# National University of Computer and Emerging Sciences



Coal Lab Project

# Computer Organization and Assembly Language

**Project Name: - Word Hunt**

**Group Members**

**Muhammad Awais 18F-0109**

**Makhdoom Haider 18F-0257**

**Umer Akram 18F-0187**

Department of Computer Science

# Introduction: -

Word hunt or Word Find is a word puzzle game that consists of different letters making a hidden word placed vertically, horizontally or diagonally.  The objective of this puzzle is to find and mark all the words hidden inside the box in the given time.

# Features: -

1: -Game will Ask For player name.

2: -Random Letters will Appear on the screen having hidden words in it.

3: - Player will find those Hidden words in the game in limited Tries.

4: - Every time player will find a word it will appear on the side of the screen with congratulations and supportive phrases.

5: - If player find all the hidden words, game will show player’s score according to the Tries taken by the player to find those words.

6: -Player Can ask for a hint if he/she is unable to find a word.

7: -There will be limited number of hints every Time player start new game.

8: -If Player didn’t find all given words in required Tries game will automatically over showing player current score.

9: - Game will ask if player want to end or want to start new game.

10: - Every time player start new game new words will appear on random location on the screen.

11: -Fun to play and Relaxing😊

Code: -

include irvine32.inc

.data

table byte 0

name1 byte "Please Enter Your Name :",0

name\_input byte 100 dup (?),0

welcome byte "Welcome To WordHunt",0ah,0dh,0

start byte "Press 1 to start new game ", 0ah,0dh,"Press 2 To know game rules",0ah,0dh,"Press 0 to Exit",0ah,0dh,0

start1 byte ?

save\_esi1 dword 0

word1 byte "compute",0

word2 byte"Success",0

word3 byte"winner",0

word4 byte"Faisalabad",0

word5 byte"competition",0

word6 byte"Random",0

word7 byte "whisper",0

word8 byte"organ",0

word9 byte"exhibition",0

word10 byte"protest",0

word11 byte"presentation",0

word12 byte"discovery",0

word13 byte "blackmail",0

word14 byte"regular",0

word15 byte"scatter",0

word16 byte"withdraw",0

word17 byte"accumulation",0

word18 byte"imagine",0

result byte 140 dup(?),0

count dword ?

save\_esi dword ?

Score byte 0

input byte "Please enter the Word you Found or Enter hint to get a hint or Enter 0 to Return In Main Menu: ",0

player\_input byte 100 dup(?)

input\_length byte 0

warning byte "Please Remember that each hint will decrement your score by 2 :",0ah,0dh,0

hint\_count byte 0

found\_words byte 0

invalid\_count byte 0

Congrats byte "Congratulations! You Found Correct word. We incremented your Score By 5",0ah,0dh,0

Error byte "OOPS ! Wrong Input Please try Again. We decremented your Score By 3 ",0ah,0dh,0

hint1 byte "HINT = Try to trace small letters it will always help ",0ah,0dh,0

hint2 byte "HINT = Some words might Have some alphabets replaced with 0",0ah,0dh,0

hint3 byte "HINT = Word You are trying to find might be placed Diagonaly ",0ah,0dh,0

hint4 byte "HINT = Focus on Capital And Small Letters and Enter the Word Accordingly ",0ah,0dh,0

score\_str byte "Your score is : ",0

check\_hint byte "hint"

wait1 byte "Please Wait . . . . .",0ah,0dh,0

help byte "1:- Try to Locate hidden Words from the Given Table.",0ah,0dh,"2:- You Have To Find Atleast 5 Words To win the Game",0ah,0dh,"3:- Enter the correct word to Get 5 Score(8 Tries)",0ah,0dh,"4:- Enter hint to Get a hint (It will decease your score by 2) ",0ah,0dh,"5:- If you Entered A Wrong Word it will decrement Your Score by 3 ",0ah,0dh," Good Luck !!!!!",0ah,0dh,0

