目录

[1程序简介 1](#_Toc474413722)

[2程序设计 1](#_Toc474413723)

[2.1程序分析程序初始化： 1](#_Toc474413724)

[2.2游戏流程 2](#_Toc474413725)

[2.3程序流程图 3](#_Toc474413726)

[2.4数据定义及重要函数 4](#_Toc474413727)

[3系统测试及改进 12](#_Toc474413728)

[3.1程序测试 12](#_Toc474413729)

[3.2系统改进 14](#_Toc474413730)

[4源码 15](#_Toc474413731)

# 1程序简介

贪吃蛇游戏是一个经典小游戏，在封闭的围墙里面，通过键盘的上下左右控制蛇的方向，寻找随机出现的食物。蛇头撞到食物，则食物被吃掉，蛇身体长度+1，同时记100分，蛇的身子越吃越长，身子越长玩的难度就越大，如果蛇在移动中撞到墙或身体交叉蛇头撞倒自己身体游戏结束。

# 2程序设计

## 2.1程序分析程序初始化：

加载在数据段定义好相关的提示信息以及储存蛇身的数组，主要是如何开始游戏的提示信息

蛇的移动：

程序的关键是表示蛇的图形以及蛇的移动。一开始用4个小矩形表示蛇的身体，每吃一个食物，身体增加一个矩形，移动时必须从蛇头开始，所以蛇不能向相反方向移动，蛇头的前进方向也就是蛇的方向，蛇尾不能改作蛇头。设置一个等待时间，如果不按任何按键，蛇在当前方向上前进。按了有效的方向键后，先确定蛇头的位置然后蛇身体随着蛇头移动，图形的实现是从蛇头的新位置开始画出蛇，这时由于没有清屏的原因，原来蛇的位置和新蛇的位置相差一个单位，所以看起来蛇会多一节身体。

食物的产生：

通过一系列运算产生在要求范围内的的随机数，此时的随机数便是随即点，可以通过产生的随即点的坐标画出随即点，也就是食物了。

蛇吃食物：

蛇头的位置可以与随即点的位置坐标判断，如果此时两点的坐标是相同的，那么该食物被蛇吃掉，蛇的长度+1。

判断游戏结束的条件：

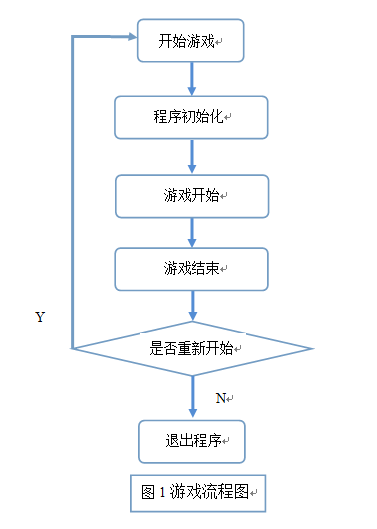
当蛇的头部位置坐标与边界坐标重合或者蛇头的位置与自身相重合，游戏结束

暂停：

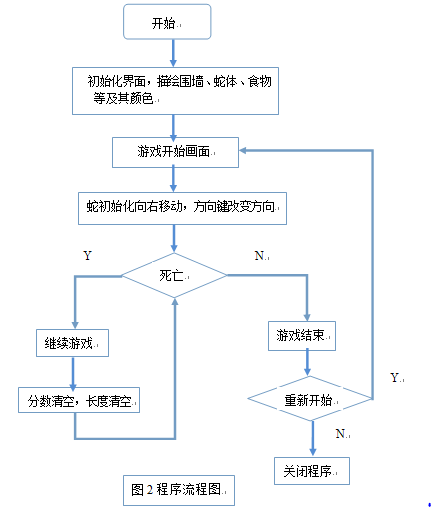
游戏过程中，按空格键可以使游戏暂停，在按空格键进入被暂停的游戏

## 2.2游戏流程

由程序分析知，该游戏是用方向键实现贪吃蛇的移动，游戏大概流程分为：游戏初始化（描绘围墙、蛇体、食物等及其颜色），游戏过程（通过四个方向键控制蛇头带动身体移动），游戏结束（显示总分并判断是否重新游戏）。



