National University of Computer and Emerging Sciences



Laboratory Manual

for

Computer Organization and Assembly Language

|  |  |
| --- | --- |
| Lab Instructor(s) | Uzair Tahir |
| Semester | Fall 2018 |

**COAL LAB PROJECT**

**Department of Computer Science**

**CODE**

INCLUDE Irvine32.inc

INCLUDE Macros.inc

INCLUDE VirtualKeys.inc

.data

delayOnScreen EQU 750 ; FOR DELAY ON CONSOLE MEANS (1000 == 1SEC) => (750/1000 = 0.75 SEC)

endl EQU <0Ah,0Dh> ; FOR NEWLINE

DASH BYTE '-'

;;------------------------------------------

;;--------------INTRO IMAGE-----------------

;;------------------------------------------

HANGMAN LABEL BYTE

BYTE "########################################################################################################################", endl

BYTE " #### ##", endl

BYTE " ## ##", endl

BYTE " ## ## ######## ####### ## ######### #### #### ######## ####### ## ", endl

BYTE " ## ## ########## ####### ## ########### ###### ####### ########## ####### ## ", endl

BYTE " ########### ## ## ## ## ## ## ## ##### ## ## ## ## ## ## ", endl

BYTE " ########### ## ## ## ## ## ## ##### ## ### ## ## ## ## ## ## ", endl

BYTE " ## ## ########## ## ## ## ## ####### ## # ## ########## ## ## ## ", endl

BYTE " ## ## ########## ## ## ## ## ## ## ## ########## ## ## ## ", endl

BYTE " ## ## ## ## ## ####### ########## ## ## ## ## ## ####### ", endl

BYTE " ## #### ### ### ## ####### ######## #### #### ### ### ## ####### ", endl

BYTE endl

BYTE "########################################################################################################################", 0ah,0dh,0

