

**实验十二**

**指针+一维数组**

学期：2024-2025 第一学期

编制日期：2024 年 11月 28 日

编制人：江家玮

学号：22281188

班级：计科2204

**实验十二**

**知识覆盖：指针变量基础，指针与一维数值数组**

**1． 编写程序，完成如下要求**

1. 定义类型为char, int, float, double, char \*, int \*, float \*, double \*的变量各一个，给每个变量设定或输入值。
2. 大致按如下格式要求输出所有变量的各项属性（变量名，类型，占用字节数，地址，变量值）：

VarName Type Size Address Value

nVar1 int 4 0012FF50H 10

pInt int \* 4 0012FF3CH 0012FF80H

…

提示：浮点数采用%10.2格式，获取变量占用字节数用sizeof(变量名)，用 &变量名 取变量地址，变量地址以及指针变量中的保存的地址值要求以8位16进制形式输出（后面加H），如上表所示。

1. 观察前后变量的地址值之间的关系，并在实验报告中总结所有变量地址安排规律。
2. 比较各个指针变量所占用存储空间的大小，并在实验报告中简要说明比较结果的产生原因。

#include <stdio.h>

int main() {

// 定义基本类型的变量

char c;

int i;

float f;

double d;

// 定义指针类型的变量

char\* pc;

int\* pi;

float\* pf;

double\* pd;

// 为基本类型的变量赋值或输入值

printf("请输入一个字符: ");

scanf\_s(" %c", &c); // 注意前面的空格，用于跳过换行符

printf("请输入一个整数: ");

scanf\_s("%d", &i);

printf("请输入一个浮点数: ");

scanf\_s("%f", &f);

printf("请输入一个双精度浮点数: ");

scanf\_s("%lf", &d);

// 为指针类型的变量赋值

pc = &c; // 指向字符变量的地址

pi = &i; // 指向整数变量的地址

pf = &f; // 指向浮点数变量的地址

pd = &d; // 指向双精度浮点数变量的地址

// 输出表头

printf("\nVarName\t\tType\t\tSize\t\tAddress\t\t\tValue\n");

printf("-------\t\t----\t\t----\t\t--------\t\t-----\n");

// 输出基本类型变量的属性

printf("c\t\tchar\t\t%lu\t\t%pH\t\t%c\n", sizeof(c), &c, c);

printf("i\t\tint\t\t%lu\t\t%pH\t\t%d\n", sizeof(i), &i, i);

printf("f\t\tfloat\t\t%lu\t\t%pH\t\t%10.2f\n", sizeof(f), &f, f);

printf("d\t\tdouble\t\t%lu\t\t%pH\t\t%10.2lf\n", sizeof(d), &d, d);

// 输出指针类型变量的属性

printf("pc\t\tchar\*\t\t%lu\t\t%pH\t\t%pH\n", sizeof(pc), &pc, pc);