## 2.3程序流程图



## 2.4数据定义及重要函数

数据定义及键盘扫描码

数据定义：

Len db ? ;蛇的长度

Body dw 200 dup(0) ;蛇的身体位置

Direction dw 256

Score db 'Score00000' ;得分

snake\_color db 0ah ;蛇颜色

food\_color db 0ch ;食物色

speed db 0 ;速度

gameover db 0 ;游戏结束标记

Point db 0 ;得分数

foodX dw ? ;食物坐标

foodY dw ?

seed dw 2 ;随即数种子

键盘扫描码：

LEFT EQU 4Bh

RIGHT EQU 4Dh

UP EQU 48h

DOWN EQU 50h

QUIT EQU 01h

SPACE EQU 39h

main函数

主函数首先加载定义好的数据，设置图形模式大小，调用显示开始提示信息的函数，中断等待用户按键输入，判断退出或开始游戏。进入游戏，重新设置图形模式，调用rec，clear\_area,getfood,run\_sanke函数分别打印边框，初始化蛇，随机食物，开始运动蛇。当蛇死亡时，中断，等待用户输入，根据用户的输入继续游戏或退出程序。

main proc far

mov ax , @data

mov ds , ax

start:

mov ah , 00

mov al , 12h

int 10h ;设置图形模式640\*480

call show\_start\_msg ;显示开始提示信息

call show\_title

mov ah , 00h

int 16h ;等待按键

cmp ah , QUIT

jz dos ;ESC则退出，否则进入游戏

game\_start:

mov ah , 00 ;设置图形模式320\*200，游戏在此模式下运行

mov al , 13h

int 10h

mov cl , 7 ;边框颜色

mov si , 59

mov bx , 19

Init\_snake ;初始化蛇

call getfood ;得到第一个食物

call run\_snake ;开始运动蛇

mov dx , 1010h ;设置光标

call SetGb ;设置光标位置

mov bl , white ;显示死亡信息

lea bp , msg3

mov cx , 13

call Msg ;显示字符串

mov dx , 1605h

mov bl , white ;显示提示信息

lea bp , msg5

mov cx , 29

call Msg

exit\_loop:

mov ah , 00h

int 16h ;等待按键

cmp ah , QUIT

jz start

cmp ah , SPACE

jz game\_start

jmp exit\_loop ;返回开始

dos:

mov ah , 00

mov al , 03

int 10h ;设置文本模式

mov ah , 4ch

int 21h ;退出到DOS

main endp

Run\_snake函数

蛇一开始默认向右运动，当按下空格键时游戏暂停，再按空格键游戏开始。用INT 10H的1号功能检测是否有按键输入，改变蛇头前进的方向，没有保持当前的前进方向。在蛇运动过程中，不断检查是否撞墙或蛇头触碰到蛇身，如果是，跳转到stop，提示重新或退出。当检查到食物坐标和蛇头坐标一致时，食物被吃，蛇身长度+1，保存当前前进方向。

run\_snake proc

push ax

push bx

push cx

push dx

push si

push di

mov dx, 0005h

call SetGb

mov bl,white

lea bp,msg6

mov cx,30

call Msg

loop1:

call Delay

mov al,1

cmp al,gameover

jz stop

mov bx , offset Body

mov cx,0

mov cl,Len

add bx,cx

add bx,cx

sub bx,2

A1:

mov ax,[bx]

mov [bx+2],ax

sub bx,2

loop A1

addhead:

mov bx , offset Body ;增加新头部并改写数据

mov ax , [bx]

mov dx , Direction

add ah , dh ;坐标加方向

add al , dl ;坐标加方向

mov [bx] , ax ;存入头部

mov dx,0 ;为check\_die准备参数si,di

mov dl , ah

mov si , dx

mov dl , al

mov di , dx

call check\_die ;检查死亡

mov al,1

cmp al,gameover

jz stop

call check\_eat ;检查吃

cmp cx,1

jz A2

mov bx , offset Body

mov ax,0

add Len,1

mov al , Len

add bx , ax ;取到蛇尾

add bx , ax

sub bx , 2

mov ax , [bx]

