include irvine32.inc

include macros.inc

bsize = 6

buSize = 500

.data

samp dd 1, 2, 3, 4, 5, 6, 7, 8, 9

arr1 dd 0, 0, 4, 6, 0, 8, 9, 1, 2

arr2 dd 0, 7, 2, 0, 9, 5, 3, 4, 8

arr3 dd 1, 0, 0, 3, 4, 2, 5, 0, 7

arr4 dd 0, 5, 9, 7, 0, 1, 4, 2, 0

arr5 dd 9, 2, 6, 0, 5, 0, 7, 9, 0

arr6 dd 0, 1, 3, 9, 0, 4, 8, 5, 0

arr7 dd 9, 0, 1, 5, 3, 7, 0, 0, 4

arr8 dd 2, 8, 7, 4, 0, 0, 6, 3, 0

arr9 dd 3, 4, 5, 2, 0, 6, 1, 0, 9

arr11 dd 0, 0, 4, 6, 0, 8, 9, 1, 2

arr22 dd 0, 7, 2, 0, 9, 5, 3, 4, 8

arr33 dd 1, 0, 0, 3, 4, 2, 5, 0, 7

arr44 dd 0, 5, 9, 7, 0, 1, 4, 2, 0

arr55 dd 9, 2, 6, 0, 5, 0, 7, 9, 0

arr66 dd 0, 1, 3, 9, 0, 4, 8, 5, 0

arr77 dd 9, 0, 1, 5, 3, 7, 0, 0, 4

arr88 dd 2, 8, 7, 4, 0, 0, 6, 3, 0

arr99 dd 3, 4, 5, 2, 0, 6, 1, 0, 9

row dd 0

column dd ?

index dd ?

value dd ?

a dd ?

instruction byte "Instuctions.txt"

instruct\_handler dd ?

instruction\_buffer byte buSize Dup(?)

Score dd 5

HighestScore dd 15

v2 dd ?

ScoreFile byte "Score.txt",0

Scoredh dd ?

score\_buffer dd bsize dup(?)

.code

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

LoadHighScore proc

mWrite " Highest Score "

call crlf

mov edx , offset ScoreFile

call CreateOutputFile

mov scoredh , eax

mov edx , offset score\_buffer

mov ecx , bsize

call ReadString

mov eax , scoredh

mov edx , offset score\_buffer

mov ecx , bsize

call WriteToFile

mov eax , scoredh

call closeFile

mov edx , offset ScoreFile

call OpenInputFile

mov scoredh , eax

call crlf

mov edx , offset score\_buffer

mov ecx , bsize

mov eax , scoredh

call ReadFromFile

mov edx , offset score\_buffer

call WriteString

mov eax , scoredh

call CloseFile

call crlf

ret

LoadHighScore endp

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

ShowScore proc

mov ebx, HighestScore

.if score > eax

mov HighestScore, eax

mwrite " Great! You have set a new Record! "

mwrite" Your Score: "

call writedec

.else

mwrite " Your Score: "

call writedec

mwrite " Highest Score: "

mov eax, ebx

call writedec

.endif

ret

ShowScore endp

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

Attempts proc

mov esi, score

dec esi

mov score, esi

.if esi < 1

call clrscr

mov eax, RED

call setTextColor

call crlf

call crlf

mwrite " --------------- No more attempts. You failed! ----------------"

mov eax, white

call setTextColor

call main

.endif

ret

Attempts endp

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

WrongIndex proc

call clrscr

call Print

mov eax, Red

call setTextColor

mwrite "This index is not available. Try Again"

mov eax, white

call setTextColor

ret

WrongIndex endp

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

WrongValue proc

call clrscr

call Print

mov eax, RED

call setTextColor

mwrite"This value already exists. "

mov eax, White

call setTextColor

Call Attempts

ret

WrongValue endp

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

TakeValue proc

VAL:

mov eax, White

call setTextColor

mwrite "Enter Value: "

call readint

call crlf

mov value, eax

mov eax, RED

call setTextColor

.if value > 9 || value < 0 || value == 0

call crlf

mwrite"Enter number from 1 to 9"

call crlf

jmp VAL

.endif

ret

TakeValue endp

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

SetRowColumn proc

mov edx, 0

mov eax, index

mov ebx, 10

div ebx

mov ebx, eax

mov row, ebx

mov column,edx

.if row > 9 || column > 9 || index <=10 || index == 0

call IncorrectIndex

.endif

ret

SetRowColumn endp

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

printRow proc array\_sample : dword , temp: dword

mov ecx, 1

mov edx, array\_sample

mov eax, Blue

call setTextColor

mwrite " \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ "

call crlf

call crlf

mov eax, green

call setTextColor

mwrite" "

mov eax, temp

call writedec

mov eax, Blue

call setTextColor

mwrite " | "