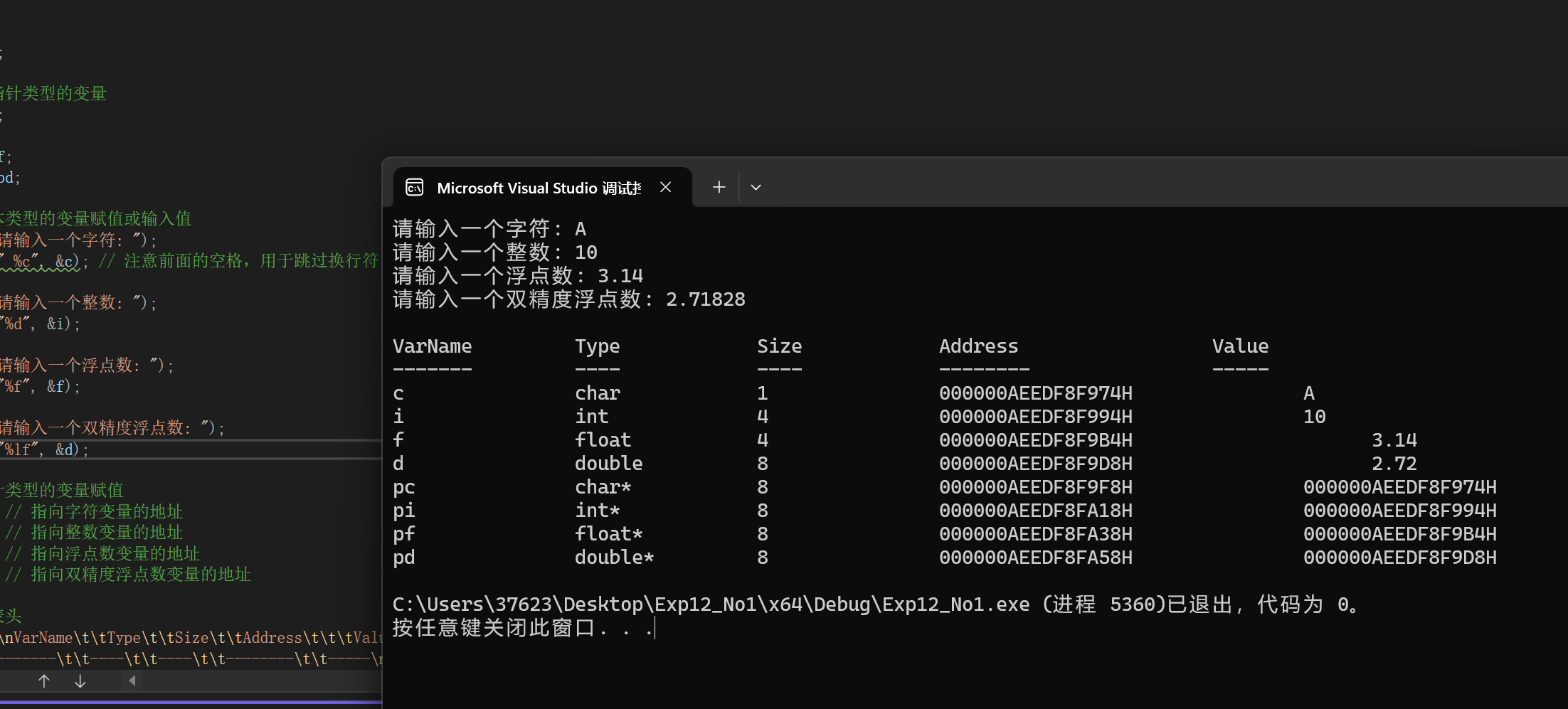
printf("pi\t\tint\*\t\t%lu\t\t%pH\t\t%pH\n", sizeof(pi), &pi, pi);

printf("pf\t\tfloat\*\t\t%lu\t\t%pH\t\t%pH\n", sizeof(pf), &pf, pf);

printf("pd\t\tdouble\*\t\t%lu\t\t%pH\t\t%pH\n", sizeof(pd), &pd, pd);

return 0;

}



原因：

指针变量占用存储空间的大小：在 32 位系统中，所有指针变量的大小都是 4 字节。在 64 位系统中，所有指针变量的大小都是 8 字节。所以显示的size是8

1. **用指针做函数参数，完成两个数交换，并在主程序中测试。**

#include <stdio.h>

// 交换两个数的函数，使用指针作为参数

void swap(int\* a, int\* b) {

int temp;

temp = \*a;

\*a = \*b;

\*b = temp;

}

int main() {

int num1, num2;

// 输入两个数

printf("请输入第一个数: ");

scanf\_s("%d", &num1);

printf("请输入第二个数: ");

scanf\_s("%d", &num2);

// 输出交换前的值

printf("\n交换前的值:\n");

printf("num1 = %d, num2 = %d\n", num1, num2);

// 调用swap函数交换两个数的值

swap(&num1, &num2);

