***Coal Lab final Project***

***Snake Game***

***Nasm code***

|  |
| --- |
| ;Printing a snake at random position and start moving |
|  |  | [org 0x0100] |
|  |  | jmp start |
|  |  | x\_pos: dw 0 |
|  |  | y\_pos: dw 0 |
|  |  | foodpos: dw 0 |
|  |  |  |
|  |  | welcome: db 'Welcome to Snake Game' |
|  |  | GroupMember1: db 'Group Members Talib Husain | 21F-9070 Saad Rehman | 21F-9640 Press Space to continue' |
|  |  | GameInstuction: db 'Welcome to snake Game Instructions The player uses the arrow keys to move a snake around the board Snake will eat food to increase its size and scores Press any key to continue' |
|  |  | GameOver: db 'Game Over' |
|  |  | currScore: db '0' |
|  |  | score: db 'Score: ' |
|  |  | snake: db 02,'\*','\*','\*','\*' |
|  |  | snake\_length: dw 5 |
|  |  | direction: db 2 |
|  |  | delaytime: dd 0 |
|  |  | ;----------------------------------------------------------------------------------- |
|  |  | clearscreen: |
|  |  | push es |
|  |  | push ax |
|  |  | push di |
|  |  | push cx |
|  |  | mov ax,0xb800 ; video memory address |
|  |  | mov es,ax |
|  |  | mov ax,0x0720 ; color code and space ASCII |
|  |  | mov di,0 |
|  |  | nextchar: |
|  |  | mov [es:di],ax |
|  |  | add di,2 |
|  |  | cmp di,4000 |
|  |  | jne nextchar |
|  |  |  |
|  |  | ;popping all values |
|  |  | pop cx |
|  |  | pop di |
|  |  | pop ax |
|  |  | pop es |
|  |  | ret |
|  |  | ;----------------------------------------------------------------------------------- |
|  |  | welcomeMsg: |
|  |  | push ax |
|  |  | push bx |
|  |  | push si |
|  |  | push di |
|  |  | push es |
|  |  | push cx |
|  |  |  |
|  |  | mov ax,0xb800 |
|  |  | mov es,ax |
|  |  | mov di,1180 |
|  |  | mov si,GroupMember1 |
|  |  | mov cx,84 |
|  |  | mov ah,0x04 |
|  |  | cld |
|  |  | newchar: |
|  |  | lodsb ; load next char in al |
|  |  | stosw ; print char/attribute pair |
|  |  |  |
|  |  | cmp cx,71 |
|  |  | jne x |
|  |  | add word di,120 |
|  |  |  |
|  |  | x: |
|  |  | cmp cx,47 |
|  |  | jne y |
|  |  | add word di,114 |
|  |  |  |
|  |  | y: |
|  |  | cmp cx,24 |
|  |  | jne next |
|  |  | add word di,274 |
|  |  |  |
|  |  | next: |
|  |  | loop newchar |
|  |  |  |
|  |  | pop cx |
|  |  | pop es |
|  |  | pop di |
|  |  | pop si |
|  |  | pop bx |
|  |  | pop ax |
|  |  | ret |
|  |  | ;----------------------------------------------------------------------------------- |
|  |  | Instruction: |
|  |  | push ax |
|  |  | push bx |
|  |  | push si |
|  |  | push di |
|  |  | push es |
|  |  | push cx |
|  |  |  |
|  |  | mov ax,0xb800 |
|  |  | mov es,ax |
|  |  | mov di,1180 |
|  |  | mov si,GameInstuction |
|  |  | mov cx,172 |
|  |  | mov ah,0x04 |
|  |  |  |
|  |  | newcharins: |
|  |  | lodsb ; load next char in al |
|  |  | stosw ; print char/attribute pair |
|  |  |  |
|  |  | cmp cx,152 |
|  |  | jne x2 |
|  |  | add word di,120 |
|  |  |  |
|  |  | x2: |
|  |  | cmp cx,138 |
|  |  | jne y2 |
|  |  | add word di,114 |
|  |  |  |
|  |  | y2: |
|  |  | cmp cx,91 |
|  |  | jne z2 |
|  |  | add word di,66 |
