**5.**

**int main() {**

**int n = 10;**

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

**int tmp = 1, ans = 1, i, now = a[0];**

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

**if(now == a[i]){**

**tmp++;**

**} else {**

**tmp = 1;**

**now = a[i];**

**}**

**if(tmp > ans) ans = tmp;**

**}**

**printf("%d", ans);**

**}**

**#define N 100**

**int main()**

**{**

**int length = 1, n, a[N], i, m;**

**printf("输入数组的元素个数(<101)：");**

**scanf("%d", &n);**

**printf("输入元素序列a\n");**

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

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

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

**if (a[i - length] == a[i])**

**length++;**

**}**

**printf("数组最长平台的元素个数为：%d\n", length);**

**return 0;**

**}**

**11.**

**#include<stdio.h>**

**void f(int n)**

**{**

**int x = 1;**

**int y;**

**int i = 0, j;**

**int a[300];**

**printf("1/%d=0.", n);**

**while (1)**

**{**

**for (j = 0; j < i; j++)**

**if (a[j] == x) break;**

**if (j != i) break;**

**a[i++] = x;**

**while (1)**

**{**

**if (x == 0) break;**

**x = 10 \* x;**

**if (x >= n) break;**

**else printf("0");**

**}**

**y = x / n;**

**x = x % n;**

**if (y == 0)**

**{**

**printf("0");**

**break;**

**}**

**printf("%d", y);**

**}**

**return;**

**}**

**void main()**

**{**

**int a;**

**for (a = 2; a <= 50; a++)**

**{**

**f(a);**

**printf("\n");**

**}**

**}**

**12.**

**#include<stdio.h>**

**#define N 100**

**void reverse(int a[], int k)**

**{**

**int i = 0, j = k - 1;**

**while (i < j)**

**{**

**int temp = a[i];**

**a[i] = a[j];**

**a[j] = temp;**

**i++; j--;**

**}**

**}**

**void swap(int a[], int n, int m)**

**{**

**reverse(a, m);**

**reverse(a + m, n - m);**

**reverse(a, n);**

**}**

**int main() {**

**int i, n, a[N], m;**

**printf("输入数组a[]的元素个数:\n");**

**scanf("%d", &n);**

**printf("顺序输入数组a[]的%d个元素:\n", n);**

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

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

**while (1) {**

**printf("输入顺移多少位置:\n");**

**scanf("%d", &m);**

**if (m >= n)**

**{**

**printf("错误！请重新输入小于%d的数：\n", n);**

**continue;**

**}**

**if (m < n) swap(a, n, m%n);**

**printf("顺移%d位的数组是\n", m);**

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

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

**}**

**return 0;**

**}**

**14．**

**（1）**

**方法1：**

**#include <stdio.h>**

**int f[100][100];**

**int max(int a, int b) {**

**return a > b ? a : b;**

**}**

**int min(int a, int b) {**

**return a > b ? b : a;**

**}**

**int main() {**

**int i, j, n, p, q, k;**

**scanf\_s("%d", &n);**

**int now = 1;**

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

**switch (i % 2) {**

**case 0:**

**for (j = min(i, n - 1); j >= max(0, i - n + 1); j--) {**

**f[i - j][j] = now++;**

**}**

**break;**

**case 1:**

**for (j = max(0, i - n + 1); j <= min(i, n - 1); j++) {**

**f[i - j][j] = now++;**

**}**

**break;**

**}**

**}**

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

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

**printf("%5d", f[i][j]);**

**}**

**printf("\n");**

**}**

**}**

**方法2：**

**#include <stdio.h>**

**#define N 100**

**void Square(int a[][N], int n)**

**{**

**int m = 1;**

**int s = n \* n + 1, i, k;**

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