bool byte 0

game\_count byte 0

thank byte "Thank You for Playing our Game",0ah,0dh,0

played byte "To play More Please Restart game",0ah,0dh,0

invalid\_input byte "Invalid Input",0ah,0dh,0

from\_table byte "Your Guess is not a proper(Meaning Full) word ",0ah,0dh,"Please Try Again and Remember We decremented Your Score by 3",0ah,0dh,0

not\_from\_table byte "The Word you Entered is not Even exist in the Table Please have a close look at the Table and try again" ,0ah,0dh ," Remember We decremented Your Score by 3",0ah,0dh,0

maximum byte "Congratulations on Finding all 5 Recquired Words. You are a Genius !!!!!!",0ah,0dh,0

max\_try byte "Sorry You already Tried maximum time(8) please start New Game",0ah,0dh,0

total\_inputs dword 0

.code

main proc

call randomize

mov edx, offset name1

call writestring

mov edx, offset name\_input

mov ecx,100

call readstring

call main\_menu

exit

main endp

main\_menu proc ;Function to Take input from user

pop save\_esi

call clrscr

mov dh,1

mov dl,50

call gotoxy

mov edx,offset welcome

call writestring

j0:

cmp table,3

je exit2

mov edx,offset start

call writestring

call readchar

call writechar

call crlf

cmp al,'0'

jne j1

jmp exit1

j1:

cmp al,'1'

jne j2

mov hint\_count,0

mov found\_words,0

mov invalid\_count,0

mov score,0

call Rand\_table

call take\_input

jmp j0

j2:

cmp al,'2'

jne invalid\_entry

call clrscr

mov edx,offset help

call writestring

jmp j0

invalid\_entry:

mov edx,offset invalid\_input

call writestring

jmp j0

j3:

exit2:

mov edx,offset played

call writestring

exit1:

mov edx,offset thank

call writestring

push save\_esi

ret

main\_menu endp

Rand\_table proc ;Function to pick random table

call clrscr

mov ecx,lengthof result

mov esi,offset result

call ini\_str

mov al,table

call writedec

call crlf

cmp al,0

jne j1

inc table

call table1

jmp exit1

j1:

cmp al,1

jne j2

inc table

call table2

jmp exit1

j2:

inc table

call table3

exit1:

mov esi,offset result

call print\_table

ret

Rand\_table endp

ini\_str proc ; initializing string by random alphabets pointed by esi,having length in ecx

l1:

call RAND\_U

mov [esi],al

inc esi

loop l1

ret

ini\_str endp

RAND\_U proc ;Function to creat Random Alphabate

mov eax,0

mov eax, 26

call randomrange

add al,'A'

ret

RAND\_U endp

take\_input proc ;Function to take Gussed Word from user

pop save\_esi1

mov bool, 1

mov ecx,8

l1:

mov total\_inputs,ecx

mov al,bool

cmp al,1

jne j1

pushf

cmp found\_words,5

jae j0

call input\_word

popf

mov ecx,total\_inputs

loop l1

mov edx,offset max\_try

call writestring

jmp j1

j0:

mov edx,offset maximum

call writestring

mov bx,70

call print\_space

call print\_score

j1:

