**实验六 指针（一）**

1. **目的要求**
2. 掌握指针的定义和使用指针变量；
3. 学会使用字符串的指针和指向数组的指针变量；
4. 学会使用指针作为函数参数；
5. 学会查看内存，观察变量/数组指针的变化；
6. **实验内容与要求**

**1、**

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

main()

{

char xCh, str[50];

int i, n, flag;

int f(char[], char, int);

flag = 1;

while (flag) //循环确保可以重复查询

{

system("cls");

i = 0;

/\*输入时使用\_getch()函数获取xCh避免错误\*/

printf("请在键盘按下待查字符xCh\n");

xCh = \_getch();

printf("%c\n", xCh);

printf("请输入被查的字符串str:");

scanf("%s", str);

n = f(str, xCh, i);

printf("\nstr中xCh出现的次数为%d\n", n);

/\*通过改变flag的值来控制循环查询\*/

printf("\n是否继续查询？（输入1或0）");

scanf("%d", &flag);

/\*此段查询输入flag的合法性\*/

if (flag != 1 && flag != 0)

{

printf("\n是否继续查询？（输入1或0）");

scanf("%d", &flag);

}

printf("\n");

}

}

/\*下面编写一个查找xCh的函数\*/

int f(char str[], char xCh, int i)

{

int n = 0;

char\* p;

printf("xCh出现的位置有:\n");

for (p = str; \*p != 0; p++)

{

if (\*p == xCh)

{

n++;