mov cx,0

mov [bx],cx

mov dx,0

mov dl , ah

mov si , dx

mov dl , al

mov di , dx

sub Len,1

call clearcut

mov bx , offset Body

mov ax, [bx]

mov dx,0

mov dl , ah

mov si , dx

mov dl , al

mov di , dx

call cutsnake

jmp A3

A2:

add Len,1

mov bx , offset Body

mov ax, [bx]

mov dx,0

mov dl , ah

mov si , dx

mov dl , al

mov di , dx

call cutsnake ;画新头部

call getfood

A3:

call show\_point

call check\_key ;检查按键

jmp loop1 ;蛇运动循环DATAS SEGMENT

stop:

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

run\_snake endp

getfood函数

设置了一个随机数坐标x最大值50，调用Rand函数获取随机x轴坐标，同理可得随机y轴坐标，该坐标就是食物的随机坐标。调用cutfood函数，将随机食物在显示屏输出

getfood proc

push ax

push bx

push cx

push dx

push si

push di

M1:

mov si,50 ;置随机数范围（最大值+1）

call Rand ;产生随机数

mov foodx,ax ;赋予食物坐标x

mov si,50 ;置随机数范围（最大值+1）

call Rand ;产生随机数

mov foody,ax ;赋予食物坐标y

mov si,foodx

mov di,foody

mov ax,di ;取坐标值

mul cut\_y ;乘象素值

add ax,topline

mov dx,ax

push dx

mov ax,si ;取坐标值

mul cut\_x ;乘象素值

add ax,leftline

mov cx,ax

pop dx

mov ah,0dh

mov bh,0

int 10h

cmp al,snake\_color

jz M1

call cutfood ;画食物

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

getfood endp

Check\_key函数

响应键盘输入，控制蛇的移动以及控制游戏退出

Check\_key proc

push ax

push bx

push cx

push dx

push si

push di

mov ah , 01h

int 16h

jz check\_nokey ;如果没有按键就继续

mov ah , 0 ;取得扫描码

int 16h

push ax

cmp ah , SPACE

jnz S2

S1:

mov ah,0

int 16h

cmp ah, SPACE

jnz S1

S2:

pop ax

cmp ah , LEFT ;往左移动

jz check\_LEFT

cmp ah , RIGHT ;往右边移动

jz check\_RIGHT

cmp ah , UP ;往上移动

jz check\_UP

cmp ah , DOWN ;往下移动

jz check\_DOWN

cmp ah , QUIT ;结束游戏

jz check\_QUIT

jmp check\_nokey

check\_LEFT:

mov ah , -1 ;(-1,0)

mov al , 0

mov bx , ax ;检查是否反向

add bx , Direction

jz check\_nokey

mov Direction , ax

jmp check\_nokey

check\_RIGHT:

mov ah , 1 ;(1,0)

mov al , 0

mov bx , ax ;检查是否反向

add bx , Direction

jz check\_nokey

mov Direction , ax

jmp check\_nokey

check\_UP:

mov ah , 0 ;(0,-1)

mov al , -1

mov bx , Direction ;检查是否反向

sub bl , 1

jz check\_nokey

mov Direction , ax

jmp check\_nokey

check\_DOWN:

mov ah , 0 ;(0,1)

mov al , 1

mov bx , Direction ;检查是否反向

add bl , 1

jz check\_nokey

mov Direction , ax

jmp check\_nokey

check\_QUIT:

jmp check\_exit

check\_exit: ;按下ESC键的情况

mov gameover,1

check\_nokey:

mov ah,0ch

mov al,0

int 21h

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

Check\_key endp

# 3系统测试及改进

## 3.1程序测试

打开程序，显示欢迎界面，提醒用户输入，



按任意键进入程序，游戏开始时，贪吃蛇默认向右移动，使用键盘的上下左右控制贪吃蛇的前进的方向，最下方的Score实时更新分数，游戏过程中空格键可以暂停和继续、

每吃一个食物，Socre+100，蛇身长度+1



当蛇头撞墙之后，弹出信息—Game is over！程序中断，提示并等待用户操作，当按ESC键时退回主界面，按空格键重新开始游戏，此时分数归零，蛇身长度也恢复到开始时的默认值