push save\_esi1

ret

take\_input endp

Table1 proc ;Table 1 inserting first Six Words

mov eax,49

mov ebx,lengthof word5-1

mov esi,offset word5

call insert\_hr\_string

mov eax,122

mov ebx,lengthof word3-1

mov esi,offset word3

call insert\_hr\_string

mov eax,90

mov ebx,lengthof word4-1

mov esi,offset word4

call insert\_hr\_string

mov eax,9

mov ebx,lengthof word2-1

mov esi,offset word2

call insert\_vr\_string

mov eax,20

mov ebx,lengthof word6-1

mov esi,offset word6

call insert\_vr\_string

mov eax,0

mov ebx,lengthof word1-1

mov esi,offset word1

call insert\_dia\_string

ret

Table1 endp

Table2 proc ;Table 2 inserting 2nd Six Words

mov eax,120

mov ebx,lengthof word11-1

mov esi,offset word11

call insert\_hr\_string

mov eax,60

mov ebx,lengthof word9-1

mov esi,offset word9

call insert\_hr\_string

mov eax,27

mov ebx,lengthof word12-1

mov esi,offset word12

call insert\_hr\_string

mov eax,19

mov ebx,lengthof word7-1

mov esi,offset word7

call insert\_vr\_string

mov eax,6

mov ebx,lengthof word10-1

mov esi,offset word10

call insert\_vr\_string

mov eax,13

mov ebx,lengthof word8-1

mov esi,offset word8

call insert\_dia\_string

ret

Table2 endp

Table3 proc ;Table 3 inserting 3rd Six Words

mov eax,86

mov ebx,lengthof word17-1

mov esi,offset word17

call insert\_hr\_string

mov eax,41

mov ebx,lengthof word13-1

mov esi,offset word13

call insert\_hr\_string

mov eax,62

mov ebx,lengthof word16-1

mov esi,offset word16

call insert\_hr\_string

mov eax,18

mov ebx,lengthof word14-1

mov esi,offset word14

call insert\_vr\_string

mov eax,1

mov ebx,lengthof word15-1

mov esi,offset word15

call insert\_dia\_string

mov eax,11

mov ebx,lengthof word18-1

mov esi,offset word18

call insert\_dia\_string

ret

Table3 endp

print\_space proc ;Function to print Spaces BX time

movzx ecx,bx

mov al,' '

loopspace:

call writechar

loop loopspace

ret

print\_space endp

print\_under proc ;Function to print underscore BX time

movzx ecx,bx

mov al,'\_'

loopunder:

call writechar

loop loopunder

ret

print\_under endp

print\_table proc ; Function To print our Main Table & string pointed by esi

mov bx,70

call print\_space

call print\_score

movzx ecx,bx

mov bx,15 ;Printing First line

call print\_space

mov bx,41

call print\_under

call crlf

;starting main table from here

mov ecx,7

mainloop:

mov count,ecx

mov bx,15

call print\_space

mov ecx,20

l1:

mov al,'|'

call writechar

mov al,[esi]

call writechar

inc esi

loop l1

mov al,'|'

call writechar

call crlf

cmp count,1

je exitloop

mov bx,15

call print\_space

mov al,'|'

call writechar

mov bx,20

call empty\_line

call crlf

mov ecx,count

loop mainloop

exitloop:

mov bx,15

call print\_space

mov bx,41

call last\_line

call crlf

ret

print\_table endp

empty\_line proc ;Function to print Empty Line Bx time

movzx ecx,bx

loopminus:

mov al,'-'

call writechar

mov al,'|'

call writechar

loop loopminus

ret

empty\_line endp

last\_line proc ;Function to print Last Line Bx time

movzx ecx,bx

mov al,'\*'

loopasteric:

call writechar

loop loopasteric

ret

last\_line endp

;Function to Insert Word string horizontal pointed by esi,bx having size of the word string

insert\_hr\_string proc ;and eax size from where insertion start

mov edi,(offset result)

add edi,eax

movzx ecx,bx

cld

movstring:

movsb

loop movstring

ret

insert\_hr\_string endp

insert\_vr\_string proc ;and eax size from where insertion start

mov edi,(offset result)

add edi,eax

cld

movzx ecx,bx

movstring:

movsb

add edi,19

loop movstring

ret

insert\_vr\_string endp

insert\_dia\_string proc ;and eax size from where insertion start

mov edi,(offset result)

add edi,eax

cld

movzx ecx,bx

movstring:

movsb

add edi,20

loop movstring

ret

insert\_dia\_string endp

print\_table1 proc ; Function to print table 1

call table1

mov esi,offset result

call print\_table

ret

print\_table1 endp

print\_table2 proc ; Function to print table 2

call table2

mov esi,offset result

call print\_table

ret

print\_table2 endp

print\_table3 proc ; Function to print table 3

call table3

mov esi,offset result

call print\_table

ret

print\_table3 endp

print\_score proc ;Function to calculate Score

mov eax,0

mov al,5

mul found\_words

mov score,al

mov al,2

mul hint\_count

sub score,al

mov al,3

mul invalid\_count

sub score,al

cmp score,0

;jl j1

mov al,score

jmp exit2

j1:

mov al,0

exit2:

mov edx,offset score\_str

call writestring

movsx eax,al

call writeint

call crlf

ret

print\_score endp

input\_word proc ;Function to Take input from user

mov edx,offset input

call writestring

mov ecx,lengthof player\_input

mov edx,offset player\_input

call readstring

mov input\_length,al

call check\_input

ret

input\_word endp

check\_input proc ;Function to Check input entered user

mov esi,offset player\_input

mov al,'0'

cmp al,[esi]

jne j1

mov bool,0

jmp exit1

j1:

mov edi, offset check\_hint

mov ecx,lengthof check\_hint

cld

l1:

cmpsb

jne j2

loop l1

call print\_hint

jmp exit1

j2:

mov al,table

cmp al,1

jne j3

call cmp\_table1

jmp exit1

j3:

cmp al,2

jne j4

call cmp\_table2

jmp exit1

j4:

call cmp\_table3

exit1:

ret

check\_input endp

print\_hint proc ; Function to print random hint

mov edx,offset warning

call writestring

mov eax, 4

call randomrange

inc al

cmp al,1

jne j1

mov edx,offset hint1

call writestring

jmp exit1

j1:

cmp al,2

jne j2

mov edx,offset hint2

call writestring

jmp exit1

j2:

cmp al,3

jne j3

mov edx,offset hint3

call writestring

jmp exit1

j3:

mov edx,offset hint4

call writestring

exit1:

inc hint\_count

call crlf

mov bx,70

call print\_space

call print\_score

call crlf

ret

print\_hint endp

print\_congrats proc ;Function to print congratulations

mov edx,offset congrats

call writestring

inc found\_words

mov edx,offset wait1

call writestring

mov eax,1000

call delay

ret

print\_congrats endp

zero\_word proc ;mov 0 in the string pointed by edi having length in ecx

cld

mov al,'0'

rep stosb

ret

zero\_word endp

cmp\_table1 proc ; Function to cmpare words of table1

mov esi,offset player\_input

mov edi,offset word1

mov ecx,lengthof word1

l1:

cmpsb

jne j1

loop l1

call print\_congrats

mov edi,offset word1

mov ecx,lengthof word1

call zero\_word

jmp exit1

j1:

mov esi,offset player\_input

mov edi,offset word2

mov ecx,lengthof word2

l2:

cmpsb

jne j2

loop l2

call print\_congrats

mov edi,offset word2

mov ecx,lengthof word2

call zero\_word

jmp exit1

j2:

mov esi,offset player\_input

mov edi,offset word3

mov ecx,lengthof word3

l3:

cmpsb

jne j3

loop l3

call print\_congrats

mov edi,offset word3

mov ecx,lengthof word3

call zero\_word

jmp exit1

j3:

mov esi,offset player\_input

mov edi,offset word4

mov ecx,lengthof word4

l4:

cmpsb

jne j4

loop l4

call print\_congrats

mov edi,offset word4

mov ecx,lengthof word4

call zero\_word

jmp exit1

j4:

mov esi,offset player\_input

mov edi,offset word5

mov ecx,lengthof word5

l5:

cmpsb

jne j5

loop l5

mov edi,offset word5

mov ecx,lengthof word5

call zero\_word

call print\_congrats

jmp exit1

j5:

mov esi,offset player\_input

mov edi,offset word6

mov ecx,lengthof word6

l6:

cmpsb

jne j6

loop l6

call print\_congrats

mov edi,offset word6

mov ecx,lengthof word6

call zero\_word

jmp exit1

j6:

inc invalid\_count

call check\_Random\_input

exit1:

call print\_table1

ret

cmp\_table1 endp

cmp\_table2 proc ; Function to cmpare words of table2

mov esi,offset player\_input

mov edi,offset word7

mov ecx,lengthof word7

l1:

cmpsb

jne j1

loop l1

call print\_congrats

mov edi,offset word7

mov ecx,lengthof word7

call zero\_word

jmp exit1

j1:

mov esi,offset player\_input

mov edi,offset word8

mov ecx,lengthof word8

l2:

cmpsb

jne j2

loop l2

call print\_congrats

mov edi,offset word8

mov ecx,lengthof word8

call zero\_word

jmp exit1

j2:

mov esi,offset player\_input

mov edi,offset word9

mov ecx,lengthof word9

l3:

cmpsb

jne j3

loop l3

call print\_congrats

mov edi,offset word9

mov ecx,lengthof word9

call zero\_word

jmp exit1

j3:

mov esi,offset player\_input

mov edi,offset word10

mov ecx,lengthof word10

l4:

cmpsb

jne j4

loop l4

call print\_congrats

mov edi,offset word10

mov ecx,lengthof word10

call zero\_word

jmp exit1

j4:

mov esi,offset player\_input

mov edi,offset word11

mov ecx,lengthof word11

l5:

cmpsb

jne j5

loop l5

mov edi,offset word11

mov ecx,lengthof word11

call zero\_word

call print\_congrats

jmp exit1

j5:

mov esi,offset player\_input

mov edi,offset word12

mov ecx,lengthof word12

l6:

cmpsb

jne j6

loop l6

call print\_congrats

mov edi,offset word12

mov ecx,lengthof word12

call zero\_word

jmp exit1

j6:

inc invalid\_count

call check\_Random\_input

exit1:

call print\_table2

ret

cmp\_table2 endp

cmp\_table3 proc ; Function to cmpare words of table3

mov esi,offset player\_input

mov edi,offset word13

mov ecx,lengthof word13

l1:

cmpsb

jne j1

loop l1

call print\_congrats

mov edi,offset word13

mov ecx,lengthof word13

call zero\_word

jmp exit1

j1:

mov esi,offset player\_input

mov edi,offset word14

mov ecx,lengthof word14

l2:

cmpsb

jne j2

loop l2

call print\_congrats

mov edi,offset word14

mov ecx,lengthof word14

call zero\_word

jmp exit1

j2:

mov esi,offset player\_input

mov edi,offset word15

mov ecx,lengthof word15

l3:

cmpsb

jne j3

loop l3

call print\_congrats

mov edi,offset word15

mov ecx,lengthof word15

call zero\_word

jmp exit1

j3:

mov esi,offset player\_input

mov edi,offset word16

mov ecx,lengthof word16

l4:

cmpsb

jne j4

loop l4

call print\_congrats

mov edi,offset word16

mov ecx,lengthof word16

call zero\_word

jmp exit1

j4:

mov esi,offset player\_input

mov edi,offset word17

mov ecx,lengthof word17

l5:

cmpsb

jne j5

loop l5

mov edi,offset word17

mov ecx,lengthof word17

call zero\_word

call print\_congrats

jmp exit1

j5:

mov esi,offset player\_input

mov edi,offset word18

mov ecx,lengthof word18

l6:

cmpsb

jne j6

loop l6

call print\_congrats

mov edi,offset word18

mov ecx,lengthof word18

call zero\_word

jmp exit1

j6:

inc invalid\_count

call check\_Random\_input

exit1:

call print\_table3

ret

cmp\_table3 endp

check\_Random\_input proc ; Function to cmpare words of table3

mov ecx,lengthof result

mov eax,0

l1:

mov count,ecx

mov esi,offset player\_input

mov edi,offset result

add edi,eax

movzx ecx,input\_length

cld

repe cmpsb

je found

inc eax

mov ecx,count

loop l1

mov edx,offset not\_from\_table

call writestring

jmp exit1

found:

mov edx,offset from\_table

call writestring

exit1:

ret

check\_Random\_input endp

end main

Output: -





