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

}

i++;//指针的每次移动也带动xCh的位置移动

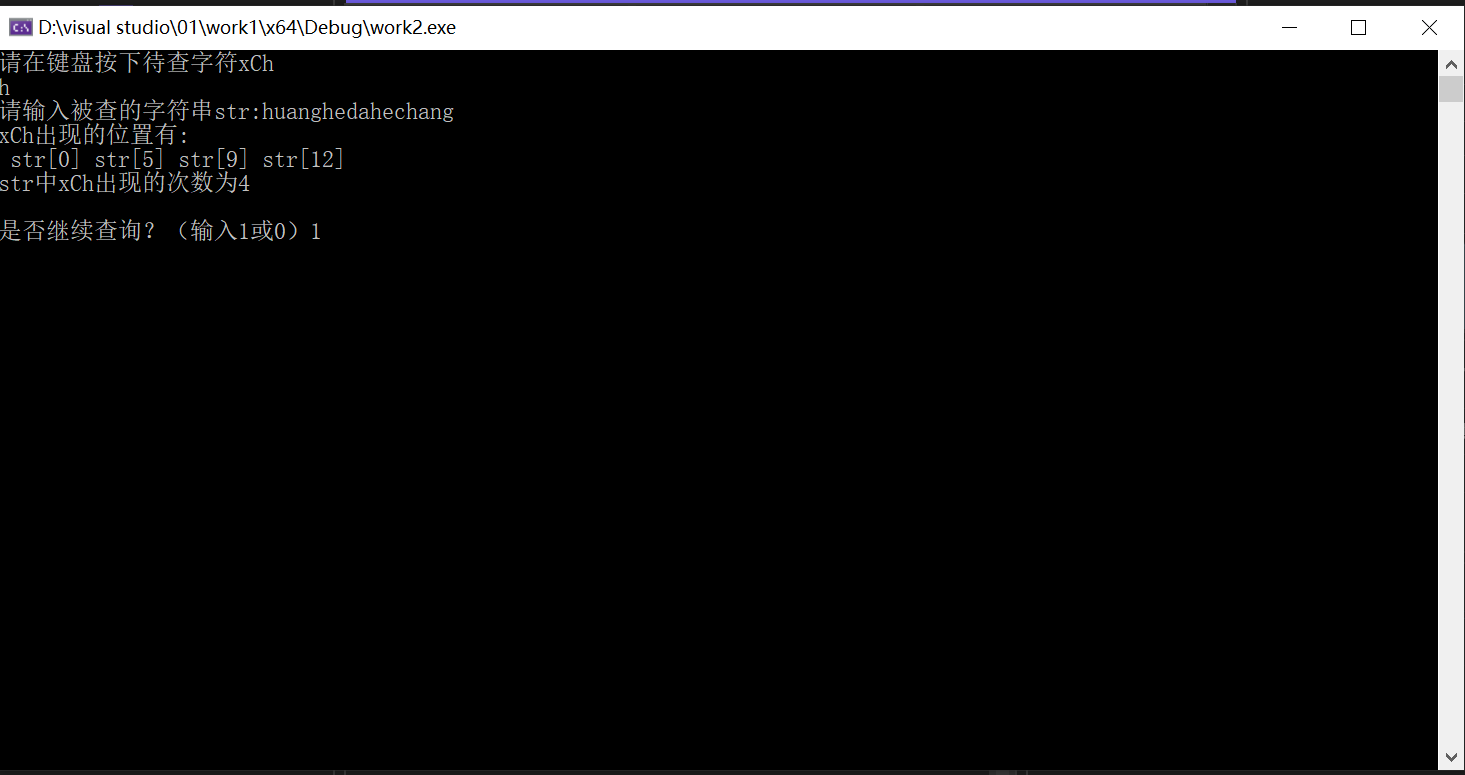
}

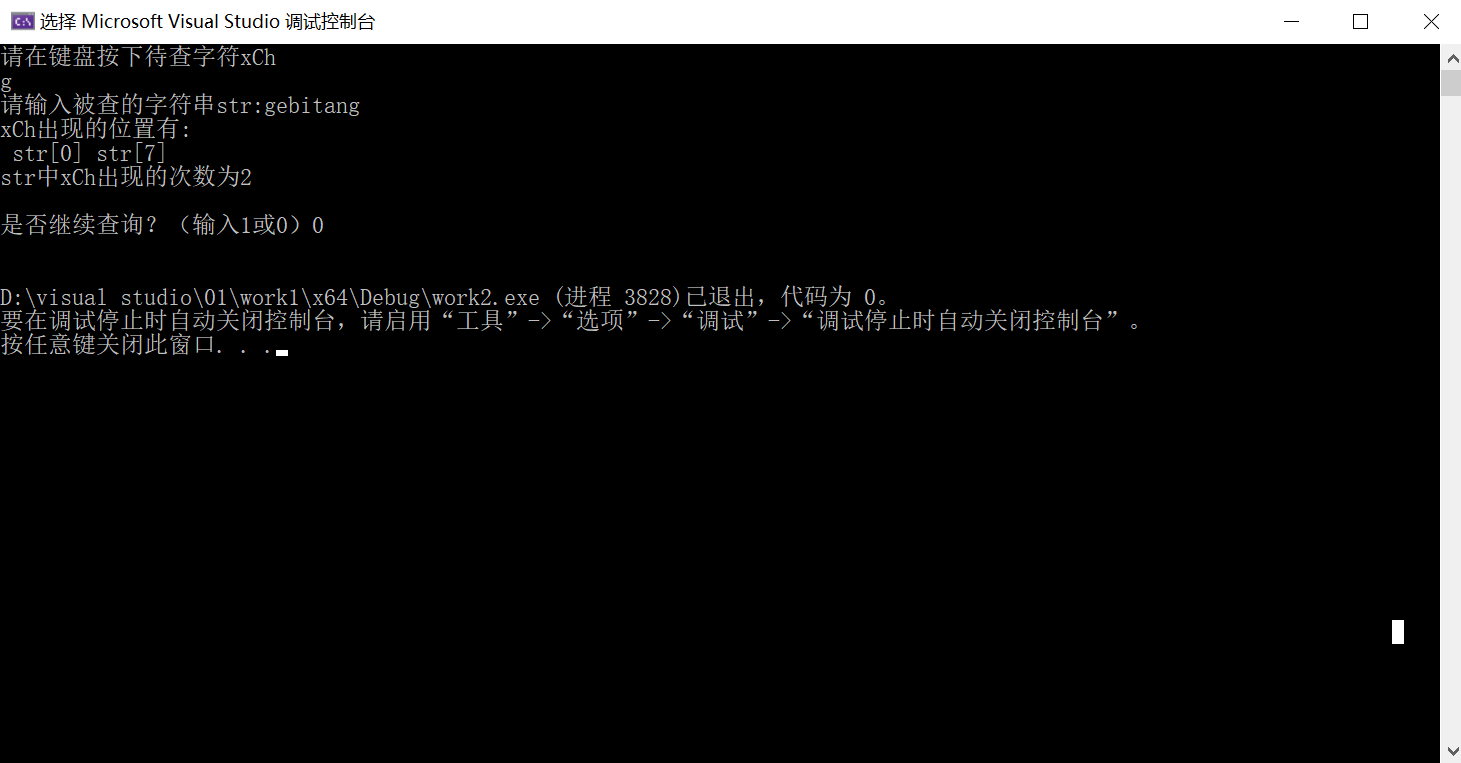
if (n == 0)

