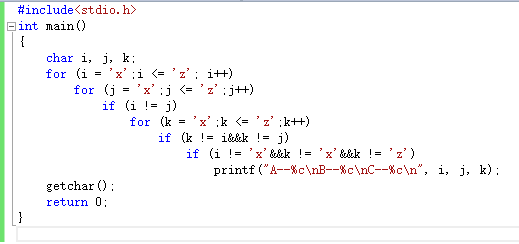
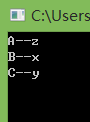
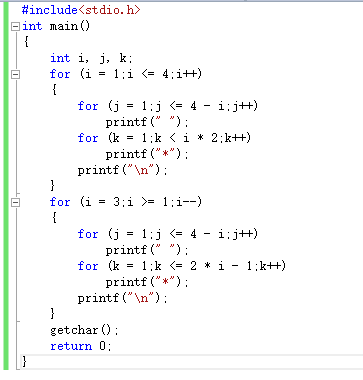
**1，两个乒乓球队进行比赛，各出3人，甲队为A,B,C 3人，乙队为X,Y,Z 3人。已抽签决定比赛名单。有人向队员打听比赛的名单，A说他不和X比，C说他不和X,Z比，请编程序找出3对赛手的名单。**





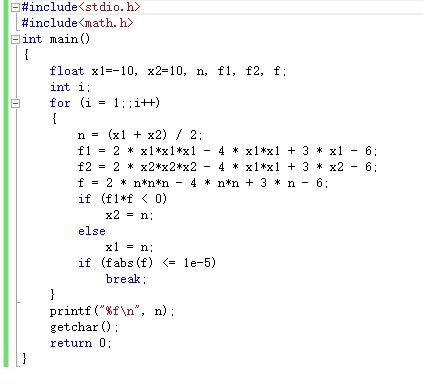
**2.输出下图**



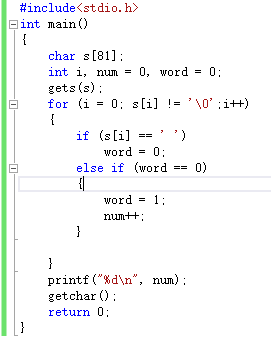


**3.用二分法求下面方程在（-10,10）之间的根**：

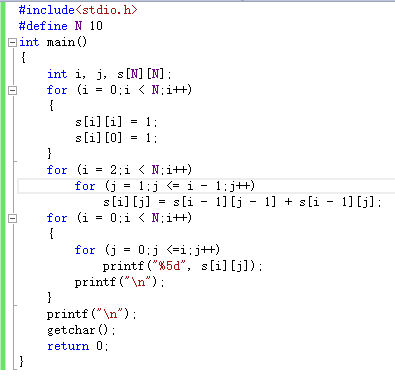
**2\*x\*x\*x-4\*x\*x+3\*x-6=0**

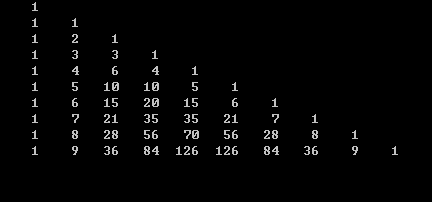


**4.输入一行字符，统计其中有多少单词，单词之间用空格分开。**



**5.杨辉三角**





**6.输出“魔方阵”。所谓魔方阵是指这样的方阵，它的每一行、每一列和对角线之和均相等。例如，三阶魔方阵为**

**8 1 6**

**3 5 7**

**4 9 2**

**要求输出1~n2的自然数构成的魔方阵。**

**解：**

**1）将1放在第一行的中间一列。**

**2）从2开始直到n\*n止各数依次按下列规律存放：每一个数存放的行比前一个数的行减1，列数加1.**

**3）如果上一数的行数为1，则下一数的行数为n(指最后一行)。**

**4）如果上一数的列数为n，则下一数的列数为1，行数减1.**

**5）如果按上面规则确定的位置上已有数，或上一个数是第1行第N列时，则把下一个数放在上一个数的下面。**

**程序：**

#include<stdio.h>

int main()

