

COAL pROJECT

Pac-Man



MUhammad awais zahid 18f-0269

fahad usman 18f-0181

zulqarnain 18f-0375

Game Introduction

User can move the pacman in four directions: up, down, right & left. A map is shown t which will includes some walls, moving path, eatable things (do), & a pacman at its starting position. The pacman cannot cross the wall of maps (it is an obstacle). The user has to eat all the dots by moving the pacman to the appropriate location(s).. Pacman has Three lives if pacman touches mine he will lose his one life. Pacman will get 10 score on eating per dot and similarly his score will increase on basis of number of dots he will eat. As soon as the pacman will hit a mine, it will explode the pacman and the pacman will lose a **life.**

**Features of Pac-Man:**

* Display Menu :

Display a menu on the screen which will guide the user how to move the pacman (up, down, left, right).

* Display Map:

Display a map for pac-man in Which Pac-Man can move. The walls of map is obstacles for Pac-Man. Pac-Man cannot move outside walls.

* Pacman Movements

Allow the pacman to move on the map as per the instructions you have given in your menu

* Working of pac-Man

Allow the pacman to eat everything (dots, mines) on the map except Wall wall

* Getting Scores

So 1 points will be added to the score as as Pac-Man eats a dot.

Code

Include Irvine32.inc

Include macros.inc

buffer\_size = 2000

.data

;lebel 30 dots

lev\_1 db " ###################",0dh,0ah

db " #P . . • . . .#",0dh,0ah

db " ####### • . #",0dh,0ah

db " # . • .• #",0dh,0ah

db " #. . .#######",0dh,0ah

db " #. .•. . • . #",0dh,0ah

db " #.. ######### . #",0dh,0ah

db " #. . • . . . . #",0dh,0ah

db " #. . .•. . #",0dh,0ah

db " ###################",0

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

lev\_2 db " ###################",0dh,0ah

db " #P . . • . . • .#",0dh,0ah

db " ####### • . . #",0dh,0ah

db " # . • .•. .•.#",0dh,0ah

db " ######## • #######",0dh,0ah

db " # • . ...#",0dh,0ah

db " #.. #########.• . #",0dh,0ah

db " #. . • . . . . #",0dh,0ah

db " # . • . • . .#",0dh,0ah

db " ###################",0

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

lev\_3 db " ###################",0dh,0ah

db " #P . . • . . • .#",0dh,0ah

db " ####### • • . . #",0dh,0ah

db " # . . . •. .•.#",0dh,0ah

db " ########## ########",0dh,0ah

db " # • . . . •. .#",0dh,0ah

db " #•. #########.• . #",0dh,0ah

db " #. •. • . . • . #",0dh,0ah

db " # • . • . . .#",0dh,0ah

db " ###################",0

score\_str db " \_\_\_Score is = ",0

life\_str db " \_\_\_Life is = ",0

score dw 0

life dw 3

home db "\_\_\_\_\_Press 'h' back to HOME\_\_\_\_",0

level dw 0

len = 499

cstr1 db len dup(?)

cstr2 db len dup(?)

cstr3 db len dup(?)

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

.code

main PROC

mov ecx,len

mov edi,offset cstr1

mov esi,offset lev\_1

L\_1:

mov al,[esi]

mov [edi],al

inc esi

inc edi

loop L\_1

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

mov ecx,len

mov edi,offset cstr2

mov esi,offset lev\_2

L\_2:

mov al,[esi]

mov [edi],al

inc esi

inc edi

loop L\_2

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

mov ecx,len

mov edi,offset cstr3

mov esi,offset lev\_3

L\_3:

mov al,[esi]