printf("无！\nstr中无xCh!");

return (n);

}





**2、**

**用普通指针，如下：**

#include<stdio.h>

main()

{

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

int\* p;

printf("二维数组各元素值如下：\n");

for (p = a[0]; p < a[0] + 9; p++)

{

if ((p - a[0]) % 3 == 0)

printf("\n");

printf("%15d", \*p);

}

printf("\n二维数组各元素地址如下：\n");

for (p = a[0]; p < a[0] + 9; p++)

{

if ((p - a[0]) % 3 == 0)

printf("\n");

printf("%15u", p);

}

}

**用行指针，如下：**

#include<stdio.h>

main()

{

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

int \*q, (\*p)[3];

printf("二维数组各元素值如下：\n");

for (p = a; p < a+3; p++)

{

for (q = p[0]; q < p[0] + 3; q++)

printf("%15d", \*q);

printf("\n");

}

printf("\n二维数组各元素地址如下：\n");

for (p = a; p < a + 3; p++)

{

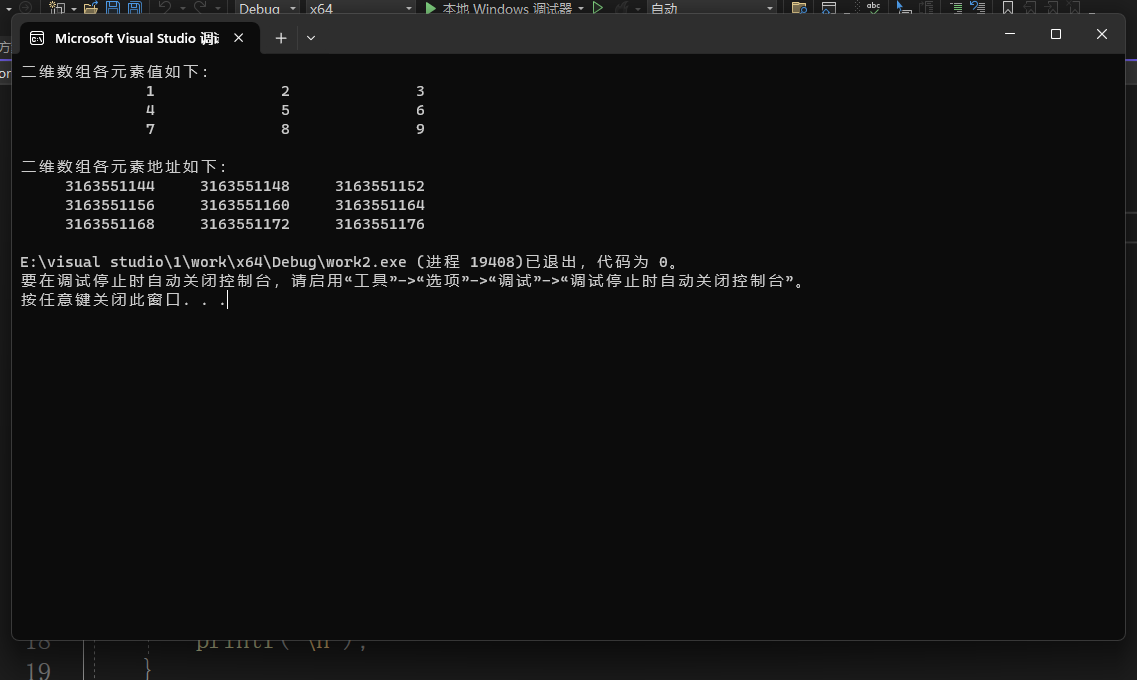
for (q = p[0]; q < p[0] + 3; q++)