|  |  |  |
|  |  | z2: |
|  |  | cmp cx,43 |
|  |  | jne a2 |
|  |  | add word di,80 |
|  |  |  |
|  |  | a2: |
|  |  | cmp cx,23 |
|  |  | jne next2 |
|  |  | add word di,120 |
|  |  |  |
|  |  |  |
|  |  | next2: |
|  |  | loop newcharins |
|  |  |  |
|  |  | pop cx |
|  |  | pop es |
|  |  | pop di |
|  |  | pop si |
|  |  | pop bx |
|  |  | pop ax |
|  |  | ret |
|  |  | ;----------------------------------------------------------------------------------- |
|  |  | draw\_snake: |
|  |  | push bp |
|  |  | mov bp, sp |
|  |  | push ax |
|  |  | push bx |
|  |  | push si |
|  |  | push cx |
|  |  | push dx |
|  |  |  |
|  |  | mov si, [bp + 6] |
|  |  | mov cx, 5 |
|  |  | mov di, 1500 |
|  |  | mov ax, 0xb800 |
|  |  | mov es, ax |
|  |  |  |
|  |  | mov bx, [bp + 4] |
|  |  | mov ah, 0x04 |
|  |  | snake\_next\_part: |
|  |  | mov al, [si] |
|  |  | mov [es:di], ax |
|  |  | mov [bx], di |
|  |  | inc si |
|  |  | add bx, 2 |
|  |  |  |
|  |  | add di, 2 |
|  |  | loop snake\_next\_part |
|  |  |  |
|  |  | pop dx |
|  |  | pop cx |
|  |  | pop si |
|  |  | pop bx |
|  |  | pop ax |
|  |  | pop bp |
|  |  | ret 6 |
|  |  | ;----------------------------------------------------------------------------------- |
|  |  | printScore: |
|  |  | push bp |
|  |  | mov bp,sp |
|  |  | push ax |
|  |  | push bx |
|  |  | push cx |
|  |  | push si |
|  |  | push di |
|  |  | push es |
|  |  |  |
|  |  | cmp byte[currScore], 58 |
|  |  | jne s |
|  |  |  |
|  |  | mov byte[currScore],'A' |
|  |  | s: |
|  |  | mov si,score |
|  |  | mov ax,0xb800 |
|  |  | mov ah,0x07 |
|  |  | mov di,[bp+4] |
|  |  | mov cx,7 |
|  |  | p: |
|  |  | lodsb |
|  |  | stosw |
|  |  | loop p |
|  |  | mov al,[currScore] |
|  |  | mov [es:di],ax |
|  |  |  |
|  |  | pop es |
|  |  | pop di |
|  |  | pop si |
|  |  | pop cx |
|  |  | pop bx |
|  |  | pop ax |
|  |  | pop bp |
|  |  | ret 2 |
|  |  | ;----------------------------------------------------------------------------------- |
|  |  | move\_snake\_left: |
|  |  | push bp |
|  |  | mov bp, sp |
|  |  | push ax |
|  |  | push bx |
|  |  | push cx |
|  |  | push dx |
|  |  | push es |
|  |  | push di |
|  |  | push si |
|  |  | ;snake itself parts colision check |
|  |  | mov bx, [bp + 4] ; snake locations |
|  |  | mov dx, [bx] ; snake head |
|  |  |  |
|  |  | mov cx, [bp + 8]; len of snake |
|  |  | dec cx |
|  |  | sub dx, 2 ; dx = 1978 |
|  |  | check\_left\_colision: |
|  |  | cmp dx, [bx] |
|  |  | je no\_left\_movement |
|  |  | add bx, 2 |
|  |  | loop check\_left\_colision |
|  |  | left\_movement: |
|  |  | mov si, [bp + 6] ;snake |
|  |  | mov bx, [bp + 4] ;snake location |
|  |  | mov dx, [bx] |
|  |  | sub dx, 2 |
|  |  | mov di, dx |
|  |  |  |
|  |  | mov ax, 0xb800 |
|  |  | mov es, ax |
|  |  | mov ah, 0x04 |
|  |  | mov al, [si] |
|  |  | mov [es:di],ax ;snake head placed |
|  |  |  |
|  |  | mov cx, [bp + 8] |
|  |  | mov di, [bx] |
|  |  | inc si |
|  |  | mov ah, 0x04 |
|  |  | mov al, [si] |
|  |  | mov [es:di],ax |
|  |  | left\_location\_sort: |
|  |  | mov ax, [bx] |