{

int s[15][15], i, j, k, n;

printf("please input a number1--15:\n");

scanf("%d", &n);

for (i = 1;i <= 15;i++)

for (j = 1;j <= 15;j++)

s[i][j] = 0;

j = n / 2 + 1;

s[1][j] = 1;

for (k = 2, i = 1;k <= n\*n;k++)

{

i = i - 1;

j = j + 1;

if (i<1 && j>n)

{

i = i + 2;

j = j - 1;

}

else

{

if (i < 1)i = n;

if (j > n)j = 1;

}

if (s[i][j] == 0)

s[i][j] = k;

else

{

i = i + 2;

j = j - 1;

s[i][j] = k;

}

}

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

{

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

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

printf("\n");

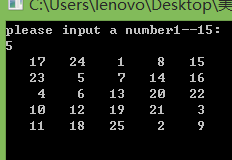
}

getchar();

getchar();

return 0;

}



**7，写一个用矩形法求定积分的通用函数，分别求、。**

解：用指向函数的指针

#include<stdio.h>

#include<math.h>

int main()

{

float gg(float(\*p)(float), float a, float b, int n);

void fsin(float x);

void fcos(float x);

void fexp(float x);

float a1, a2, b1, b2, c1, c2, z,(\*p)(float);

int n = 20;

printf("please inter a1 and a2:");

scanf("%f,%f", &a1, &a2);

printf("please inter b1 and b2:");

scanf("%f,%f", &b1, &b2);

printf("please inter c1 and c2:");

scanf("%f,%f", &c1, &c2);

p = fsin;

z = gg(p, a1, a2, n);

printf("The gg of sin(x) is:%f\n", z);

p = fcos;

z = gg(p, b1, b2, n);

printf("The gg of cos(x) is:%f\n",z);

p = fexp;

z = gg(p, c1, c2, n);

printf("The gg of exe(x) is:%f\n", z);

getchar();

getchar();

return 0;

}

float gg(float(\*p)(float), float a, float b, int n)

{

int i;

float x, h, s;

h = (b - a) / n;

s = 0;

x = a;

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

{

x = x + h;

s = s + (\*p)(x)\*h;

}

return (s);

}

void fsin(float x)

{

return(sin(x));

}

void fcos(float x)

{

return(cos(x));

}

void fexp(float x)

{

return(exp(x));

}

**排序方法：1、冒泡排序法2、选择排序法3、普通排序法。**

**8，用指向指针的指针的方法对5个字符串排序并输出。**

**程序：**

#include<stdio.h>

#include<string.h>

int main()

{

void gg(char \*\*s);

int i;

char \*\*p, \*pstr[5], str[5][20];

for (i = 0;i < 5;i++)

pstr[i] = str[i];

printf("please enter 5 strings:\n");

for (i = 0;i < 5;i++)

scanf("%s", pstr[i]);

p = pstr;

gg(p);

printf("\nstring ok:\n");

for (i = 0;i < 5;i++)

printf("%s\n", pstr[i]);

getchar();

getchar();

return 0;

}

void gg(char \*\*s)

{

char \*t;

int i, j;

for(i=0;i<4;i++)

for(j=i+1;j<5;j++)

if (strcmp(\*(s + i), \*(s + j)) > 0)

{

t = \*(s + i);

\*(s + i) = \*(s + j);

\*(s + j) = t;

}

}

**9，用指向指针的方法对n个整数排序输出。要求将排序单独写成一个函数。N个整数在主函数中输入，最后在主函数中输出。**

**程序：**

#include<stdio.h>

int main()

{

void gg(int \*\*s, int n);

int i,n, str[20], \*\*p, \*pstr[20];

printf("please enter n:");

scanf("%d", &n);

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

pstr[i] = &str[i];

printf("please enter these number:");

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

scanf("%d", pstr[i]);

p = pstr;

gg(p,n);

printf("Now,the sequence is:");