mov [edi],al

inc esi

inc edi

loop L\_3

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

call display

call menu\_proc

call control\_menu

exit

main endp

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

display proc

mov eax,yellow

call settextcolor

mWrite <" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*",0dh,0ah>

mWrite <" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ",0dh,0ah>

mWrite <" | | ",0dh,0ah>

mWrite <" | |ppppppp A CCCCC MMMM MMMMM A |N N| | ",0dh,0ah>

mWrite <" | |pp pp A A C M MM MM M A A |N N N| |",0dh,0ah>

mWrite <" | |ppppppp AAAAAA C M M M AAAAAA |N N N| |",0dh,0ah>

mWrite <" | |pp A A C M M A A |N N N| |",0dh,0ah>

mWrite <" | |pp A A CCCCC M M A A |N N| |",0dh,0ah>

mWrite <" |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| ",0dh,0ah>

mWrite <" ",0dh,0ah>

mWrite <" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*",0dh,0ah>

call crlf

call crlf

mWrite <" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_MADE BY\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ",0ah,0dh>

call crlf

call crlf

mov eax,lightcyan

call settextcolor

mWrite <" Domintaors",0dh,0ah>

call crlf

mWrite <" # MUHAMMAD AWAIS ZAHID",0dh,0ah>

mWrite <" # ZULQARNAIN HAIDER",0dh,0ah>

mWrite <" # FAHAD USMAN",0dh,0ah>

call crlf

call crlf

mov eax,LightMagenta

call settextcolor

mWrite <" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ",0dh,0ah>

mWrite <" \* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*",0dh,0ah>

mwrite <" \* | | \*",0dh,0ah>

mWrite <" \* | Game is Loading . . . . . . . | \*",0dh,0ah>

mWrite <" \* |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| \*",0dh,0ah>

mWrite <" \* \*",0dh,0ah>

mWrite <" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ",0dh,0ah>

;;;;;;;;;;;;;;;;;;for delay

;mov eax,3000

;call delay

call readchar

call CLRSCR

ret

display endp

control\_menu proc

cmp level,1

je l1

cmp level,2

je l2

cmp level,3

je l3

l1:

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

mov eax,yellow

call settextcolor

mov esi,offset lev\_1

add esi,82

jmp again

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

l2:

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

mov eax,lightgreen

call settextcolor

mov esi,offset lev\_2

add esi,82

jmp again

l3:

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

mov eax,red

call settextcolor

mov esi,offset lev\_3

add esi,82

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

again:

mGotoxy 0,0 ;goto condion overriding the same string

call crlf

call crlf

call crlf

call crlf

call crlf

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

cmp level,1

je l\_1

cmp level,2

je l\_2

cmp level,3

je l\_3

l\_1:

mwrite<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ LEVEL 1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_",0dh,0ah>

call crlf

call crlf

mov edx,offset lev\_1

call writestring

jmp next\_go

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

l\_2:

mwrite<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ LEVEL 2 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_",0dh,0ah>

call crlf

call crlf

mov edx,offset lev\_2

call writestring

jmp next\_go

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

l\_3:

mwrite<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ LEVEL 3 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_",0dh,0ah>

call crlf

call crlf

mov edx,offset lev\_3

call writestring

next\_go:

call crlf

call crlf

call crlf

mov eax,Cyan

call settextcolor

mov edx,offset score\_str

call writestring

mov eax,white

call settextcolor

mov ax,score

call writedec

call crlf

call crlf

mov eax,Cyan

call settextcolor

mov edx,offset life\_str

call writestring

mov eax,white

call settextcolor

mov ax,life

call writedec

call crlf

call crlf

mov eax,lightred

call settextcolor

mov edx,offset home

call writestring

mov eax,lightgray

call settextcolor

;;;;;;;;;;;;;;;;;

cmp score,30

je won1

;again:

; mov dl,row ;row

; mov dh,col ;column

; call Gotoxy

call readchar

cmp al,'h'

je hom

cmp al,'d'

je d

cmp al,'s'

je s

cmp al,'w'

je w

cmp al,'a'

je a

jmp again

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

d:

inc esi

mov al,'.'