如图，当蛇头撞上蛇身时，等同蛇头撞上墙壁



## 3.2系统改进

（1）游戏制作的精细度和画面的质量有待提高，改善蛇和食物的样子

（2）联机，可以多台机器联机比赛，

（3）可以记录历史最好成绩

（4）某些食物具有特效，比如缩短蛇身的长度，可以穿墙

（5）可以改变游戏的速度，

（6）长时间游戏提醒玩家休息

（7）实现背景音乐

（8）可以选择不同形状的蛇身，改变蛇的颜色

（9）可以保存当前游戏，玩家下次可以继续当前游戏

# 4源码

.model small

.stack 200

.data

Len db ? ;蛇的长度

Body dw 200 dup(0) ;蛇的身体位置

Direction dw 256

Score db 'Score00000' ;得分

snake\_color db 0ah ;蛇颜色

food\_color db 0ch ;食物色

speed db 0 ;速度

gameover db 0 ;游戏结束标记

Point db 0 ;得分数

foodX dw ? ;食物坐标

foodY dw ?

seed dw 2 ;随即数种子

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;显示方面的参数

topline dw 20 ;上边界

leftline dw 60 ;左边界

cut\_x dw 4 ;每格宽

cut\_y dw 3 ;每格长

max\_x dw 49 ;最大格数（横）

max\_y dw 49 ;最大格数（纵）

msg1 db 'Greedy Snakes'

msg2 db 'by:xiaobin'

msg3 db 'Game is over!'

msg4 db 'Press any key for start! Esc for exit'

msg5 db 'ESC for back, Space for retry'

msg6 db 'press space for pause/continue'

msg7 db ' '

title1 db 6 dup(20h), 5 dup(0dch), 5 dup(20h) ,5 dup(0dch), 5 dup(20h)

title2 db 6 dup(20h), 0dbh,3 dup(20h),0dbh,5 dup(20h),0dbh,3 dup(20h),0dbh ,5 dup(20h)

title3 db 6 dup(20h), 0dbh,3 dup(20h),0dbh,5 dup(20h) ,0dbh,3 dup(20h),0dbh,5 dup(20h)

title4 db 7 dup(0dfh), 3 dup(20h),7 dup(0dfh), 3 dup(20h),5 dup(0dfh),0fdh

yello db 0eh

green db 02h

white db 0fh

blue db 01h

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;键盘扫描码

LEFT EQU 4Bh

RIGHT EQU 4Dh

UP EQU 48h

DOWN EQU 50h

QUIT EQU 01h

SPACE EQU 39h

.code

main proc far

mov ax , @data

mov ds , ax

start:

mov ah , 00

mov al , 12h

int 10h ;设置图形模式640\*480

call show\_start\_msg ;显示开始提示信息

call show\_title

mov ah , 00h

int 16h ;等待按键

cmp ah , QUIT

jz does ;ESC则退出，否则进入游戏

game\_start:

mov ah , 00 ;设置图形模式320\*200，游戏在此模式下运行

mov al , 13h

int 10h

mov cl , 7 ;边框颜色

mov si , 59

mov bx , 19

mov di , 260

mov dx , 170

call rec ;打印边框

call clear\_area ;游戏区域清空（效果）

call Init\_snake ;初始化蛇

call getfood ;得到第一个食物

call run\_snake ;开始运动蛇

exit:

mov dx , 1010h ;设置光标

call SetGb ;设置光标位置

mov bl , white ;显示死亡信息

lea bp , msg3

mov cx , 13

call Msg ;显示字符串

mov dx , 1605h

mov bl , white ;显示提示信息

lea bp , msg5

mov cx , 29

call Msg

exit\_loop:

mov ah , 00h

int 16h ;等待按键

cmp ah , QUIT

jz start

cmp ah , SPACE

jz game\_start

jmp exit\_loop ;返回开始

does:

mov ah , 00

mov al , 03

int 10h ;设置文本模式

mov ah , 4ch

int 21h ;退出到DOS

main endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: SetGb

;功能: 设置光标位置

;传递参数: dh/dl 行/列

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SetGb proc near

push ax

push bx

mov bh , 0

mov ah , 02h

int 10h

pop bx

pop ax

ret

SetGb endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: Msg

;功能: 输出字符串

;传递参数:

; bl 颜色号（BL RGB IRGB）

; bp 字符串偏移地址

; cx 字符串长度

; dh/dl 行/列

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Msg proc near

push ax

push bx

push es

push ds

pop es ;将数据段ds送es

mov al , 0

mov ah , 13h

mov bh , 0

int 10h

pop es

pop bx

pop ax

ret

Msg endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: show\_title

;功能: 显示开始提示信息

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

show\_title proc near

push ax

push bx

push cx

push dx

mov dl,10h

mov cx,10h

yidong:

push cx

dec dl

mov dh,0fh

call SetGb

mov al,020h

mov ah,0eh

int 10h

inc dl

mov dh,0ch

call SetGb

mov bl,yello

lea bp,title1

mov cx,26

call Msg

mov dh,0dh

call SetGb

mov bl,yello

lea bp,title2

mov cx,26

call Msg

mov dh,0eh

call SetGb

mov bl,yello

lea bp,title3

mov cx,26

call Msg

mov dh,0fh

call SetGb

mov bl,yello

lea bp,title4

mov cx,26

call Msg

call delay

inc dl

pop cx

call show\_start\_msg

loop yidong

mov ah,0ch

mov al,0

int 21h

pop dx

pop cx

pop bx

pop ax

ret

show\_title endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: show\_start\_msg

;功能: 显示开始提示信息

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

show\_start\_msg proc near

push ax

push bx

push cx

push dx

mov dx,1426h

call SetGb

mov bl,0a1h

lea bp,msg1

mov cx,12

call Msg

mov dx , 190bh

call SetGb ;设置光标

mov bl,08h

lea bp,msg2 ;显示作者信息

mov cx,11

call Msg ;显示字符串

mov dx , 1619h ;设置光标

call SetGb

mov bl , white

lea bp , msg4 ;显示提示信息

mov cx , 37

call Msg ;显示字符串

pop dx

pop cx

pop bx

pop ax

ret

show\_start\_msg endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: cutsnake

;功能: 显示蛇的一格

;传递参数:

; si/di 游戏空间列/游戏空间行(50\*50)

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

cutsnake proc near

push ax

push bx

push cx

push dx

push si

push di

mov cl, snake\_color ;置颜色

mov ax,di ;取坐标值

mul cut\_y ;乘象素值

add ax,topline ;加上边界

mov bx , ax

add ax , 2

mov dx , ax

push bx

push dx

mov ax,si ;取坐标值

mul cut\_x ;乘象素值

add ax,leftline ;加上边界

mov si , ax

add ax , 3

mov di , ax

pop dx

pop bx

call Rec ;Rec的参数是cl颜色 si左边 bx上边 di右边 dx下边

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

cutsnake endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: cutfood

;功能: 显示食物

;传递参数:

; si/di 游戏空间列/游戏空间行(50\*50)

;函数返回: 空

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

cutfood proc near

push ax

push bx

push cx

push dx

push si

push di

mov cl , food\_color ;置颜色

mov ax,di ;取坐标值

mul cut\_y ;乘象素值

add ax,topline ;加上边界

mov bx , ax

add ax , 2

mov dx , ax

push bx

push dx

mov ax,si ;取坐标值

mul cut\_x ;乘象素值

add ax,leftline ;加上边界

mov si , ax

add ax , 3

mov di , ax

pop dx

pop bx

call Rec ;Rec的参数是cl颜色 si左边 bx上边 di右边 dx下边

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

cutfood endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: clearcut

;功能: 清除格

;传递参数:

; si/di 游戏空间列/游戏空间行(50\*50)

;函数返回: 空

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

clearcut proc near

push ax

push bx

push cx

push dx

push si

push di

mov cl , 0 ;置颜色(黑)

mov ax,di ;取坐标值

mul cut\_y ;乘象素值

add ax,topline ;加上边界

mov bx , ax

add ax , 2

mov dx , ax

push bx

push dx

mov ax,si ;取坐标值

mul cut\_x ;乘象素值

add ax,leftline ;加上边界

mov si , ax

add ax , 3

mov di , ax

pop dx

pop bx

call Rec ;Rec的参数是cl颜色 si左边 bx上边 di右边 dx下边

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

clearcut endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: Rand

;功能: 产生指定范围大小的随机数

;传递参数: si 数字的范围最大值+1

;函数返回: ax 返回随机数的值

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Rand proc

push bx

push cx

push dx

push si

mov ah,2ch

int 21h

mov ax,dx

mov dx,0

mov bx,si ;指定随机数的范围

div bx

mov ax,dx

pop si

pop dx

pop cx

pop bx

ret

Rand endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: getfood

;功能: 产生新食物,即随机产生一组49\*49的坐标

;函数返回: foodx与foody的值

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

getfood proc

push ax

push bx

push cx

push dx

push si

push di

M1:

mov si,50 ;置随机数范围（最大值+1）

call Rand ;产生随机数

mov foodx,ax ;赋予食物坐标x

mov si,50 ;置随机数范围（最大值+1）

call Rand ;产生随机数

mov foody,ax ;赋予食物坐标y

mov si,foodx

mov di,foody

mov ax,di ;取坐标值

mul cut\_y ;乘象素值

add ax,topline

mov dx,ax

push dx

mov ax,si ;取坐标值

mul cut\_x ;乘象素值

add ax,leftline

mov cx,ax

pop dx

mov ah,0dh

mov bh,0

int 10h

cmp al,snake\_color

jz M1

call cutfood ;画食物

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

getfood endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: Delay

;功能: 延时

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Delay proc

push ax

push bx

push cx

push dx

mov cx,33144

waitf:

in al,61h

and al,10h

cmp al,ah

je waitf

mov ah,al

loop waitf

pop dx

pop cx

pop bx

pop ax

ret

Delay endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: Check\_key

;功能: 处理键盘响应

;函数返回: Direction的值,gameover的值（按ESC时）

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Check\_key proc

push ax

push bx

push cx

push dx

push si

push di

mov ah , 01h

int 16h

jz check\_nokey ;如果没有按键就继续

mov ah , 0 ;取得扫描码

int 16h

push ax

cmp ah , SPACE

jnz S2

S1:

mov ah,0

int 16h

cmp ah, SPACE

jnz S1

S2:

pop ax

cmp ah , LEFT ;往左移动

jz check\_LEFT

cmp ah , RIGHT ;往右边移动

jz check\_RIGHT

cmp ah , UP ;往上移动

jz check\_UP

cmp ah , DOWN ;往下移动

jz check\_DOWN

cmp ah , QUIT ;结束游戏

jz check\_QUIT

jmp check\_nokey

check\_LEFT:

mov ah , -1 ;(-1,0)

mov al , 0

mov bx , ax ;检查是否反向

add bx , Direction

jz check\_nokey

mov Direction , ax

jmp check\_nokey

check\_RIGHT:

mov ah , 1 ;(1,0)

mov al , 0

mov bx , ax ;检查是否反向

add bx , Direction

jz check\_nokey

mov Direction , ax

jmp check\_nokey

check\_UP:

mov ah , 0 ;(0,-1)

mov al , -1

mov bx , Direction ;检查是否反向

sub bl , 1

jz check\_nokey

mov Direction , ax

jmp check\_nokey

check\_DOWN:

mov ah , 0 ;(0,1)

mov al , 1

mov bx , Direction ;检查是否反向

add bl , 1

jz check\_nokey

mov Direction , ax

jmp check\_nokey

check\_QUIT:

jmp check\_exit

check\_exit: ;按下ESC键的情况

mov gameover,1

check\_nokey:

mov ah,0ch

mov al,0

int 21h

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

Check\_key endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: Check\_die

;功能: 处理死亡的响应

;传递参数: SI,DI为当前蛇头部的坐标

;函数返回: 蛇参数的值

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Check\_die proc

push ax

push bx

push cx

push dx

push si

push di

cmp si , 50 ;检查是否到达边界

jz dead

cmp si , 255

jz dead

cmp di , 50

jz dead

cmp di , 255

jz dead

mov ax,si ;检查是否接触蛇身

mov dh,al ;DX中获得当前坐标

mov ax,di

mov dl,al

mov cx,0

mov cl, Len

sub cl,1

mov bx,offset Body

check\_die\_loop:

add bx , 2

mov ax , [bx]

cmp ax , dx ;取出蛇身信息进行比较

jz dead ;触及蛇身即死亡

loop check\_die\_loop

jmp next

dead:

mov gameover,1 ;死亡即赋gameover值1

next:

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

Check\_die endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: Check\_eat

;功能: 处理吃到食物的响应

;传递参数: SI,DI为当前蛇头部的坐标

;函数返回: cx

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Check\_eat proc

push ax

push bx

push dx

push si

push di

mov cx,0

mov ax , foodx

mov bx , foody

cmp ax , si

jz ok1

jmp check\_eat\_out

ok1: ;横坐标相等

cmp bx , di

jz ok2

jmp check\_eat\_out

ok2: ;纵坐标也相等

add point,1 ;加分

mov cx,1

check\_eat\_out:

pop di

pop si

pop dx

pop bx

pop ax

ret

Check\_eat endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: Run\_snake

;功能: 蛇运动

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

run\_snake proc

push ax

push bx

push cx

push dx

push si

push di

mov dx, 0005h

call SetGb

mov bl,white

lea bp,msg6

mov cx,30

call Msg

loop1:

call Delay

mov al,1

cmp al,gameover

jz stop

mov bx , offset Body

mov cx,0

mov cl,Len

add bx,cx

add bx,cx

sub bx,2

A1:

mov ax,[bx]

mov [bx+2],ax

sub bx,2

loop A1

addhead:

mov bx , offset Body ;增加新头部并改写数据

mov ax , [bx]

mov dx , Direction

add ah , dh ;坐标加方向

add al , dl ;坐标加方向

mov [bx] , ax ;存入头部

mov dx,0 ;为check\_die准备参数si,di

mov dl , ah

mov si , dx

mov dl , al

mov di , dx

call check\_die ;检查死亡

mov al,1

cmp al,gameover

jz stop

call check\_eat ;检查吃

cmp cx,1

jz A2

mov bx , offset Body

mov ax,0

add Len,1

mov al , Len

add bx , ax ;取到蛇尾

add bx , ax

sub bx , 2

mov ax , [bx]

mov cx,0

mov [bx],cx

mov dx,0

mov dl , ah

mov si , dx

mov dl , al

mov di , dx

sub Len,1

call clearcut

mov bx , offset Body

mov ax, [bx]

mov dx,0

mov dl , ah

mov si , dx

mov dl , al

mov di , dx

call cutsnake

jmp A3

A2:

add Len,1

mov bx , offset Body

mov ax, [bx]

mov dx,0

mov dl , ah

mov si , dx

mov dl , al

mov di , dx

call cutsnake ;画新头部

call getfood

A3:

call show\_point

call check\_key ;检查按键

jmp loop1 ;蛇运动循环DATAS SEGMENT

stop:

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

run\_snake endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: Init\_snake

;功能: 蛇初始化

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Init\_snake proc

push ax

push bx

push cx

push dx

push si

push di

mov al , 4

mov Len, al ;蛇的初始大小为4

mov ax,256 ;初始方向为右

mov Direction,ax

mov gameover,0

mov point,0

mov bx , offset Body

mov ch , 25 ;蛇的初始位置设为25，25

mov cl , 25

mov [bx] , cx

mov cx,3

init\_first: ;生成第一个蛇

mov ax , [bx]

mov dx , Direction

sub ah , dh ;坐标x减方向

sub al , dl ;坐标y减方向

add bx , 2

mov [bx] , ax ;保存一格蛇的信息

loop init\_first

init\_print: ;开始画第一个蛇

mov cx,0

mov cl, Len

mov dx,0

mov bx , offset Body

init\_print\_loop:

mov ax , [bx]

mov dl , ah

mov si , dx

mov dl , al

mov di , dx

call cutsnake ;画一格

add bx , 2

loop init\_print\_loop

Init\_over:

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

Init\_snake endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: clear\_area

;功能: 清空游戏区域

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

clear\_area proc

push ax

push bx

push cx

push dx

push si

push di

mov ax,0

mov dx,0

clear\_loop1: ;填充一遍黑颜色

cmp ax,50

jz clear\_loop1\_over

clear\_loop2:

cmp dx,50

jz clear\_loop2\_over

mov si,ax

mov di,dx

call clearcut

inc dx

jmp clear\_loop2

clear\_loop2\_over:

mov dx,0

inc ax

jmp clear\_loop1

clear\_loop1\_over:

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

clear\_area endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: HLine

;功能: 画横线

;传递参数:

; ah 0ch

; si 起点

; di 终点

; al 颜色号

; dx 行号

;函数返回: 空

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

HLine proc near

push ax

push bx

push cx

push dx

push si

push di

mov bx,1

push si

clc

sub si , di

jnc M1\_1 ;负数

jmp M1\_2 ;正数

M1\_1: neg bx

M1\_2: pop si

M1\_3: ;循环画线

mov cx , si

int 10h

add si , bx

cmp si , di

je M1\_4

jmp M1\_3

M1\_4:

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

HLine endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: VLine

;功能: 画竖线

;传递参数:

; ah 0ch

; si 起点

; di 终点

; al 颜色号

; cx 列号

;函数返回: 空

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

VLine proc near

push ax

push bx

push cx

push dx

push si

push di

mov bx , 1

push si

clc

sub si , di

jnc M2\_1 ;负数

jmp M2\_2 ;正数

M2\_1: neg bx

M2\_2: pop si

M2\_3: ;循环画线

mov dx , si

int 10h

add si , bx

cmp si , di

je M2\_4

jmp M2\_3

M2\_4:

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

VLine endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: Rec

;功能: 画矩形

;传递参数:

; cl 颜色

; si 左边

; bx 上边

; di 右边

; dx 下边

;函数返回: 空

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Rec proc near

push ax

push bx

push cx

push dx

push si

push di

mov ah , 0ch

mov al , cl

push dx

mov dx , bx

call HLine ;画上边线

pop dx

call HLine ;画下边线

mov cx , si

mov si , bx

push di

mov di , dx

call VLine ;画左边线

pop cx

inc di ;补右下角一点

call VLine ;画右边线

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

Rec endp

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

;函数名: Show\_point

;功能: 打印分数(进制转换)

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Show\_point proc

push ax

push bx

push cx

push dx

push si

push di

mov ax,0

mov al, point ;二进制数

mov bx, offset Score ;取得目标字符串地址

add bx, 7 ;更改的末位在字串中的位置（后面保留2个0）

HS:

mov cx,0ah

mov dx,0

div cx

add dl,30h

mov [bx],dl

sub bx,1

cmp ax,0

jz PR

jmp HS

PR:

mov dx,1810h

call SetGb ;设置光标

mov bl,green

lea bp,Score ;显示分数

mov cx,10

call Msg ;显示字符串

pop di

pop si

pop dx

pop cx

pop bx

pop ax

ret

Show\_point endp

end main