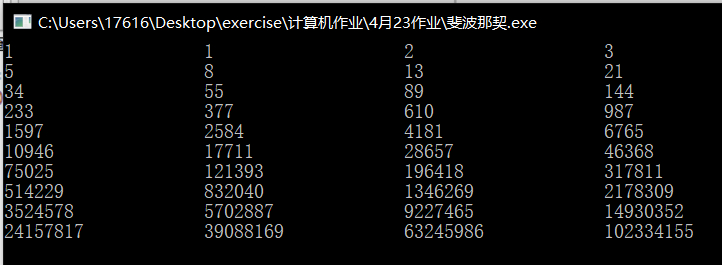
}

int main()

{

fun();

}



1. 将两个有序序列合并为一个有序序列。

#include <stdio.h>

#define M 5

#define N 6

int main()

{

int A[M] = {1, 3, 5, 6, 8}, B[N] = {1, 2, 5, 7, 8, 9}, C[M + N];

int i = 0, j = 0, k = 0;

while(i <= M && j <= N)

{

if(A[i] < B[j])

{

C[k] = A[i];

k++;

i++;

}

else

{

C[k] = B[j];

k++;

j++;

}

if(i <= M && j > N)

{

for(; k < M + N; k++)

{

C[k] = A[i];

i++;

}

}

if(j <= N && i > M)

{

for(; k < M + N; k++)

{

C[k] = B[j];

j++;

}

}

}

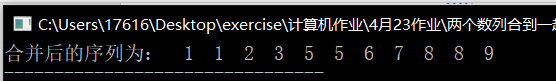
printf("合并后的序列为：");

for (k = 0; k < M + N; k++)

printf("%3d", C[k]);

return 0;

}



4、.输入任意的一个数，用顺序查找法查找该数是否在一个数组中，并计算查找次数。 写出选择排序和顺序查找的算法思想。

#include <stdio.h>

int main()

{

//顺序查找法：利用循环从数组里一个一个依次查找，找到对应的数则跳出循环

//选择排序 :在一个数组里从第一个数字开始，利用循环寻找之后数字比他更小的数字，

//找到后进行交换；直到进行外循环到倒数第二个数字为止

int a[10] = {7,8,9,55,44,12,45,65,32,84}, num, i;

printf("请输入要查找的数：");

scanf("%d",&num);

for(i = 0; i < 10; i++)

{

if(a[i] == num)

{

printf("找到了！下标为：%d，查找次数为：%d",i,i+1);

break;

}

}

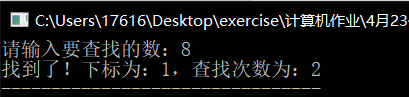
if(i >= 10)

{

printf("没找到！");

}

}



5、.输入任意的一个数，用二分法查找该数是否在一个排好序的数组中，并计算查找 次数。写出冒泡排序和二分查找的算法思想。

#include <stdio.h>

#define N 10

int main()

{

int a[N] = {7,8,9,54,25,65,84,66,88,10}, t, num, low = 0, high = 9, mid, find = 0;

//冒泡排序：外循环次数等于元素个数-1,内循环每一轮为元素个数-当前的轮数-1，每次循环都会把最小的数向前进行交换

for(int i = 0; i < N - 1; i++)

{

for(int j = 0; j < N - i - 1; j++)

{

if(a[j] > a[j + 1])

{

t = a[j];

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

a[j + 1] = t;

}

}

}

//二分法查找，从排序好的数组的中间一个数开始比较，一步步缩小范围，如果中间数比查找数小，则锁定数组的后半部分，锁定前半部分

printf("请输入要查找的数：",&num);

scanf("%d",&num);

while(low <= high)

{

mid = (low + high) / 2;

if(a[mid] == num)

{

printf("找到了,下标是%d\n",mid);

find = 1;

}

if(a[mid] < num)

{

low = mid + 1;

}

else

{

high = mid - 1;

}

}

if(find == 0)

{

printf("未找到！");

}

}