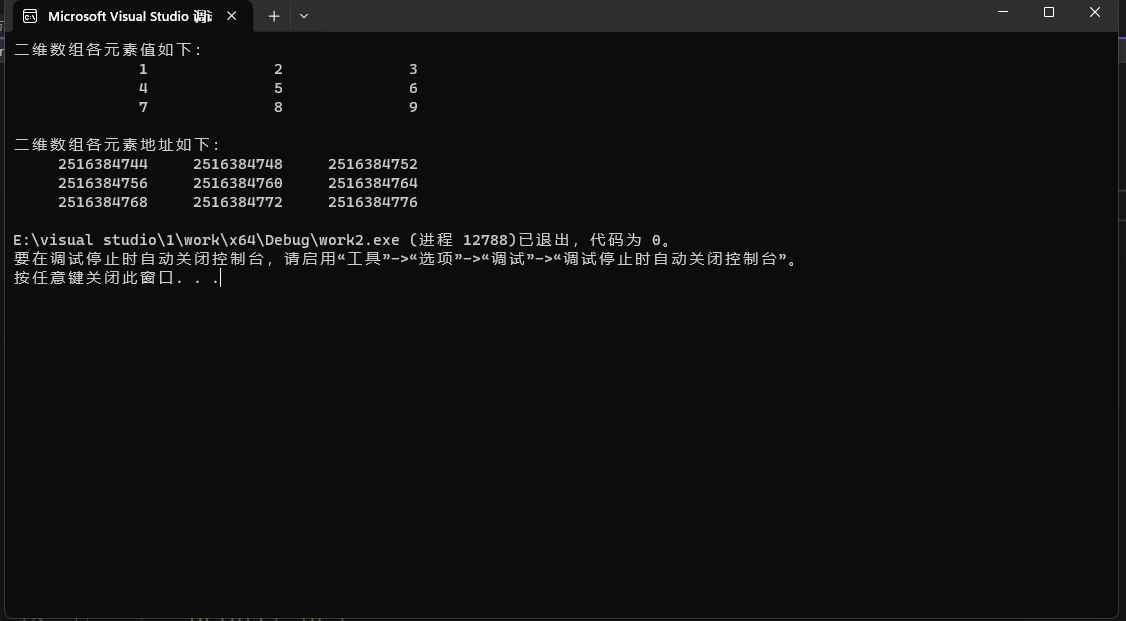
printf("%15u", q);