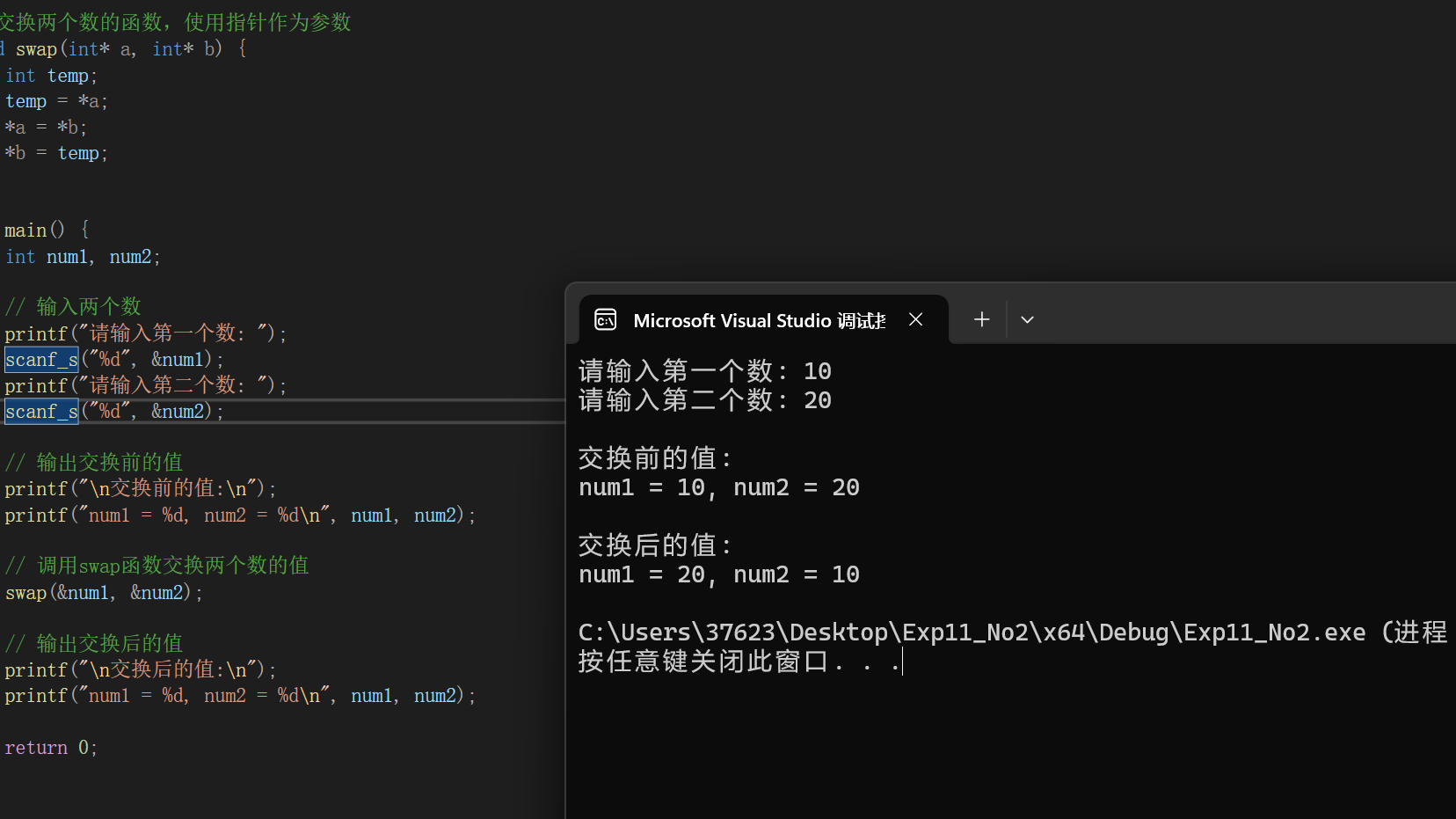
// 输出交换后的值

printf("\n交换后的值:\n");

printf("num1 = %d, num2 = %d\n", num1, num2);

return 0;

}



1. **输入10个整数，将其中最小的数与第一个数对换，把最大的数与最后一个数对换。请分别用数组写法和指针法完成上述功能。**

函数原型分别为：void action1(int dArray[],int n)和void action2(int \*pInt,int n)。

#include <stdio.h>

// 使用数组实现

void action1(int dArray[], int n) {

int minIndex = 0, maxIndex = 0;