**{**

**if (i % 2 == 0)**

**{**

**for (k = 0; i - k >= 0; k++) {**

**a[k][i - k] = m;**

**a[n - 1 - k][n - 1 - i + k] = s - m;**

**m++;**

**}**

**}**

**else**

**{**

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

**{**

**a[k][i - k] = m;**

**a[n - 1 - k][n - 1 - i + k] = s - m;**

**m++;**

**}**

**}**

**}**

**}**

**int main() {**

**int n, i, j;**

**int a[N][N];**

**printf("输入矩阵边长:\n");**

**scanf("%d", &n);**

**Square(a, n);**

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

**{**

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

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

**printf("\n");**

**}**

**return 0;**

**}**

**（2）**

**方法1:**

**#include <stdio.h>**

**int f[100][100];**

**int main(void)**

**{**

**//用户输入的值，创建n\*n的矩阵**

**int n;**

**//蛇形从1开始计数**

**int count = 1;**

**//a[x][y],x是二维数组的第一个下标，y是第二个。**

**//round是蛇形矩阵的第几圈，从0开始。**

**int x, y, round;**

**scanf\_s("%d", &n);**

**//如果n是1，则直接输出。**

**if (n == 1) {**

**f[0][0] = count;**

**}**

**else {**

**//下面以n=5为例**

**//一共有2(5/2)圈蛇形**

**for (round = 0; round < n / 2; round++) {**

**/\* 以下循环执行后输出如下：**

**1 2 3 4 5**

**\*/**

**x = round;**

**for (y = round; y < n - round; y++) {**

**f[x][y] = count;**

**count++;**

**}**

**/\* 以下循环执行后输出如下：**

**1 2 3 4 5**

**6**

**7**

**8**

**\*/**

**y = n - round - 1;**

**for (x = round + 1; x < n - round - 1; x++) {**

**f[x][y] = count;**

**count++;**

**}**

**/\* 以下循环执行后输出如下：**

**1 2 3 4 5**

**6**

**7**

**8**

**13 12 11 10 9**

**\*/**

**x = n - round - 1;**

**for (y = n - round - 1; y >= round; y--) {**

**f[x][y] = count;**

**count++;**

**}**

**/\* 以下循环执行后输出如下：**

**1 2 3 4 5**

**16 6**

**15 7**

**14 8**

**13 12 11 10 9**

**\*/**

**y = round;**

**for (x = n - round - 1 - 1; x > round; x--) {**

**f[x][y] = count;**

**count++;**

**}**

**}**

**/\* 上面的大循环执行后输出如下：**

**1 2 3 4 5**

**16 17 18 19 6**

**15 24 20 7**

**14 23 22 21 8**

**13 12 11 10 9**

**\*/**

**if (n % 2 == 1) {**

**//如果n值奇数，将最中间的空填上**

**f[n / 2][n / 2] = count;**

**}**

**}**

**//打印矩阵**

**for (x = 0; x < n; x++) {**

**for (y = 0; y < n; y++) {**

**printf("%5d ", f[x][y]);**

**}**

**printf("\n");**

**}**

**printf("\n");**

**return 0;**

**}**

**方法2：**

**#include <stdio.h>**

**#define N 300**

**void main() {**

**int i, j, n, k = 1;**

**int a[N][N];**

**printf("请输入方阵的阶数：\n");**

**scanf("%d", &n);**

**for (i = 0; i < (n + 1) / 2; i++) {**

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

**a[i][j] = k++;**

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

**a[j][n - 1 - i] = k++;**

**for (j = n - 2 - i; j >= i; j--)**

**a[n - 1 - i][j] = k++;**

**for (j = n - 2 - i; j > i; j--)**

**a[j][i] = k++;**

**}**

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

**{**

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

**printf("%4d", a[i][j]);**

**printf("\n");**

**}**

**printf("\n");**

**}**

**16.**

**#include <stdio.h>**

**int Is\_void(char c) {**

**return c == ' ' || c == '\t' || c == '\n';**

**}**

**int main() {**

**char s[100] = "HELLOWORLD";**

**char tmp[100] = "";**

**int len = 0;**

**int i = 0;**

**while(Is\_void(s[i])){**

**i++;**

**}**

**while (s[i] != '\0'){**

**if (!Is\_void(s[i])) {**

**s[len++] = s[i];**

**}**

**else if(!Is\_void(s[i + 1])){**

**s[len++] = s[i];**

**}**

**}**

**if (Is\_void(s[len - 1])) {**

**len--;**

**}**

**s[len] = '\0';**

**}**