cmp [esi],al

;;;;;;;;;;;;;;;;;;;;;checking dot;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

je do\_d

mov al,' '

cmp [esi],al

;;;;;;;;;;;;;;;;;;;;;checking space;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

je do\_d

;;;;;;;;;;;;;;;;;;;;;checking mine;;;;;;;;;;;;;;;;;;;;;;;;;

mov al,'•'

cmp [esi],al

je dec\_d

;;;;;;;;;;;;;;;;;;;;;;else again;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

dec esi

jmp again

dec\_d:

dec esi

jmp mine\_1

do\_d:

mov al,' '

cmp [esi],al

je go\_d

inc score

go\_d:

dec esi

mov al,' '

mov [esi],al

inc esi

mov al,'P'

mov [esi],al

jmp again

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

s:

add esi,51

mov al,'.'

cmp [esi],al

je do\_s

mov al,' '

cmp [esi],al

je do\_s

;;;;;;;;;;;;;;;;;;;checking mine;;;;;;;;;;;;;;;;;;;

mov al,'•'

cmp [esi],al

je sub\_s

;;;;;;;;;;;;;;;;;;;;else;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

sub esi,51

jmp again

sub\_s:

sub esi,51

jmp mine\_1

do\_s:

mov al,' '

cmp [esi],al

je go\_s

inc score

go\_s:

sub esi,51

mov al,' '

mov [esi],al

add esi,51

mov al,'P'

mov [esi],al

jmp again

;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

w:

sub esi,51

mov al,'.'

cmp [esi],al

je do\_w

mov al,' '

cmp [esi],al

je do\_w

;;;;;;;;;;;;;;;;;;;checking mine;;;;;;;;;;;;;;;;;;;

mov al,'•'

cmp [esi],al

je add\_w

;;;;;;;;;;;;;;;;;;;;else;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

add esi,51

jmp again

add\_w:

add esi,51

jmp mine\_1

do\_w:

mov al,' '

cmp [esi],al

je go\_w

inc score

go\_w:

add esi,51

mov al,' '

mov [esi],al

sub esi,51

mov al,'P'

mov [esi],al

jmp again

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

a:

dec esi

mov al,'.'

cmp [esi],al

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;checking dot;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

je do\_a

mov al,' '

cmp [esi],al

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;checking space;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

je do\_a

;;;;;;;;;;;;;;;;;;;;;checking mine;;;;;;;;;;;;;;;;;;;;;;;;;

mov al,'•'

cmp [esi],al

je inc\_a

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;else again;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

inc esi

jmp again

inc\_a:

inc esi

jmp mine\_1

do\_a:

mov al,' '

cmp [esi],al

je go\_a

inc score

go\_a:

inc esi

mov al,' '

mov [esi],al

dec esi

mov al,'P'

mov [esi],al

jmp again

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

mine\_1:

mov al,' '

mov [esi],al

call mine\_proc

won1:

call won\_proc

hom:

call menu\_proc

ret

control\_menu endp

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

mine\_proc proc

dec life

cmp life,0

je zero

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

cmp level,1

je l\_1

cmp level,2

je l\_2

cmp level,3

je l\_3

l\_1:

mov esi,offset lev\_1

add esi,82

jmp space\_

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

l\_2:

mov esi,offset lev\_2

add esi,82

jmp space\_

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

l\_3:

mov esi,offset lev\_3

add esi,82

jmp space\_

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

space\_:

mov al,'P'

mov [esi],al

call control\_menu

zero:

call loss\_proc

ret

mine\_proc endp

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

won\_proc proc

call crlf

call crlf

mov eax,Lightblue

call settextcolor

mWrite <" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CONGRATULATIONS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ",0dh,0ah>

mWrite <" \* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*",0dh,0ah>

mwrite <" \* | | \*",0dh,0ah>

mWrite <" \* | YOU WON THE GAME | \*",0dh,0ah>

mWrite <" \* |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| \*",0dh,0ah>

mWrite <" \* \*",0dh,0ah>

mWrite <" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ",0dh,0ah>

mov eax,2000

call delay

call menu\_proc

exit

ret

won\_proc endp

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

loss\_proc proc

call crlf

call crlf

mov eax,Lightblue

call settextcolor

mWrite <" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TRY AGAIN\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ",0dh,0ah>