;;------------------------------------------

;;------------------------------------------

TOTAL\_COUNTS BYTE 10 ; Total Number of Guess

END\_IN BYTE 0 ; End\_in means How much Guesses User Take To Win The Game

TOTAL\_GUESSED BYTE 0

;;------------------------------------------

;;--------------QUESTIONS-------------------

;;------------------------------------------

WORD1 BYTE "ICECREAM" ,0

WORD2 BYTE "TABLE" ,0

WORD3 BYTE "CHAIR" ,0

WORD4 BYTE "BOTTLE" ,0

WORD5 BYTE "HAIR" ,0

WORD6 BYTE "GLASS" ,0

WORD7 BYTE "MOBILE" ,0

WORD8 BYTE "SCHOOL" ,0

WORD9 BYTE "BIKE" ,0

;;------------------------------------------

;;------------------------------------------

;;------------------------------------------

STR\_LENGTH\_GUESS BYTE 0

STR\_WORD BYTE 10 DUP (?), 0

USED\_LETTERS BYTE 10 DUP (?), 0

IS\_FOUND BYTE 0

;;------------------------------------------

;;------------GAME WON----------------------

;;------------------------------------------

GOODGAME\_0 LABEL BYTE

BYTE "+------+ ",endl

BYTE "| | ",endl

BYTE "| ",endl

BYTE "| O ",endl

BYTE "| /|\ ",endl

BYTE "| / \ ",endl

BYTE "+------------+ ",endl

BYTE "| YOU WIN | ",endl

BYTE "+------------+ ",0

GOODGAME\_1 LABEL BYTE

BYTE "+------+ ",endl

BYTE "| | ",endl

BYTE "| ",endl

BYTE "| O\_ ",endl

BYTE "| /| ",endl

BYTE "| / \ ",endl

BYTE "+------------+ ",endl

BYTE "| YOU WIN | ",endl

BYTE "+------------+ ",0

GOODGAME\_2 LABEL BYTE

BYTE "+------+ ",endl

BYTE "| | ",endl

BYTE "| ",endl

BYTE "| O/ ",endl

BYTE "| /| ",endl

BYTE "| / \ ",endl

BYTE "+------------+ ",endl

BYTE "| YOU WIN | ",endl

BYTE "+------------+ ",0

GOODGAME\_3 LABEL BYTE

BYTE "+------+ ",endl

BYTE "| | ",endl

BYTE "| ",endl

BYTE "| O\_ ",endl

BYTE "| /| ",endl

BYTE "| / \ ",endl

BYTE "+------------+ ",endl

BYTE "| YOU WIN | ",endl

BYTE "+------------+ ",0

;;------------------------------------------

;;-------------GAMEOVER---------------------

;;------------------------------------------

GAMEOVER\_0 LABEL BYTE

BYTE "+------+ ",endl

BYTE "| | ",endl

BYTE "| O ",endl

BYTE "| /|\ ",endl

BYTE "| / \ ",endl

BYTE "| ",endl

BYTE "+------------+ ",endl

BYTE "| YOU DIE | ",endl

BYTE "+------------+ ",0

GAMEOVER\_1 LABEL BYTE

BYTE "+------+ ",endl

BYTE "| / ",endl

BYTE "| \_O ",endl

BYTE "| \_/\ ",endl

BYTE "| \ ",endl

BYTE "| ",endl

BYTE "+------------+ ",endl

BYTE "| YOU DIE | ",endl

BYTE "+------------+ ",0

GAMEOVER\_2 LABEL BYTE

BYTE "+------+ ",endl

BYTE "| | ",endl

BYTE "| O ",endl

BYTE "| /|\ ",endl

BYTE "| / \ ",endl

BYTE "| ",endl

BYTE "+------------+ ",endl

BYTE "| YOU DIE | ",endl

BYTE "+------------+ ",0

GAMEOVER\_3 LABEL BYTE

BYTE "+------+ ",endl

BYTE "| \ ",endl

BYTE "| O\_ ",endl

BYTE "| /\\_ ",endl

BYTE "| / ",endl

BYTE "| ",endl

BYTE "+------------+ ",endl

BYTE "| YOU DIE | ",endl

BYTE "+------------+ ",0

;;------------------------------------------

;;------------CURRENT LIVES LEFT------------

;;------------------------------------------

LIVES\_6 LABEL BYTE

BYTE "+------+ ",endl

BYTE "| | ",endl

BYTE "| ",endl

BYTE "| ",endl

BYTE "| ",endl

BYTE "| ",endl

BYTE "+------------+ ",endl

BYTE "| | ",endl

BYTE "+------------+ ",0