|  |  | mov [bx], dx |
|  |  | mov dx, ax |
|  |  | add bx, 2 |
|  |  |  |
|  |  | loop left\_location\_sort |
|  |  | mov di, dx |
|  |  | mov ax, 0x0720 |
|  |  | mov [es:di], ax |
|  |  | jmp end1 |
|  |  | no\_left\_movement: |
|  |  | call over |
|  |  | end1: |
|  |  | pop si |
|  |  | pop di |
|  |  | pop es |
|  |  | pop dx |
|  |  | pop cx |
|  |  | pop bx |
|  |  | pop ax |
|  |  | pop bp |
|  |  | ret 6 |
|  |  | ;----------------------------------------------------------------------------------- |
|  |  | move\_snake\_up: |
|  |  | push bp |
|  |  | mov bp, sp |
|  |  | push ax |
|  |  | push bx |
|  |  | push cx |
|  |  | push dx |
|  |  | push es |
|  |  | push di |
|  |  | push si |
|  |  | ;snake\_parts colision detection |
|  |  | mov bx, [bp + 4] ;snake location |
|  |  | mov dx, [bx] |
|  |  |  |
|  |  | mov cx, [bp + 8] |
|  |  | dec cx |
|  |  |  |
|  |  | sub dx, 160 |
|  |  |  |
|  |  | check\_up\_colision: |
|  |  | cmp dx, [bx] |
|  |  | je no\_up\_movement |
|  |  | add bx, 2 |
|  |  | loop check\_up\_colision |
|  |  |  |
|  |  | upward\_movement: |
|  |  | mov si, [bp + 6] ;snake |
|  |  | mov bx, [bp + 4] ;snake location |
|  |  | mov dx, [bx] |
|  |  | sub dx, 160 |
|  |  | mov di, dx |
|  |  |  |
|  |  | mov ax, 0xb800 |
|  |  | mov es, ax |
|  |  | mov ah, 0x04 |
|  |  | mov al, [si] |
|  |  | mov [es:di],ax ;snake head placed |
|  |  |  |
|  |  | mov cx, [bp + 8] |
|  |  | mov di, [bx] |
|  |  | inc si |
|  |  | mov ah, 0x04 |
|  |  | mov al, [si] |
|  |  | mov [es:di],ax |
|  |  | up\_location\_sort: |
|  |  | mov ax, [bx] |
|  |  | mov [bx], dx |
|  |  | mov dx, ax |
|  |  | add bx, 2 |
|  |  |  |
|  |  | loop up\_location\_sort |
|  |  |  |
|  |  | mov di, dx |
|  |  | mov ax, 0x0720 |
|  |  | mov [es:di], ax |
|  |  | jmp end2 |
|  |  |  |
|  |  | no\_up\_movement: |
|  |  | call over |
|  |  | end2: |
|  |  | pop si |
|  |  | pop di |
|  |  | pop es |
|  |  | pop dx |
|  |  | pop cx |
|  |  | pop bx |
|  |  | pop ax |
|  |  | pop bp |
|  |  | ret 6 |
|  |  | ;----------------------------------------------------------------------------------- |
|  |  | move\_snake\_down: |
|  |  | push bp |
|  |  | mov bp, sp |
|  |  | push ax |
|  |  | push bx |
|  |  | push cx |
|  |  | push dx |
|  |  | push es |
|  |  | push di |
|  |  | push si |
|  |  | ;snake\_parts colision detection |
|  |  | mov bx, [bp + 4] ;snake location |
|  |  | mov dx, [bx] |
|  |  |  |
|  |  | mov cx, [bp + 8] |
|  |  | dec cx |
|  |  | add dx, 160 |
|  |  | check\_down\_colision: |
|  |  | cmp dx, [bx] |
|  |  | je no\_down\_movement |
|  |  | add bx, 2 |
|  |  | loop check\_down\_colision |
|  |  |  |
|  |  | downward\_movement: |
|  |  | mov si, [bp + 6] ;snake |
|  |  | mov bx, [bp + 4] ;snake location |
|  |  | mov dx, [bx] |
|  |  | add dx, 160 |
|  |  | mov di, dx |
|  |  |  |
|  |  | mov ax, 0xb800 |
|  |  | mov es, ax |
|  |  | mov ah, 0x04 |
|  |  | mov al, [si] |
|  |  | mov [es:di], ax ;snake head placed |
|  |  |  |
|  |  | mov cx, [bp + 8] ;snake length |
|  |  | mov di, [bx] |
|  |  | inc si |