mWrite <" \* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \*",0dh,0ah>

mwrite <" \* | | \*",0dh,0ah>

mWrite <" \* | YOU LOSS THE GAME | \*",0dh,0ah>

mWrite <" \* |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_| \*",0dh,0ah>

mWrite <" \* \*",0dh,0ah>

mWrite <" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ",0dh,0ah>

mov eax,2000

call delay

call menu\_proc

exit

ret

loss\_proc endp

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

menu\_proc proc

mov eax,LightMagenta

call settextcolor

again1:

mov score,0

mov life,3

call clrscr

call crlf

call crlf

call crlf

call crlf

mwrite<" 1.Play Game",0dh,0ah>

call crlf

mwrite<" 2.Instruction",0dh,0ah>

call crlf

mwrite<" 3.Levels",0dh,0ah>

call crlf

mwrite<" 4.EXIT",0dh,0ah>

call crlf

mov al,0

call readdec

cmp al,1

je play\_1\_auto

cmp al,3

je lev\_lab

cmp al,2

je instruction

cmp al,4

je exit\_lab

jmp again1

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

play\_1\_auto:

mov level,1

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

mov ecx,len

mov esi,offset lev\_1

mov edi,offset cstr1

L\_\_1:

mov al,[edi]

mov [esi],al

inc esi

inc edi

loop L\_\_1

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

call clrscr

call control\_menu

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

instruction:

call clrscr

mwrite<" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ",00dh,0ah>

mwrite<"1.| Display a menu on the screen which will guide the user how to move the pacman (up 'W', down 'S', left 'A', right 'D'). |",0dh,0ah>

mwrite<" | |",0dh,0ah>

mwrite<"2.| Display a map for pac-man in Which Pac-Man can move. The walls of map is obstacles for Pac-Man. Pac-Man cannot move outside walls. |",0dh,0ah>

mwrite<" | |",0dh,0ah>

mwrite<"3.| Allow the pacman to move on the map as per the instructions you have given in your menu |",0dh,0ah>

mwrite<" | |",0dh,0ah>

mwrite<"4.| Allow the pacman to eat everything (dots '.', mines '•') on the map except Wall . |",0dh,0ah>

mwrite<" | |",0dh,0ah>

mwrite<"5.| So 1 points will be added to the score as as Pac-Man eats a dot. |",0dh,0ah>

mwrite<" | |",0dh,0ah>

mwrite<" | \_\_\_\_\_PRESS 6 TO GOTO TO BACK MENU\_\_\_\_\_\_ |",0dh,0ah>

mwrite<" |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|",0dh,0ah>

mov al,0

call readdec

cmp al,6

je again1

jmp instruction

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

lev\_lab:

call clrscr

mwrite<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_select the level you want to play\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_",0dh,0ah>

CALL CRLF

CALL CRLF

CALL CRLF

mwrite<" press 1. LEVEL 1 ",0dh,0ah>

CALL CRLF

mwrite<" press 2. LEVEL 2 ",0dh,0ah>

CALL CRLF

mwrite<" press 3. LEVEL 3 ",0dh,0ah>

CALL CRLF

mov ax,0

call readdec

mov bx,ax

cmp bx,1

jne ll\_2

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

mov ecx,len

mov esi,offset lev\_1

mov edi,offset cstr1

L\_\_\_1:

mov al,[edi]

mov [esi],al

inc esi

inc edi

loop L\_\_\_1

jmp n

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

ll\_2:

cmp bx,2

jne ll\_3

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

mov ecx,len

mov esi,offset lev\_2

mov edi,offset cstr2

L\_\_2:

mov al,[edi]

mov [esi],al

inc esi

inc edi

loop L\_\_2

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

jmp n

ll\_3:

cmp bx,3

jne lev\_lab

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

mov ecx,len

mov esi,offset lev\_3

mov edi,offset cstr3

L\_\_3:

mov al,[edi]

mov [esi],al

inc esi

inc edi

loop L\_\_3

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

n:

mov level,bx

call clrscr

call control\_menu

exit\_lab:

exit

ret

menu\_proc endp

END main

Output









