INCLUDE Irvine32.inc

INCLUDE macros.inc

.data

Caption BYTE "Error Found!",0

HelloMsg BYTE "Entered File Doesn't Exit.",0dh,0ah

BYTE " ",0dh,0ah

BYTE "Click Ok To Continue...",0

Caption1 BYTE "Instructions!",0

Instruction BYTE "[1]. Enter File Name With Extension.",0dh,0ah

BYTE " OR",0dh,0ah

BYTE "[2]. Enter Valid File.",0

Caption2 BYTE "Error Found!",0

BYTE " ",0dh,0ah

Option\_num BYTE "Enter Option 1 and 2 Only.",0

GOO BYTE ?

string111 BYTE "ANTO",0

string121 BYTE "SYNO",0

string5 BYTE "W",0dh,0ah

string6 BYTE "F",0

string7 BYTE "( F)",0

string8 BYTE "Press [1].Search For A Word",0

string9 BYTE "Press [2].Exit",0

string10 BYTE "Enter Your Choice:- ",0

string11 BYTE 177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177,177

chuice BYTE ?

decide BYTE ?

choice BYTE "(a). Antonyms (s). Synonyms",0

BlueTextOnGray = White + (Blue \* 16)

DefaultColor = lightGray + (black \* 16)

BUFFER\_SIZE = 100000

buffer BYTE BUFFER\_SIZE DUP(?)

filename BYTE "pera.txt",0

fileHandle HANDLE ?

input byte 15 dup(0)

nulstr BYTE 15 dup(0)

counter DWORD 0

wordsize DWORD 0

position DWORD 0

A BYTE "A.txt",0

B BYTE "B.txt",0

CO BYTE "C.txt",0

D BYTE "D.txt",0

E BYTE "E.txt",0

F BYTE "F.txt",0

G BYTE "G.txt",0

H BYTE "H.txt",0

I BYTE "I.txt",0

J BYTE "J.txt",0

K BYTE "K.txt",0

L BYTE "L.txt",0

M BYTE "M.txt",0

N BYTE "N.txt",0

O BYTE "O.txt",0

P BYTE "P.txt",0

Q BYTE "Q.txt",0

R BYTE "R.txt",0

S BYTE "S.txt",0

T BYTE "T.txt",0

U BYTE "U.txt",0

V BYTE "V.txt",0

W BYTE "W.txt",0

X BYTE "X.txt",0

Y BYTE "Y.txt",0

Z BYTE "Z.txt",0

File\_Name BYTE "Enter File Name: ",0

Question BYTE "Enter Word To Search It Meaning,Synonyms,and Antonyms",0

Enter\_Word BYTE "Enter Word: ",0

buffSize = 100000

string1 BYTE "===============",0

string2 BYTE 177,177,177,177,177,177,177,177,177,177,177,177,177,177,177

string4 BYTE "Loading...",0

String3 BYTE 186

string12 BYTE "Thank You!",0

.code

main PROC

Enter\_Again:

;For screen background color

mov eax, BlueTextOnGray

call SetTextColor

call Clrscr

;end

;Display screen

mov ecx,LENGTHOF string5

mov esi,0

mov dh,4

mov dl,50

l1:

movzx eax,string5[esi]

Call gotoxy

Call WriteChar

mov eax,20

Call Delay

inc esi

inc dl

loop l1

mov ecx,LENGTHOF string6

mov esi,0

mov dh,6

mov dl,71

l2:

movzx eax,string6[esi]

Call gotoxy

Call WriteChar

mov eax,20

Call Delay

inc esi

inc dl

loop l2

mov ecx,LENGTHOF string7

mov esi,0

mov dh,8

mov dl,55

l3:

movzx eax,string7[esi]

Call Gotoxy

Call WriteChar

mov eax,20

Call Delay

inc esi

inc dl

loop l3

mov edx,0

mov dh,14

mov dl,65

Call Gotoxy

mov edx,OFFSET string11

Call WriteString

mov edx,0

mov dh,16

mov dl,65

Call Gotoxy

mov edx,OFFSET string8

Call WriteString

mov dh,18

mov dl,65

Call Gotoxy

mov edx,OFFSET string9

Call WriteString

mov edx,0

mov dh,20

mov dl,65

Call Gotoxy

mov edx,OFFSET string11

Call WriteString

;For error handling

;takes input again

mov dh,23

mov dl,65

Call Gotoxy

mov edx,OFFSET string10

Call WriteString

Call ReadDec

Call Clrscr

cmp eax,1

je Word\_Search

cmp eax,2

je Program\_End

mov ebx,OFFSET Caption2

mov edx,OFFSET Option\_num

Call MsgBox

jmp Enter\_Again

Call Crlf

Call Crlf

mov dh,25

mov dl,55

Call Gotoxy

Call WaitMsg

;======================================================.