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

printf("%3d", \*pstr[i]);

printf("\n");

getchar();

getchar();

return 0;

}

void gg(int \*\*s,int n)

{

int i, j, \*t;

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

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

if(\*\*(s+i)>\*\*(s+j))

{

t = \*(s + i);

\*(s + i) = \*(s + j);

\*(s + j) = t;

}

}

**10.贪吃蛇**

**#include<stdio.h>**

**#include<time.h>**

**#include<windows.h>**

**#include<stdlib.h>**

**#define U 1**

**#define D 2**

**#define L 3**

**#define R 4 //蛇的状态，U：上 ；D：下；L:左 R：右**

**typedef struct SNAKE //蛇身的一个节点**

**{**

**int x;**

**int y;**

**struct SNAKE \*next;**

**}snake;**

**//全局变量//**

**int score = 0, add = 10;//总得分与每次吃食物得分。**

**int status, sleeptime = 200;//每次运行的时间间隔**

**snake \*head, \*food;//蛇头指针，食物指针**

**snake \*q;//遍历蛇的时候用到的指针**

**int endGamestatus = 0; //游戏结束的情况，1：撞到墙；2：咬到自己；3：主动退出游戏。**

**//声明全部函数//**

**void Pos(int,int);**

**void creatMap();**

**void initSnake();**

**int biteSelf();**

**void createFood();**

**void cantCrossWall();**

**void snakeMove();**

**void pause();**

**void runGame();**

**void initGame();**

**void endGame();**

**void gameStart();**

**void Pos(int x, int y)//设置光标位置**

**{**

**COORD pos;//定义控制台坐标**

**HANDLE hOutput;//定义句柄**

**pos.X = x;**

**pos.Y = y;**

**hOutput = GetStdHandle(STD\_OUTPUT\_HANDLE);//返回标准的输入、输出或错误的设备的句柄，也就是获得输入、输出/错误的屏幕缓冲区的句柄**

**SetConsoleCursorPosition(hOutput, pos);//设置控制台坐标**

**}**

**void creatMap()//创建地图**

**{**

**int i;**

**for (i = 0; i<58; i += 2)//打印上下边框**

**{**

**Pos(i, 0);**

**printf("■");//一个方块占两个位置**

**Pos(i, 26);**

**printf("■");**

**}**

**for (i = 1; i<26; i++)//打印左右边框**

**{**

**Pos(0, i);**

**printf("■");**

**Pos(56, i);**

**printf("■");**

**}**

**}**

**void initSnake()//初始化蛇身**

**{**

**snake \*tail;**

**int i;**

**tail = (snake\*)malloc(sizeof(snake));//从蛇尾开始，头插法，以x,y设定开始的位置//**

**tail->x = 24;**

**tail->y = 5;**

**tail->next = NULL;**

**for (i = 1; i <= 4; i++)//初始长度为4**

**{**

**head = (snake\*)malloc(sizeof(snake));**

**head->next = tail;**

**head->x = 24 + 2 \* i;**

**head->y = 5;**

**tail = head;**

**}**

**while (tail != NULL)//从头到为，输出蛇身**

**{**

**Pos(tail->x, tail->y);**

**printf("■");**

**tail = tail->next;**

**}**

**}**

**//??**

**int biteSelf()//判断是否咬到了自己**

**{**

**snake \*self;**

**self = head->next;**

**while (self != NULL)**

**{**

**if (self->x == head->x && self->y == head->y)**

**{**

**return 1;**

**}**

**self = self->next;**

**}**

**return 0;**

**}**

**void createFood()//随机出现食物**

**{**

**snake \*food\_1;**

**srand((unsigned)time(NULL));//为了防止每次产生的随机数相同，种子设置为time**

**food\_1 = (snake\*)malloc(sizeof(snake));**

**while ((food\_1->x % 2) != 0) //保证其为偶数，使得食物能与蛇头对其**

**{**

**food\_1->x = rand() % 52 + 2;**

**}**

**food\_1->y = rand() % 24 + 1;**

**q = head;**

**while (q->next == NULL)**

**{**

**if (q->x == food\_1->x && q->y == food\_1->y) //判断蛇身是否与食物重合**

**{**

**free(food\_1);**

**createFood();**

**}**

**q = q->next;**

**}**

**Pos(food\_1->x, food\_1->y);**

**food = food\_1;**

**printf("■");**

**}**

**void cantCrossWall()//不能穿墙**

**{**

**if (head->x == 0 || head->x == 56 || head->y == 0 || head->y == 26)**

**{**

**endGamestatus = 1;**

**endGame();**

**}**

**}**

**void snakeMove()//蛇前进,上U,下D,左L,右R**

**{**

**snake \* nexthead;**

**cantCrossWall();**

**nexthead = (snake\*)malloc(sizeof(snake));**

**if (status == U)**

**{**

**nexthead->x = head->x;**

**nexthead->y = head->y - 1;**

**if (nexthead->x == food->x && nexthead->y == food->y)//如果下一个有食物//**

**{**

**nexthead->next = head;**

**head = nexthead;**

**q = head;**

**while (q != NULL)**

**{**

**Pos(q->x, q->y);**

**printf("■");**

**q = q->next;**

**}**

**score = score + add;**

**createFood();**

**}**

**else //如果没有食物//**

**{**

**nexthead->next = head;**

**head = nexthead;**

**q = head;**

**while (q->next->next != NULL)**

**{**

**Pos(q->x, q->y);**

**printf("■");**

**q = q->next;**

**}**

**Pos(q->next->x, q->next->y);**

**printf(" ");**

**free(q->next);**

**q->next = NULL;**

**}**

**}**

**if (status == D)**

**{**

**nexthead->x = head->x;**

**nexthead->y = head->y + 1;**

**if (nexthead->x == food->x && nexthead->y == food->y) //有食物**

**{**

**nexthead->next = head;**

**head = nexthead;**

**q = head;**

**while (q != NULL)**

**{**

**Pos(q->x, q->y);**

**printf("■");**

**q = q->next;**

**}**

**score = score + add;**

**createFood();**

**}**

**else //没有食物**

**{**

**nexthead->next = head;**

**head = nexthead;**

**q = head;**

**while (q->next->next != NULL)**

**{**

**Pos(q->x, q->y);**

**printf("■");**

**q = q->next;**

**}**

**Pos(q->next->x, q->next->y);**

**printf(" ");**

**free(q->next);**

**q->next = NULL;**

**}**

**}**

**if (status == L)**

**{**

**nexthead->x = head->x - 2;**

**nexthead->y = head->y;**

**if (nexthead->x == food->x && nexthead->y == food->y)//有食物**

**{**

**nexthead->next = head;**

**head = nexthead;**

**q = head;**

**while (q != NULL)**

**{**

**Pos(q->x, q->y);**

**printf("■");**

**q = q->next;**

**}**

**score = score + add;**

**createFood();**

**}**

**else //没有食物**

**{**

**nexthead->next = head;**

**head = nexthead;**

**q = head;**

**while (q->next->next != NULL)**

**{**

**Pos(q->x, q->y);**

**printf("■");**

**q = q->next;**

**}**

**Pos(q->next->x, q->next->y);**

**printf(" ");**

**free(q->next);**

**q->next = NULL;**

**}**

**}**

**if (status == R)**

**{**

**nexthead->x = head->x + 2;**

**nexthead->y = head->y;**

**if (nexthead->x == food->x && nexthead->y == food->y)//有食物**

**{**

**nexthead->next = head;**

**head = nexthead;**

**q = head;**

**while (q != NULL)**

**{**

**Pos(q->x, q->y);**

**printf("■");**

**q = q->next;**

**}**

**score = score + add;**

**createFood();**

**}**