|  |  | mov ah, 0x04 |
|  |  | mov al, [si] |
|  |  | mov [es:di],ax |
|  |  | down\_location\_sort: |
|  |  | mov ax, [bx] |
|  |  | mov [bx], dx |
|  |  | mov dx, ax |
|  |  | add bx, 2 |
|  |  | loop down\_location\_sort |
|  |  | mov di, dx |
|  |  | mov ax, 0x0720 |
|  |  | mov [es:di], ax |
|  |  | jmp end3 |
|  |  |  |
|  |  | no\_down\_movement: |
|  |  | call over |
|  |  | end3: |
|  |  |  |
|  |  | pop si |
|  |  | pop di |
|  |  | pop es |
|  |  | pop dx |
|  |  | pop cx |
|  |  | pop bx |
|  |  | pop ax |
|  |  | pop bp |
|  |  | ret 6 |
|  |  | ;----------------------------------------------------------------------------------- |
|  |  | move\_snake\_right: |
|  |  | push bp |
|  |  | mov bp, sp |
|  |  | push ax |
|  |  | push bx |
|  |  | push cx |
|  |  | push dx |
|  |  | push es |
|  |  | push di |
|  |  | push si |
|  |  | ;snake\_parts colision detection |
|  |  | mov bx, [bp + 4] ;snake location |
|  |  | mov dx, [bx] |
|  |  |  |
|  |  | mov cx, [bp + 8] |
|  |  | dec cx |
|  |  | add dx, 2 |
|  |  | check\_right\_colision: |
|  |  | cmp dx, [bx] |
|  |  | je no\_right\_movement |
|  |  | add bx, 2 |
|  |  | loop check\_right\_colision |
|  |  |  |
|  |  | right\_movement: |
|  |  | mov si, [bp + 6] ;snake |
|  |  | mov bx, [bp + 4] ;snake location |
|  |  | mov dx, [bx] |
|  |  | add dx, 2 |
|  |  | mov di, dx |
|  |  |  |
|  |  | mov ax, 0xb800 |
|  |  | mov es, ax |
|  |  | mov ah, 0x04 |
|  |  | mov al, [si] |
|  |  | mov [es:di], ax ;snake head placed |
|  |  |  |
|  |  | mov cx, [bp + 8] ;snake length |
|  |  | mov di, [bx] |
|  |  | inc si |
|  |  | mov ah, 0x04 |
|  |  | mov al, [si] |
|  |  | mov [es:di],ax |
|  |  | right\_location\_sort: |
|  |  | mov ax, [bx] |
|  |  | mov [bx], dx |
|  |  | mov dx, ax |
|  |  | add bx, 2 |
|  |  |  |
|  |  | loop right\_location\_sort |
|  |  | mov di, dx |
|  |  | mov ax, 0x0720 |
|  |  | mov [es:di], ax |
|  |  | jmp end4 |
|  |  | no\_right\_movement: |
|  |  | call over |
|  |  | end4: |
|  |  | pop si |
|  |  | pop di |
|  |  | pop es |
|  |  | pop dx |
|  |  | pop cx |
|  |  | pop bx |
|  |  | pop ax |
|  |  | pop bp |
|  |  | ret 6 |
|  |  | ;----------------------------------------------------------------------------------- |
|  |  | check\_death: |
|  |  | push ax |
|  |  | push di |
|  |  | push cx |
|  |  |  |
|  |  | mov ax, [snake\_locations] |
|  |  | cmp ax, 160 |
|  |  | jb finished |
|  |  | mov di, 160 |
|  |  | mov cx, 24 |
|  |  |  |
|  |  | check1: |
|  |  | cmp ax, di |
|  |  | je finished |
|  |  | add di, 158 |
|  |  | cmp ax, di |
|  |  | je finished |
|  |  | add di, 2 |
|  |  |  |
|  |  | loop check1 |
|  |  |  |
|  |  | mov di,3840 |
|  |  | cmp ax, di |
|  |  | ja finished |
|  |  | jmp else |
|  |  |  |
|  |  | finished: |
|  |  | call over |
|  |  | else: |
|  |  | pop cx |
|  |  | pop di |
|  |  | pop ax |
|  |  | ret |
|  |  | ;----------------------------------------------------------------------------------- |
|  |  | play\_game: |
|  |  | call clearscreen |
|  |  | call welcomeMsg |
|  |  | mov ah,00 |
|  |  | int 16h |