;start of word searching

Word\_Search:

Call Clrscr

Call My\_Instruction

Call WaitMsg

Call Clrscr

start:

mov edx,OFFSET File\_Name

Call WriteString

;takes file input

mov edx,OFFSET filename

mov ecx,SIZEOF filename

;Call ReadString

mov edx,OFFSET filename ; Open the file for input.

Call OpenInputFile

mov fileHandle,eax

cmp eax,INVALID\_HANDLE\_VALUE ;Checking For Errors.

JNE File\_Is\_Ok

;mWrite <"Cannot open file",0dh,0ah> ; Need To Change 1

jmp quit

inc ecx

File\_Is\_Ok:

mov edx,OFFSET buffer ; Read the file into a buffer.

mov ecx,BUFFER\_SIZE

Call ReadFromFile

JNC check\_buffer\_size ; error reading?

;mWrite "Error reading file. " ; Need To Change 2

jmp Close\_File

check\_buffer\_size:

cmp eax,BUFFER\_SIZE ; buffer large enough?

jb buf\_size\_ok ; yes

;mWrite <"Error: Buffer too small for the file",0dh,0ah> ; and quit ; Need to change 3;pop up message.

jmp quit

buf\_size\_ok:

mov buffer[eax],0 ; insert null terminator

;mWrite "File size: "

call WriteDec ; display file size

call Crlf

Call Clrscr

;mWrite <"Buffer:",0dh,0ah,0dh,0ah> ; Display the buffer.

mov edx,OFFSET buffer ; display the buffer

Call WriteString

Call Crlf

jmp Close\_File

quit:

mov ebx,OFFSET Caption

mov edx,OFFSET HelloMsg

Call MsgBox

mov ebx,OFFSET Caption1

mov edx,OFFSET Instruction

Call MsgBox

Call Clrscr

jmp start

Close\_File:

mov eax,fileHandle

Call CloseFile

mov edx,0

mov eax,0

mov ebx,0

mov ecx,0

mov esi,0

Call Crlf

mov edx,OFFSET Question

call writestring

call crlf

Call Crlf

;=====TAKES INPUT====

l12::

mov edx,OFFSET choice

Call WriteString

mov ecx,lengthof chuice

mov edx,offset chuice

call readstring

call CHOICE\_PROC

mov decide,al

mov edx,OFFSET Enter\_Word

Call WriteString

mov ecx,lengthof input

mov edx,offset input

call readstring

;call Smaller\_to\_Capital

call Text\_Search ;Word Search

call crlf

Program\_End:

mov edx,OFFSET string12

Call WriteString

Exit

main ENDP

;

CHOICE\_PROC PROC

cmp chuice,91

ja S\_input

cmp chuice,'A' ;Capital\_input 1.Synonym 2.Antonym

je Antonym

mov al,1

jmp exits

S\_input:

cmp chuice,'a' ;Small\_input 1.Synonym 2.Antonym

je Antonym

mov al,1

jmp exits

Antonym:

mov al,2

jmp exits

exits:

ret

CHOICE\_PROC endP

;==================================================;

Smaller\_to\_Capital proc

mov ecx,lengthof input

mov esi,0

L1:

cmp input[esi],0

je back

cmp input[esi],90

ja S\_t\_L

inc esi

cmp ecx,0

je back

loop L1

S\_T\_L:

sub input[esi],32

inc esi

jmp L1

call crlf

back:

;mov edx,OFFSET input

;Call WriteString

;Call Crlf

Smaller\_to\_Capital endp

;===================================================;

Text\_Search PROC

mov edx,0

mov eax,0

mov ebx,0

mov ecx,0

mov esi,0

mov al,input[0]

cmp al,'A'

jne Word\_B

mov edx,OFFSET A

call OpenInputFile

jmp Word\_Meaning

Word\_B:

cmp al,'B'

jne Word\_C

mov edx,OFFSET B

call OpenInputFile

jmp Word\_Meaning

Word\_C:

cmp al,'C'

jne Word\_D

mov edx,OFFSET CO

call OpenInputFile

jmp Word\_Meaning

Word\_D:

cmp al,'D'

jne Word\_E

mov edx,OFFSET D

call OpenInputFile

jmp Word\_Meaning

Word\_E:

cmp al,'E'

jne Word\_F

mov edx,OFFSET E

call OpenInputFile

jmp Word\_Meaning

Word\_F:

cmp al,'F'

jne Word\_G

mov edx,OFFSET F

call OpenInputFile

jmp Word\_Meaning

Word\_G:

cmp al,'G'

jne Word\_H

mov edx,OFFSET G

call OpenInputFile

jmp Word\_Meaning

Word\_H:

cmp al,'H'

jne Word\_I

mov edx,OFFSET H

call OpenInputFile

jmp Word\_Meaning

Word\_I:

cmp al,'I'

jne Word\_J

mov edx,OFFSET I

call OpenInputFile

jmp Word\_Meaning

Word\_J:

cmp al,'J'

jne Word\_K

mov edx,OFFSET J

call OpenInputFile

jmp Word\_Meaning

Word\_K:

cmp al,'K'

jne Word\_L

mov edx,OFFSET K

call OpenInputFile

jmp Word\_Meaning

Word\_L:

cmp al,'L'

jne Word\_M

mov edx,OFFSET L

call OpenInputFile

jmp Word\_Meaning

Word\_M:

cmp al,'M'

jne Word\_N

mov edx,OFFSET M

call OpenInputFile

jmp Word\_Meaning

Word\_N:

cmp al,'N'

jne Word\_O

mov edx,OFFSET N

call OpenInputFile

jmp Word\_Meaning

Word\_O:

cmp al,'O'

jne Word\_P

mov edx,OFFSET O

call OpenInputFile

jmp Word\_Meaning

Word\_P:

cmp al,'P'

jne Word\_Q

mov edx,OFFSET P

call OpenInputFile

jmp Word\_Meaning

Word\_Q:

cmp al,'Q'

jne Word\_R

mov edx,OFFSET Q

call OpenInputFile

jmp Word\_Meaning

Word\_R:

cmp al,'R'

jne Word\_S

mov edx,OFFSET R

call OpenInputFile

jmp Word\_Meaning

Word\_S:

cmp al,'S'

jne Word\_T

mov edx,OFFSET S

call OpenInputFile

jmp Word\_Meaning

Word\_T:

cmp al,'T'

jne Word\_UI

mov edx,OFFSET T

call OpenInputFile

jmp Word\_Meaning

Word\_UI:

cmp al,'U'

jne Word\_V

mov edx,OFFSET U

call OpenInputFile

jmp Word\_Meaning

Word\_V:

cmp al,'V'

jne Word\_W

mov edx,OFFSET V

call OpenInputFile

jmp Word\_Meaning

Word\_W:

cmp al,'W'

jne Word\_X

mov edx,OFFSET W

call OpenInputFile

jmp Word\_Meaning

Word\_X:

cmp al,'X'

jne Word\_Y

mov edx,OFFSET X

call OpenInputFile

jmp Word\_Meaning

Word\_Y:

cmp al,'Y'

jne Word\_Z

mov edx,OFFSET Y

call OpenInputFile

jmp Word\_Meaning

Word\_Z:

cmp al,'Z'

mov edx,OFFSET Z

call OpenInputFile

jmp Word\_Meaning

; pop up mmessgae need to change 3.

Synonyms:: ;FIND SYNONYMS

mov edx,offset string121

call writestring

call crlf

mov edx, OFFSET BUFFER

mov ecx, buffSize

call ReadFromFile

mov edx,0

mov eax,0

mov ebx,0

mov ecx,0

mov esi,0

mov ecx,lengthof input ;Counts the size of input string

lengthofinput: ;For the lenght of the input word

mov al,input[esi]

add esi,1

cmp al,0

jne incrementlength

je endingloop

incrementlength: ;Counter to count the value till the Null terminator

mov ebx,wordsize

add bl,1

mov wordsize,ebx

jmp lengthofinput

endingloop:

mov edx,0

mov eax,0

mov ebx,0

mov ecx,0

mov esi,0

;compares letter by letter in the file

mov ecx,buffSize

l1: ;Comparing all letters from Input and in file

mov al, BUFFER[esi]

mov esi,0

mov esi,counter

cmp al,input[esi]

je l2 ;Comparing the character if true

jne l3 ;If comparing Fail

l2: ;Counting the letters to be

mov ebx,counter

inc bl

mov counter,ebx

mov esi,0

mov esi,position

inc esi

mov position,esi

cmp ebx,wordsize

je printword

jne l1

l3: ;For the next character if the first fail

mov ebx,0

mov counter,ebx

mov esi,0

mov esi,position

inc esi

mov position,esi

loop l1

cmp al,2

je ANT

call ANTONYMS

mov ecx,10000

ANT:

add esi,10

printword: ;Printing the word till the terminating condition

mov al,BUFFER[esi]

;call writechar

dec esi

cmp al,'.' ;Terminating Condition

jne printword

mov ecx,0

add esi,3

Print:

;Printing the word till the terminating condition

mov al,BUFFER[esi]

call writechar

inc esi

cmp al,'.' ;Terminating Condition

jne Print

je l5

l5: ;Closing the file

mov eax, filehandle

call CloseFile

call crlf

call crlf

mov edx,0

mov eax,0

mov ebx,0

mov ecx,0

mov esi,0

jmp l12 ;again call

ret

Text\_Search ENDP

;

;==================================================;

Loading PROC

mov edx,0

mov dh,25

mov dl,72

call gotoxy

mov edx,OFFSET string1

Call WriteString

mov dh,24

mov dl,70

call gotoxy

mov edx,OFFSET string3

Call WriteString

mov dh,24

mov dl,88

call gotoxy

mov edx,OFFSET string3

Call WriteString

mov dh,23

mov dl,72

call gotoxy

mov edx,OFFSET string1

Call WriteString

mov dh,26

mov dl,74

call gotoxy

mov edx,OFFSET string4

Call WriteString

mov ecx,LENGTHOF string2

mov esi,0

mov dh,24

mov dl,72

l1:

movzx eax,string2[esi]

Call gotoxy

Call WriteChar

mov eax,100

Call Delay

inc esi

inc dl

loop l1

ret

Loading ENDP

My\_Instruction PROC

mov edx,0

mov dh,19

mov dl,65

Call Gotoxy

mov eax,10

Call Delay

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

mov edx,0

mov dh,20

mov dl,65

Call Gotoxy

mov edx,OFFSET string11

Call WriteString

mov edx,0

mov dh,22

mov dl,65

Call Gotoxy

mWrite <"[1]. Enter Appropraite Name Of The File",0dh,0ah>

mov edx,0

mov dh,24

mov dl,65

Call Gotoxy

mWrite <"[2]. Your File Name Must Have Extension(.txt)",0dh,0ah>

mov edx,0

mov dh,26

mov dl,65

Call Gotoxy

mWrite <"[3]. Enter Word You Want To Search.",0dh,0ah>

mov edx,0

mov dh,28

mov dl,65

Call Gotoxy

mWrite <"[4]. Press (-1) For Main Menu.",0dh,0ah>

Call Crlf

mov edx,0

mov dh,30

mov dl,65

Call Gotoxy

mov edx,OFFSET string11

Call WriteString

Call Crlf

Call Crlf

mov edx,0

mov dh,32

mov dl,65

Call Gotoxy

ret

My\_Instruction ENDP

ANTONYMS PROC ;------------------------;FIND ANTOPNYMS

mov edx,offset string111

call writestring

call crlf

mov edx, OFFSET BUFFER

mov ecx, buffSize

call ReadFromFile

mov edx,0

mov eax,0

mov ebx,0

mov ecx,0

mov esi,0

mov ecx,lengthof input ;Counts the size of input string

lengthofinput: ;For the lenght of the input word

mov al,input[esi]

add esi,1

cmp al,0

jne incrementlength

je endingloop

incrementlength: ;Counter to count the value till the Null terminator

mov ebx,wordsize

add bl,1

mov wordsize,ebx

jmp lengthofinput

endingloop:

mov edx,0

mov eax,0

mov ebx,0

mov ecx,0

mov esi,0

;compares letter by letter in the file

mov ecx,buffSize

l1: ;Comparing all letters from Input and in file

mov al, BUFFER[esi]

mov esi,0

mov esi,counter

cmp al,input[esi]

je l2 ;Comparing the character if true

jne l3 ;If comparing Fail

l2: ;Counting the letters to be

mov ebx,counter

inc bl

mov counter,ebx

mov esi,0

mov esi,position

inc esi

mov position,esi

cmp ebx,wordsize

je printword

jne l1

l3: ;For the next character if the first fail

mov ebx,0

mov counter,ebx

mov esi,0

mov esi,position

inc esi

mov position,esi

loop l1

cmp al,2

je ANT

call ANTONYMS

ANT:

printword: ;Printing the word till the terminating condition

mov al,BUFFER[esi]

;call writechar

dec esi

cmp al,'.' ;Terminating Condition

jne printword

je print

Print:

sub esi,10 ;Printing the word till the terminating condition

mov al,BUFFER[esi]

call writechar

inc esi

cmp al,'.' ;Terminating Condition

jne Print

je l5

l5: ;Closing the file

mov eax, filehandle

call CloseFile

call crlf

call crlf

mov edx,0

mov eax,0

mov ebx,0

mov ecx,0

mov esi,0

jmp l12 ;again call

ANTONYMS EnDP

Word\_Meaning PROC

cmp GOO ,'a'

je ANTONYMS

Syn:

JMP Synonyms

Word\_Meaning endp

end main