// 找到最小值和最大值的索引

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

if (dArray[i] < dArray[minIndex]) {

minIndex = i;

}

if (dArray[i] > dArray[maxIndex]) {

maxIndex = i;

}

}

// 交换最小值和第一个数

int temp = dArray[0];

dArray[0] = dArray[minIndex];

dArray[minIndex] = temp;

// 交换最大值和最后一个数

temp = dArray[n - 1];

dArray[n - 1] = dArray[maxIndex];

dArray[maxIndex] = temp;

}

// 使用指针实现

void action2(int\* pInt, int n) {

int\* minPtr = pInt, \* maxPtr = pInt;

// 找到最小值和最大值的指针

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

if (\*(pInt + i) < \*minPtr) {

minPtr = pInt + i;

}

if (\*(pInt + i) > \*maxPtr) {

maxPtr = pInt + i;

}

}

// 交换最小值和第一个数

int temp = \*pInt;

\*pInt = \*minPtr;

\*minPtr = temp;

// 交换最大值和最后一个数

temp = \*(pInt + n - 1);

\*(pInt + n - 1) = \*maxPtr;

\*maxPtr = temp;

}

int main() {

int dArray[10];

// 输入10个整数

printf("请输入10个整数:\n");

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

scanf\_s("%d", &dArray[i]);

}

// 使用数组实现

action1(dArray, 10);

printf("\n使用数组实现后的结果:\n");

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

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

}

printf("\n");

// 恢复原始数组

printf("\n请再次输入10个整数:\n");

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

scanf\_s("%d", &dArray[i]);

}

// 使用指针实现

action2(dArray, 10);

printf("\n使用指针实现后的结果:\n");

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

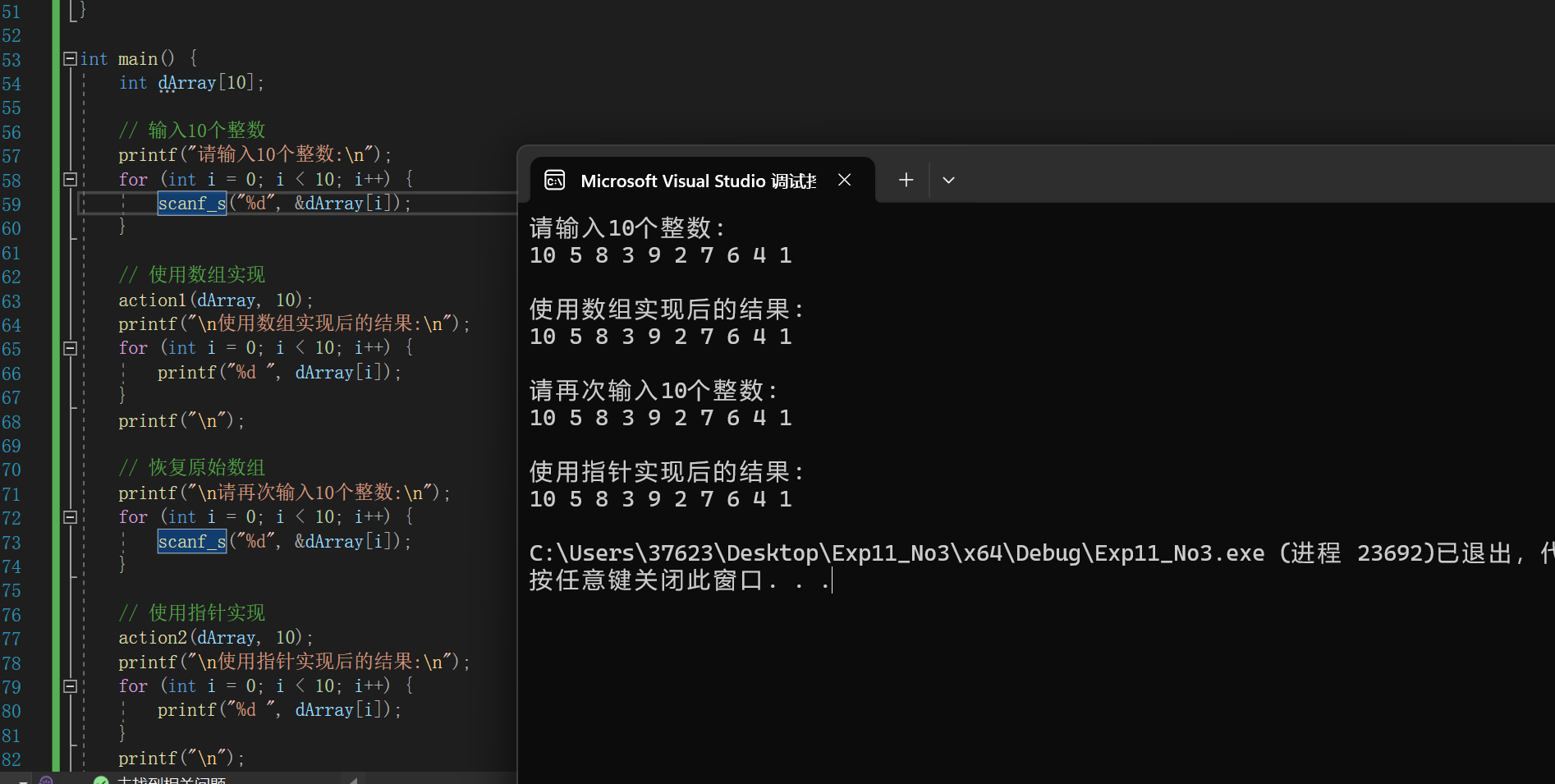
printf("%d ", dArray[i]);