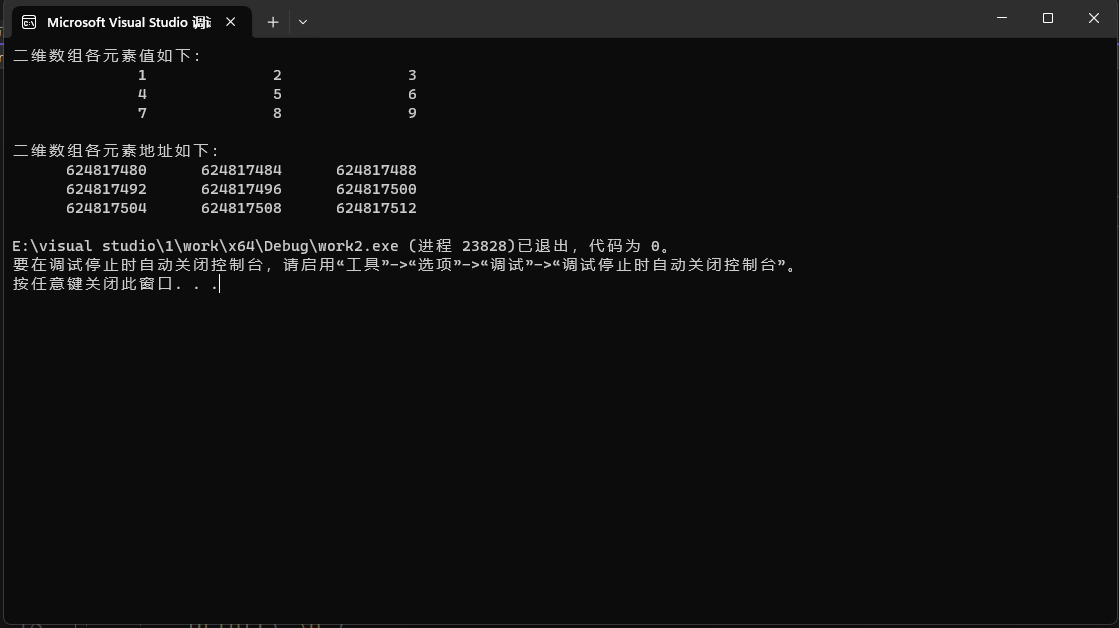
printf("\n");