LIVES\_5 LABEL BYTE

BYTE "+------+ ",endl

BYTE "| | ",endl

BYTE "| O ",endl

BYTE "| ",endl

BYTE "| ",endl

BYTE "| ",endl

BYTE "+------------+ ",endl

BYTE "| | ",endl

BYTE "+------------+ ",0

LIVES\_4 LABEL BYTE

BYTE "+------+ ",endl

BYTE "| | ",endl

BYTE "| O ",endl

BYTE "| | ",endl

BYTE "| ",endl

BYTE "| ",endl

BYTE "+------------+ ",endl

BYTE "| | ",endl

BYTE "+------------+ ",0

LIVES\_3 LABEL BYTE

BYTE "+------+ ",endl

BYTE "| | ",endl

BYTE "| O ",endl

BYTE "| /| ",endl

BYTE "| ",endl

BYTE "| ",endl

BYTE "+------------+ ",endl

BYTE "| | ",endl

BYTE "+------------+ ",0

LIVES\_2 LABEL BYTE

BYTE "+------+ ",endl

BYTE "| | ",endl

BYTE "| O ",endl

BYTE "| /|\ ",endl

BYTE "| ",endl

BYTE "| ",endl

BYTE "+------------+ ",endl

BYTE "| | ",endl

BYTE "+------------+ ",0

LIVES\_1 LABEL BYTE

BYTE "+------+ ",endl

BYTE "| | ",endl

BYTE "| O ",endl

BYTE "| /|\ ",endl

BYTE "| / ",endl

BYTE "| ",endl

BYTE "+------------+ ",endl

BYTE "| | ",endl

BYTE "+------------+ ",0

LIVES\_0 LABEL BYTE

BYTE "+------+ ",endl

BYTE "| | ",endl

BYTE "| O ",endl

BYTE "| /|\ ",endl

BYTE "| / \ ",endl

BYTE "| ",endl

BYTE "+------------+ ",endl

BYTE "| | ",endl

BYTE "+------------+ ",0

;;------------------------------------------

;;------------------------------------------

.code

;;------------------------------------------

;;-------------HANGMAN INTRO----------------

;;------------------------------------------

HANGMAN\_INTRO proc

;;------------------------------------------

MOV ECX, 7

L1:

mGotoxy 0, 0

MOV EDX, offset HANGMAN

call WriteString

MOV EAX, delayOnScreen

call Delay

mGotoxy 0, 0

MOV EAX, WHITE

SUB EAX, ECX

call SetTextColor

loop L1

MOV EAX, YELLOW

call SetTextColor

call clrscr

MOV EDX, offset HANGMAN

call WriteString

;;------------------------------------------

ret

HANGMAN\_INTRO endp

;;------------------------------------------

;;------------------------------------------

;;--------------GENERATE RANDOM WORD--------

;;------------------------------------------

GENERATE\_WORD PROC

;;------------------------------------------

MOV EAX, 9

call Randomize

call randomrange

inc EAX

CMP EAX, 1

JE \_WORD1

CMP EAX, 2

JE \_WORD2

CMP EAX, 3

JE \_WORD3

CMP EAX, 4

JE \_WORD4

CMP EAX, 5

JE \_WORD5

CMP EAX, 6

JE \_WORD6

CMP EAX, 7

JE \_WORD7

CMP EAX, 8

JE \_WORD8

CMP EAX, 9

JE \_WORD8

\_EXIT\_GENERATE\_P:

ret

\_WORD1:

MOV STR\_LENGTH\_GUESS,LENGTHOF WORD1

MOV ESI, OFFSET WORD1

JMP \_EXIT\_GENERATE\_P

\_WORD2:

MOV STR\_LENGTH\_GUESS,LENGTHOF WORD2

MOV ESI, OFFSET WORD2

JMP \_EXIT\_GENERATE\_P

\_WORD3:

MOV STR\_LENGTH\_GUESS,LENGTHOF WORD3

MOV ESI, OFFSET WORD3

JMP \_EXIT\_GENERATE\_P

\_WORD4:

MOV STR\_LENGTH\_GUESS,LENGTHOF WORD4

MOV ESI, OFFSET WORD4

JMP \_EXIT\_GENERATE\_P

\_WORD5:

MOV STR\_LENGTH\_GUESS,LENGTHOF WORD5

MOV ESI, OFFSET WORD5

JMP \_EXIT\_GENERATE\_P

\_WORD6:

MOV STR\_LENGTH\_GUESS,LENGTHOF WORD6

MOV ESI, OFFSET WORD6

JMP \_EXIT\_GENERATE\_P

\_WORD7:

MOV STR\_LENGTH\_GUESS,LENGTHOF WORD7

MOV ESI, OFFSET WORD7

JMP \_EXIT\_GENERATE\_P

\_WORD8:

MOV STR\_LENGTH\_GUESS,LENGTHOF WORD8

MOV ESI, OFFSET WORD8

JMP \_EXIT\_GENERATE\_P

\_WORD9:

MOV STR\_LENGTH\_GUESS,LENGTHOF WORD9

MOV ESI, OFFSET WORD9

JMP \_EXIT\_GENERATE\_P

;;------------------------------------------

GENERATE\_WORD ENDP

;;------------------------------------------

;;------------------------------------------

;;------------GAME WON PROCEDURE------------

;;------------------------------------------

WON\_GAME PROC

;;------------------------------------------

MOV ECX, 4

mGotoxy 0,0

MOV EDX, OFFSET GOODGAME\_0

call WriteString

mGotoxy 0,0

MOV EAX,GREEN+(black\*16)

call SetTextColor

MOV EAX, DelayOnScreen

call Delay

mGotoxy 0,0

MOV EDX, OFFSET GOODGAME\_1

call WriteString

mGotoxy 0,0

MOV EAX,YELLOW+(black\*16)

call SetTextColor

MOV EAX, DelayOnScreen

call Delay

mGotoxy 0,0

MOV EDX, OFFSET GOODGAME\_2

call WriteString

mGotoxy 0,0

MOV EAX,BLUE+(black\*16)

call SetTextColor

MOV EAX, DelayOnScreen

call Delay

mGotoxy 0,0

MOV EDX, OFFSET GOODGAME\_3

call WriteString

mGotoxy 0,0

MOV EAX,RED+(black\*16)

call SetTextColor

MOV EAX, DelayOnScreen

call Delay

;;------------------------------------------

ret

WON\_GAME ENDP

;;------------------------------------------

;;------------------------------------------

;;-------------GAME OVER PROCEDURE----------

;;------------------------------------------

OVER PROC

;;------------------------------------------

MOV ECX, 4

mWrite<"Game Over . . .">

mGotoxy 0,1

MOV EDX, OFFSET GAMEOVER\_0

call WriteString

mGotoxy 0,1

MOV EAX,GREEN+(black\*16)

call SetTextColor

MOV EAX, DelayOnScreen

call Delay

mGotoxy 0,1

MOV EDX, OFFSET GAMEOVER\_1

call WriteString

mGotoxy 0,1

MOV EAX,YELLOW+(black\*16)

call SetTextColor

MOV EAX, DelayOnScreen

call Delay

mGotoxy 0,1

MOV EDX, OFFSET GAMEOVER\_2

call WriteString

mGotoxy 0,1

MOV EAX,BLUE+(black\*16)

call SetTextColor

MOV EAX, DelayOnScreen

call Delay

mGotoxy 0,1

MOV EDX, OFFSET GAMEOVER\_3

call WriteString

mGotoxy 0,1

MOV EAX,RED+(black\*16)

call SetTextColor

MOV EAX, DelayOnScreen

call Delay

;;------------------------------------------

ret

OVER ENDP

;;------------------------------------------

;;------------------------------------------

;;-------------CURRENT CHARACTERS-----------

;;------------------------------------------

CURRENT\_CHARACTERS PROC

;;------------------------------------------

MOV EBX, OFFSET STR\_WORD

MOV ECX, 0

MOV CL, STR\_LENGTH\_GUESS

SUB CL,1

GUESS\_Q:

MOV AL, [ebx]

call WriteChar

ADD EBX, TYPE STR\_WORD

loop GUESS\_Q

;;------------------------------------------

ret

CURRENT\_CHARACTERS ENDP

;;------------------------------------------

;;------------------------------------------

;;------------------------------------------

CHECK\_EXISTANCE PROC

;;------------------------------------------

MOV EDX, ESI

MOV IS\_FOUND, 0

MOV ECX, 0

MOV CL, STR\_LENGTH\_GUESS

MOV EBX, OFFSET STR\_WORD

CHECKING\_EXISTANCE1:

CMP AL, [EDX]

JE UPDATE\_ARRAY

CHECK1:

ADD EBX, TYPE STR\_WORD

ADD EDX, TYPE STR\_WORD

loop CHECKING\_EXISTANCE1

MOV DL, IS\_FOUND

CMP DL, 1

JE \_END

inc TOTAL\_GUESSED

\_END:

ret

UPDATE\_ARRAY:

MOV [EBX], AL

MOV IS\_FOUND, 1

JMP CHECK1

;;------------------------------------------

CHECK\_EXISTANCE ENDP

;;------------------------------------------

main PROC

\_START\_OF\_GAME\_:

MOV EAX,0

MOV EBX,0

MOV ECX,0

MOV EDX,0

call HANGMAN\_INTRO

call GENERATE\_WORD

MOV EDX, OFFSET STR\_WORD

MOV CL, STR\_LENGTH\_GUESS

MOV BL, DASH

dec ECX

CLEAR\_STR:

MOV [EDX], EBX

ADD EDX, TYPE STR\_WORD

loop CLEAR\_STR

MOV ECX, 0

MOV CL, 10

MOV EAX, 0

START\_GAME:

MOV END\_IN, CL

call crlf

JMP CURRENT\_IMG

GAME1:

call crlf

call CURRENT\_CHARACTERS

call crlf

mWrite< "Enter Your Guess : ">

call ReadChar

call WriteChar

CMP AL, 'a'

JGE LOWERCASE

RE\_LOOP:

CMP AL, 'A'

JGE CHECKLETTER

RELOOP\_2:

LOOPING:

JMP CHECK\_IS\_ALL\_FOUND

ALL\_DONE:

call crlf

MOV CL, END\_IN

loop START\_GAME

CMP CL, 0

JNE GAMEWON

JMP \_GAMEOVER\_

CURRENT\_IMG:

MOV DL, TOTAL\_GUESSED

CMP DL, 3

JLE GAME2

LIVE1:

CMP DL, 4

JG LIVE2

MOV EDX,OFFSET LIVES\_1

JMP JUMPGAME1

LIVE2:

CMP DL, 5

JG LIVE3

MOV EDX,OFFSET LIVES\_2

JMP JUMPGAME1

LIVE3:

CMP DL, 6

JG LIVE4

MOV EDX,OFFSET LIVES\_3

JMP JUMPGAME1

LIVE4:

CMP DL, 7

JG LIVE5

MOV EDX,OFFSET LIVES\_4

JMP JUMPGAME1

LIVE5:

CMP DL, 8

JG LIVE6

MOV EDX,OFFSET LIVES\_5

JMP JUMPGAME1

LIVE6:

CMP DL, 9

JG \_GAMEOVER\_

MOV EDX,OFFSET LIVES\_6

call WriteString

JMP GAME1

GAME2:

MOV EDX,OFFSET LIVES\_0

JUMPGAME1:

call WriteString

JMP GAME1

LOWERCASE:

CMP AL, 'z'

JG RE\_LOOP

SUB AL , 20h

JMP RE\_LOOP

CHECKLETTER:

CMP AL,'Z'

JG RELOOP\_2

call CHECK\_EXISTANCE

JMP LOOPING

CHECK\_IS\_ALL\_FOUND:

MOV EBX, OFFSET STR\_WORD

MOV ECX, 0

MOV CL, STR\_LENGTH\_GUESS

MOV EAX,0

SUB CL, 1

CHECK\_IS\_ALL:

MOV AL, [EBX]

CMP AL, DASH

JE ALL\_DONE

ADD EBX, TYPE STR\_WORD

loop CHECK\_IS\_ALL

JMP GAMEWON

\_GAMEOVER\_:

MOV EAX, DelayOnScreen

call Delay

call clrscr

call OVER

call OVER

call OVER

JMP ASK\_AGAIN

GAMEWON:

MOV EAX, DelayOnScreen

call Delay

call clrscr

call WON\_GAME

call WON\_GAME

call WON\_GAME

;;------------------------------------------

;;------------IF USER WANTS TO PLAYAGIN-----

;;------------------------------------------

ASK\_AGAIN:

call clrscr

MOV EAX, YELLOW

call SetTextColor

mWrite<"Do You Want To Play Again :- ">

MOV EAX, 0

call ReadChar

call WriteChar

CMP AL, 'y'

JE \_START\_OF\_GAME\_

CMP AL, 'Y'

JE \_START\_OF\_GAME\_

\_END\_HANGMAN:

call clrscr

mWrite<"Thanks For Playing Game . . . ">

call clrscr

exit

main endp

END main



   