}

printf("\n");

return 0;

}



1. **写函数，它将数组b中大于等于x的数顺序复制到数组a中，其中n为两个数组的大小。请分别用数组写法和指针写法完成上述功能。**

函数原型分别为分别为select1(double a[], double b[], int n, double x)和select2(double \*a, double \*b, int n, double x)。

#include <stdio.h>

// 使用数组实现

void select1(double a[], double b[], int n, double x) {

int index = 0; // 用于记录数组a的当前索引

// 遍历数组b，将大于等于x的数复制到数组a中

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

if (b[i] >= x) {

a[index++] = b[i];

}

}

// 将数组a的剩余部分填充为0（可选）

for (int i = index; i < n; i++) {

a[i] = 0;

}

}

// 使用指针实现

void select2(double\* a, double\* b, int n, double x) {

double\* pA = a; // 指向数组a的指针

double\* pB = b; // 指向数组b的指针

// 遍历数组b，将大于等于x的数复制到数组a中

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

if (\*pB >= x) {

\*pA = \*pB;

pA++;

}

pB++;

}

// 将数组a的剩余部分填充为0（可选）

while (pA < a + n) {

\*pA = 0;

pA++;

}

}

int main() {

double b[10] = { 1.5, 3.2, 4.8, 2.1, 5.0, 3.7, 6.4, 2.9, 4.5, 1.0 };

double a1[10] = { 0 }; // 用于存储数组实现的结果

double a2[10] = { 0 }; // 用于存储指针实现的结果

int n = 10;

double x = 3.5;

// 使用数组实现

select1(a1, b, n, x);

printf("使用数组实现的结果:\n");

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

printf("%.2f ", a1[i]);

}

printf("\n");

// 使用指针实现

select2(a2, b, n, x);

printf("使用指针实现的结果:\n");