**else //没有食物**

**{**

**nexthead->next = head;**

**head = nexthead;**

**q = head;**

**while (q->next->next != NULL)**

**{**

**Pos(q->x, q->y);**

**printf("■");**

**q = q->next;**

**}**

**Pos(q->next->x, q->next->y);**

**printf(" ");**

**free(q->next);**

**q->next = NULL;**

**}**

**}**

**if (biteSelf() == 1) //判断是否会咬到自己**

**{**

**endGamestatus = 2;**

**endGame();**

**}**

**}**

**void pause()//暂停**

**{**

**while (1)**

**{**

**Sleep(300);**

**if (GetAsyncKeyState(VK\_SPACE))**

**{**

**break;**

**}**

**}**

**}**

**void runGame()//控制游戏**

**{**

**Pos(64, 15);**

**printf("不能穿墙，不能咬到自己\n");**

**Pos(64, 16);**

**printf("用↑.↓.←.→分别控制蛇的移动.");**

**Pos(64, 17);**

**printf("F1 为加速，F2 为减速\n");**

**Pos(64, 18);**

**printf("ESC ：退出游戏.space：暂停游戏.");**

**status = R;**

**while (1)**

**{**

**Pos(64, 10);**

**printf("得分：%d ", score);**

**Pos(64, 11);**

**printf("每个食物得分：%d分", add);**

**if (GetAsyncKeyState(VK\_UP) && status != D)**

**{**

**status = U;**

**}**

**else if (GetAsyncKeyState(VK\_DOWN) && status != U)**

**{**

**status = D;**

**}**

**else if (GetAsyncKeyState(VK\_LEFT) && status != R)**

**{**

**status = L;**

**}**

**else if (GetAsyncKeyState(VK\_RIGHT) && status != L)**

**{**

**status = R;**

**}**

**else if (GetAsyncKeyState(VK\_SPACE))**

**{**

**pause();**

**}**

**else if (GetAsyncKeyState(VK\_ESCAPE))**

**{**

**endGamestatus = 3;**

**break;**

**}**

**else if (GetAsyncKeyState(VK\_F1))**

**{**

**if (sleeptime >= 50)**

**{**

**sleeptime = sleeptime - 30;**

**add = add + 2;**

**if (sleeptime == 320)**

**{**

**add = 2;//防止减到1之后再加回来有错**

**}**

**}**

**}**

**else if (GetAsyncKeyState(VK\_F2))**

**{**

**if (sleeptime<350)**

**{**

**sleeptime = sleeptime + 30;**

**add = add - 2;**

**if (sleeptime == 350)**

**{**

**add = 1; //保证最低分为1**

**}**

**}**

**}**

**Sleep(sleeptime);**

**snakeMove();**

**}**

**}**

**void initGame()//开始界面**

**{**

**Pos(40, 12);**

**system("title 美气的哼");**

**printf("欢迎来到贪食蛇游戏！");**

**system("pause");**

**system("cls");**

**Pos(25, 12);**

**printf("用↑.↓.←.→分别控制蛇的移动， F1 为加速，2 为减速\n");**

**Pos(25, 13);**

**printf("加速将能得到更高的分数。\n");**

**system("pause");**

**system("cls");**

**}**

**void endGame()//结束游戏**

**{**

**system("cls");**

**Pos(24, 12);**

**if (endGamestatus == 1)**

**{**

**printf("对不起，您撞到墙了。游戏结束.");**

**}**

**else if (endGamestatus == 2)**

**{**

**printf("对不起，您咬到自己了。游戏结束.");**

**}**

**else if (endGamestatus == 3)**

**{**

**printf("您的已经结束了游戏。");**

**}**

**Pos(24, 13);**

**printf("您的得分是%d\n", score);**

**while (getchar() != 'y')**

**{**

**printf("close?[y]");**

**}**

**exit(0);**

**}**

**void gameStart()//游戏初始化**

**{**

**system("mode con cols=100 lines=30");**

**initGame();**

**creatMap();**

**initSnake();**

**createFood();**

**}**

**int main()**

**{**

**gameStart();**

**runGame();**

**endGame();**

**return 0;**

**}**