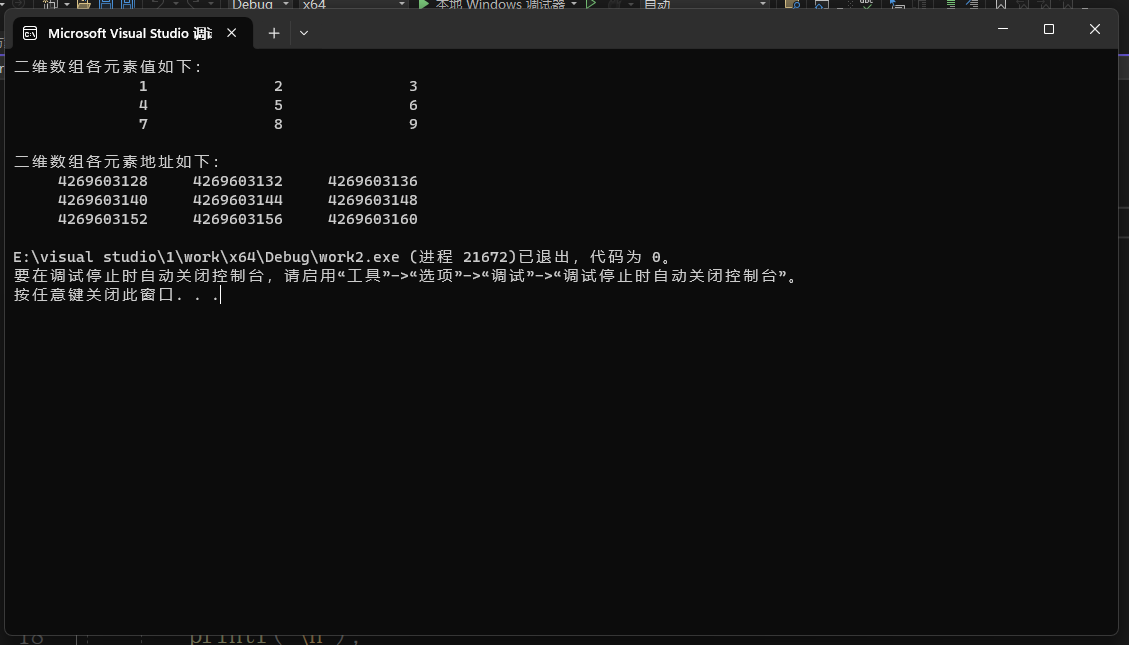
}

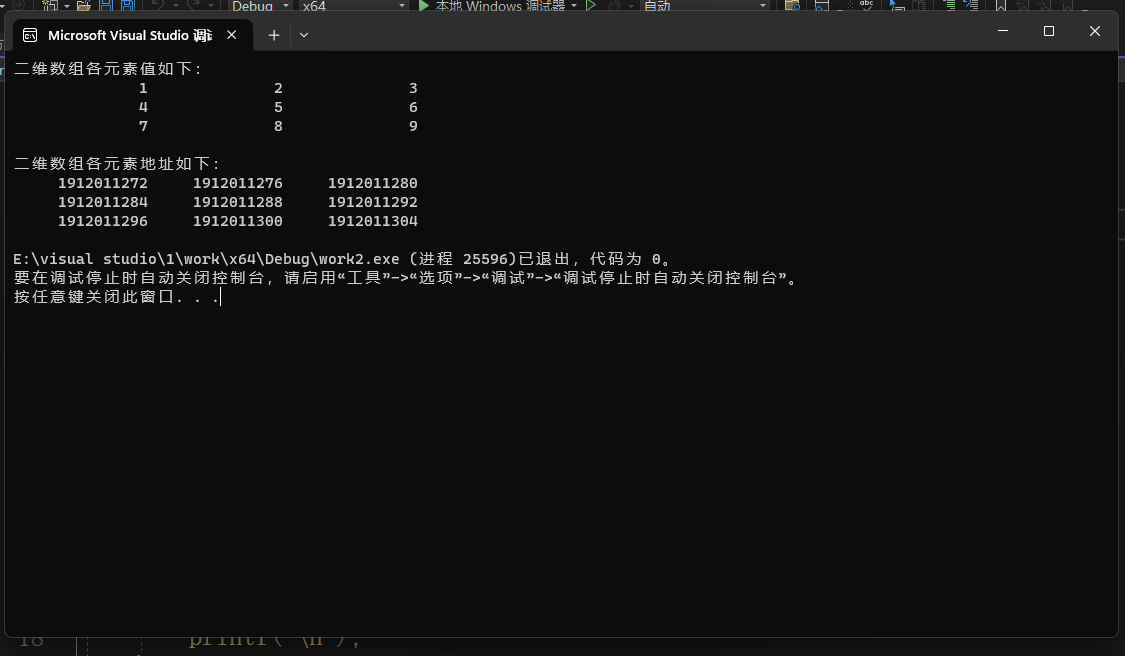
}











**3、**

#include<stdio.h>

main()

{

/\*首先定义一个二维数组并输入其各元素\*/

int i, j, a[3][3];

int n = 3;

printf("请输入二维数组的各元素：");

for (i = 0; i < 3; i++)

for (j = 0; j < 3; j++)

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

int\* b[3]; //定义一个一维数组指针指向二维数组各行

for (i = 0; i < 3; i++)

b[i] = &a[i][0];

/\*首先输出原二维数组\*/

int(\*p)[3], \* q;

printf("原二维数组各元素如下：\n");

for (p = a; p < a + 3; p++)

{

for (q = p[0]; q < p[0] + 3; q++)

printf("%10d", \*q);

printf("\n");

}

/\*下面进行ZigZag排序并输出\*/

void ZigZag(int\*a[], int);

printf("经过ZigZag排序后：\n");

ZigZag(b, n);

}

/\*下面编写一个进行Zig-Zag排序输出的函数\*/

void ZigZag(int \*a[], int n)

{

int i, j, k;

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

{

if (k % 2 == 0)

{

for (i = k; i >= 0; i--)

{

j = k - i;

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

}

}

else

{

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

{

j = k - i;

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

}

}

}

for (k = n; k <= (n \* 2 - 2); k++)

{

if (k % 2 == 0)

{

for (i = n - 1; i >= (k - n + 1); i--)

{

j = k - i;

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

}

}

else

{

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

{

j = k - i;

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

}

}

}

}