|  |  | call clearscreen |
|  |  | call Instruction |
|  |  | mov ah,00 |
|  |  | int 16h |
|  |  |  |
|  |  | call clearscreen |
|  |  | call draw\_border |
|  |  |  |
|  |  | push word [snake\_length] |
|  |  | mov bx, snake |
|  |  | push bx |
|  |  | mov bx, snake\_locations |
|  |  | push bx |
|  |  | call draw\_snake |
|  |  | call displayFood |
|  |  |  |
|  |  | repeat: |
|  |  | push 164 |
|  |  | call printScore |
|  |  | cmp byte[currScore],'5' |
|  |  | jae leve1 |
|  |  | mov dword[delaytime],80000 |
|  |  | leve1: |
|  |  | mov dword[delaytime],120000 |
|  |  |  |
|  |  | delay: |
|  |  | dec dword[delaytime] |
|  |  | cmp dword[delaytime],0 |
|  |  | jne delay |
|  |  |  |
|  |  | mov ah, 01h |
|  |  | int 16h |
|  |  | jz noKey |
|  |  | mov ah,0 |
|  |  | int 16h |
|  |  |  |
|  |  | cmp ah,0x48 ;up arrow |
|  |  | je up |
|  |  | cmp ah,0x4B ;left arrow |
|  |  | je left |
|  |  | cmp ah,0x4D ;right arrow |
|  |  | je right |
|  |  | cmp ah,0x50 ;down arrow |
|  |  | je down |
|  |  | cmp ah, ' ' |
|  |  | jne repeat |
|  |  | mov ah,0x4c |
|  |  | int 21h |
|  |  | je exit2 |
|  |  | exit2: |
|  |  | pop bx |
|  |  | pop ax |
|  |  | ret |
|  |  |  |
|  |  | noKey: |
|  |  | cmp byte[direction], 0 |
|  |  | je up |
|  |  | cmp byte[direction], 1 |
|  |  | je down |
|  |  | cmp byte[direction], 2 |
|  |  | je left |
|  |  | cmp byte[direction], 3 |
|  |  | je right |
|  |  |  |
|  |  | up: |
|  |  | mov byte[direction],0 |
|  |  | push word [snake\_length] |
|  |  | mov bx, snake |
|  |  | push bx |
|  |  | mov bx, snake\_locations |
|  |  | push bx |
|  |  | call move\_snake\_up |
|  |  | jmp new |
|  |  |  |
|  |  |  |
|  |  | down: |
|  |  | mov byte[direction],1 |
|  |  | push word [snake\_length] |
|  |  | mov bx, snake |
|  |  | push bx |
|  |  | mov bx, snake\_locations |
|  |  | push bx |
|  |  | call move\_snake\_down |
|  |  | jmp new |
|  |  |  |
|  |  |  |
|  |  | left: |
|  |  | mov byte[direction],2 |
|  |  | push word [snake\_length] |
|  |  | mov bx, snake |
|  |  | push bx |
|  |  | mov bx, snake\_locations |
|  |  | push bx |
|  |  | call move\_snake\_left |
|  |  | jmp new |
|  |  |  |
|  |  | right: |
|  |  | mov byte[direction],3 |
|  |  | push word [snake\_length] |
|  |  | mov bx, snake |
|  |  | push bx |
|  |  | mov bx, snake\_locations |
|  |  | push bx |
|  |  | call move\_snake\_right |
|  |  |  |
|  |  | new: |
|  |  | call check\_death |
|  |  |  |
|  |  | push ax |
|  |  | mov ax,word[foodpos] |
|  |  | cmp ax,[snake\_locations] |
|  |  | jne f |
|  |  | call displayFood |
|  |  | add word[snake\_length],1 |
|  |  | add byte[currScore],1 |
|  |  |  |
|  |  | f: |
|  |  | pop ax |
|  |  | jmp repeat |
|  |  | exit: |
|  |  | pop bx |
|  |  | pop ax |
|  |  | ret |
|  |  | ;----------------------------------------------------------------------------------- |
|  |  | displayFood: |
|  |  | push bp |
|  |  | push bx |
|  |  | push ax |
|  |  | push cx |
|  |  | push dx |
|  |  | push es |
|  |  | push di |
|  |  |  |
|  |  | l1: |
|  |  | MOV AH, 00h ; interrupts to get system time |