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

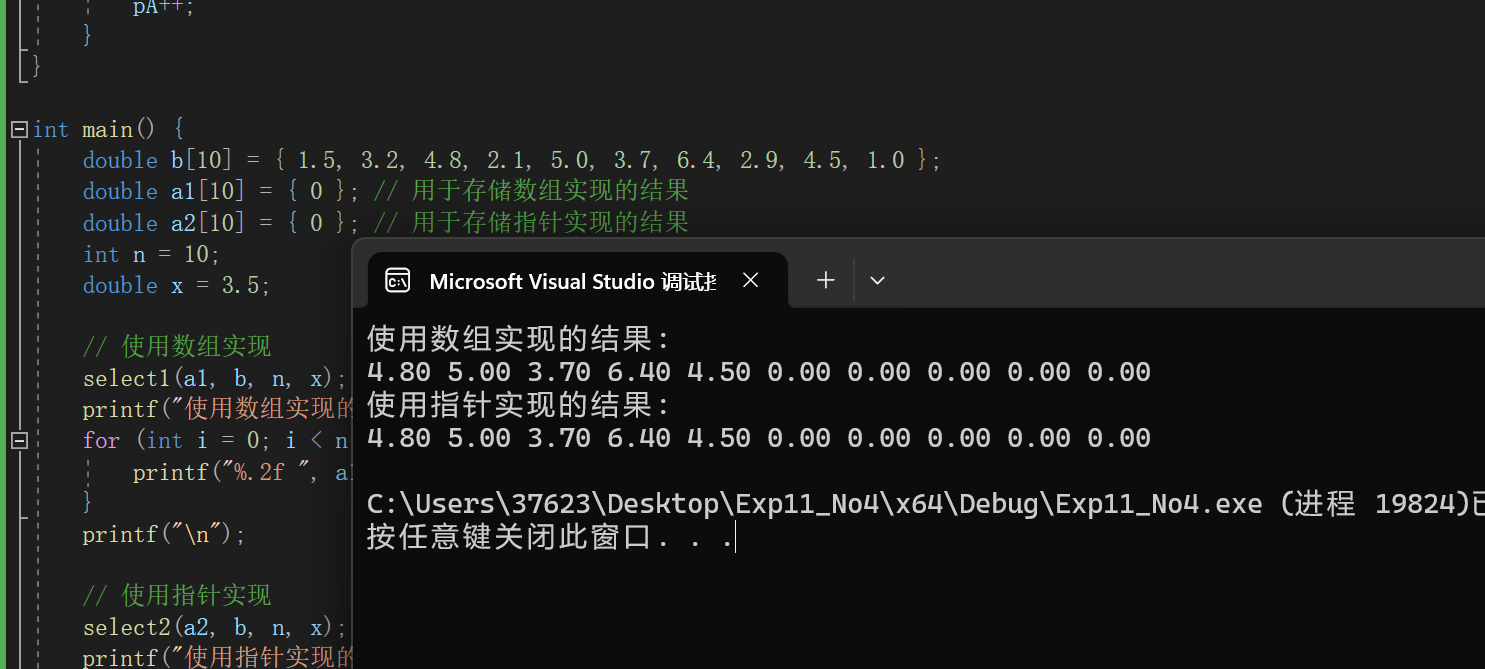
printf("%.2f ", a2[i]);

}

printf("\n");

return 0;

}



**5. 从键盘输入10个排好序的整数，用折半查找法查找某个数是否出现。**

函数原型为：

int search(int \*pArray, int n, int x)；//若找到函数返回值为1，否则返回为0。

在主程序中测试这个函数。

#include <stdio.h>

// 折半查找函数

int search(int\* pArray, int n, int x) {

int low = 0, high = n - 1;

// 二分查找

while (low <= high) {

int mid = (low + high) / 2;

if (pArray[mid] == x) {

return 1; // 找到目标数，返回1

}

else if (pArray[mid] < x) {

low = mid + 1; // 目标数在右半部分

}

else {

high = mid - 1; // 目标数在左半部分

}

}

return 0; // 未找到目标数，返回0

}

int main() {

int sortedArray[10]; // 已排序的数组

int x; // 待查找的数

// 输入10个已排序的整数

printf("请输入10个已排序的整数（从小到大）:\n");

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

scanf\_s("%d", &sortedArray[i]);

}

// 输入待查找的数

printf("请输入待查找的数: ");

scanf\_s("%d", &x);

// 调用search函数进行查找

int result = search(sortedArray, 10, x);

// 输出结果

if (result == 1) {

printf("数 %d 存在于数组中。\n", x);

}

else {

printf("数 %d 不存在于数组中。\n", x);

}

return 0;

}