|  |  | INT 1AH ; CX:DX now hold number of clock ticks since midnight |
|  |  |  |
|  |  | mov ax, dx |
|  |  | xor dx, dx |
|  |  | mov cx, 25 |
|  |  | div cx |
|  |  |  |
|  |  | mov word[x\_pos],dx |
|  |  |  |
|  |  | MOV AH, 00h ; interrupts to get system time |
|  |  | INT 1AH ; CX:DX now hold number of clock ticks since midnight |
|  |  |  |
|  |  | mov ax, dx |
|  |  | xor dx, dx |
|  |  | mov cx, 80 |
|  |  | div cx ; here dx contains the remainder of the division - from 0 to 9 |
|  |  |  |
|  |  | mov word[y\_pos],dx |
|  |  |  |
|  |  | mov ax,[x\_pos] |
|  |  | mov bx,80 |
|  |  | mul bx |
|  |  | add ax,[y\_pos] |
|  |  | shl ax,1 |
|  |  | cmp ax,3840 |
|  |  | jg l1 |
|  |  |  |
|  |  | cmp ax,190 |
|  |  | jb l1 |
|  |  |  |
|  |  | mov word[foodpos],ax |
|  |  | mov di,ax |
|  |  | mov ax,0xb800 |
|  |  | mov es,ax |
|  |  | mov ax,0x0501 |
|  |  | mov [es:di],ax |
|  |  |  |
|  |  |  |
|  |  | pop di |
|  |  | pop es |
|  |  | pop dx |
|  |  | pop cx |
|  |  | pop ax |
|  |  | pop bx |
|  |  | pop bp |
|  |  | ret |
|  |  | ;----------------------------------------------------------------------------------- |
|  |  | draw\_border: |
|  |  | push ax |
|  |  | push bx |
|  |  | push es |
|  |  | push di |
|  |  | push si |
|  |  | push cx |
|  |  |  |
|  |  | mov ax,0xb800 |
|  |  | mov es,ax |
|  |  | mov di,0 |
|  |  |  |
|  |  | mov cx,80 |
|  |  | mov ah,0x02 |
|  |  | mov al,'=' |
|  |  | top\_border: |
|  |  | mov [es:di],ax |
|  |  | add di,2 |
|  |  | loop top\_border |
|  |  |  |
|  |  | mov cx,80 |
|  |  | mov di,3840 |
|  |  | mov al,'=' |
|  |  | bottom\_border: |
|  |  | mov [es:di],ax |
|  |  | add di,2 |
|  |  | loop bottom\_border |
|  |  |  |
|  |  | mov cx,24 |
|  |  | mov al,'|' |
|  |  | mov di,160 |
|  |  | left\_border: |
|  |  | mov [es:di],ax |
|  |  | add di,160 |
|  |  | loop left\_border |
|  |  |  |
|  |  | mov cx,24 |
|  |  | mov al,'|' |
|  |  | mov di,158 |
|  |  | right\_border: |
|  |  | mov [es:di],ax |
|  |  | add di,160 |
|  |  | loop right\_border |
|  |  |  |
|  |  | pop cx |
|  |  | pop si |
|  |  | pop di |
|  |  | pop es |
|  |  | pop bx |
|  |  | pop ax |
|  |  | ret |
|  |  | ;----------------------------------------------------------------------------------- |
|  |  | over: |
|  |  | push bp |
|  |  | mov bp,sp |
|  |  | push ax |
|  |  | push bx |
|  |  | push cx |
|  |  | push es |
|  |  | push si |
|  |  | push di |
|  |  | mov ax,0xb800 |
|  |  | mov ah,0x04 |
|  |  | mov si,GameOver |
|  |  | mov cx,9 |
|  |  | mov di,1510 |
|  |  | printMsg: |
|  |  | lodsb |
|  |  | stosw |
|  |  | loop printMsg |
|  |  | push 1670 |
|  |  | call printScore |
|  |  | pop di |
|  |  | pop si |
|  |  | pop es |
|  |  | pop cx |
|  |  | pop bx |
|  |  | pop ax |
|  |  | pop bp |
|  |  | mov ax, 0x4c00 |
|  |  | int 0x21 |
|  |  | ret |
|  |  | ;----------------------------------------------------------------------------------- |
|  |  | start: |
|  |  | call play\_game |
|  |  | mov ax,0x4c00 |
|  |  | int 0x21 |
|  |  | snake\_locations: dw 0 |