.while ecx < 10

mov eax, [edx]

mov ebx, eax

.if eax > 0

mov eax, Red

call setTextColor

mov eax, ebx

call writedec

.else

mov eax, Yellow

call setTextColor

mov eax, ebx

call writedec

.endif

mov eax, Blue

call setTextColor

mwrite " | "

add edx, type samp

inc ecx

.endw

call crlf

ret

PrintRow endp

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

Print proc

mov eax, green

call setTextColor

mwrite " 1 2 3 4 5 6 7 8 9 "

call crlf

mov edx, offset arr1

Invoke PrintRow, edx, 1

mov edx, offset arr2

Invoke PrintRow, edx, 2

mov edx, offset arr3

Invoke PrintRow, edx, 3

mov edx, offset arr4

Invoke PrintRow, edx, 4

mov edx, offset arr5

Invoke PrintRow, edx, 5

mov edx, offset arr6

Invoke PrintRow, edx, 6

mov edx, offset arr7

Invoke PrintRow, edx, 7

mov edx, offset arr8

Invoke PrintRow, edx, 8

mov edx, offset arr9

Invoke PrintRow, edx, 9

mwrite " \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ \_\_\_ "

call crlf

call crlf

mov eax, White

call setTextColor

ret

Print endp

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

SetRow proc

.if row == 1

mov edx, offset arr1

.elseif row == 2

mov edx, offset arr2

.elseif row == 3

mov edx, offset arr3

.elseif row == 4

mov edx, offset arr4

.elseif row == 5

mov edx, offset arr5

.elseif row == 6

mov edx, offset arr6

.elseif row == 7

mov edx, offset arr7

.elseif row == 8

mov edx, offset arr8

.else

mov edx, offset arr9

.endif

mov edi, edx

ret

SetRow endp

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

Traverse proc

mov edx, esi

mov ecx, Column

.while ecx > 1

add edx, type samp

dec ecx

.endw

mov eax, value

.if [edx] == eax

mov ebx, 1

.endif

ret

Traverse endp

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

ColumnComparison proc

mov ebx, 0 ;flag

mov esi, offset arr1

call Traverse

mov esi, offset arr2

call Traverse

mov esi, offset arr3

call Traverse

mov esi, offset arr4

call Traverse

mov esi, offset arr5

call Traverse

mov esi, offset arr6

call Traverse

mov esi, offset arr7

call Traverse

mov esi, offset arr9

call Traverse

.if ebx == 1

call WrongValue

call crlf

;call TakeValue

call Comparison

.endif

ret

ColumnComparison endp

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

Comparison proc

L1:

call SetRow

mov edx, edi

call TakeValue

mov eax, value

;call writedec

mov ecx, lengthof samp

mov ebx, 0 ; flag

.while ecx > 0

.if [edx] == eax

call WrongValue

call crlf

jmp L1

mov ebx, 1

.endif

add edx, type samp

dec ecx

.endw

call ColumnComparison

call crlf

.if ebx ==1

jmp L1

.elseif ebx == 0

mov edx, edi

mov ecx, column

.while ecx > 1

add edx, type samp

dec ecx

.endw

mov ebx, value

mov [edx], ebx

call Congrats

.endif

call Print

ret

Comparison endp

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

TakeIndex proc

TryAgain:

mwrite "Enter Index to insert value: "

call readInt

mov index, eax

call SetRowColumn

call SetRow

mov edx, edi

mov ecx, Column

.while ecx > 1

add edx, type samp

dec ecx

.endw

mov ebx, [edx]

.if ebx != 0

call WrongIndex

call crlf

jmp TryAgain

.endif

mov index, eax

ret

TakeIndex endp

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

IncorrectIndex proc

mov eax,RED

call setTextColor

call crlf

mwrite"You have entered wrong index"

call crlf

mov eax,White

call setTextColor

call crlf

call TakeIndex

ret

IncorrectIndex endp

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

PrintTraverse proc

ret

PrintTraverse endp

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

WrongChoice proc

mov eax, RED

call setTextColor

mwrite " Wrong Choice. "

mov eax, white

call setTextColor

call crlf

ret

WrongChoice endp

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

Reset proc

mov esi,offset arr11

mov edx,offset arr1

mov ecx,9

.while ecx >= 1

mov ebx,[esi]

mov [edx],ebx

add edx,type samp

add esi,type arr11

dec ecx

.endw

mov esi,offset arr22

mov edx,offset arr2

mov ecx,9

.while ecx >= 1

mov ebx,[esi]

mov [edx],ebx

add edx,type samp

add esi,type arr22

dec ecx

.endw

mov esi,offset arr33

mov edx,offset arr3

mov ecx,9

.while ecx >= 1

mov ebx,[esi]

mov [edx],ebx

add edx,type samp

add esi,type arr3

dec ecx

.endw

mov esi,offset arr44

mov edx,offset arr4

mov ecx,9

.while ecx >= 1

mov ebx,[esi]

mov [edx],ebx

add edx,type samp

add esi,type arr44

dec ecx

.endw

mov esi,offset arr55

mov edx,offset arr5

mov ecx,9

.while ecx >= 1

mov ebx,[esi]

mov [edx],ebx

add edx,type samp

add esi,type arr55

dec ecx

.endw

mov esi,offset arr66

mov edx,offset arr6

mov ecx,9

.while ecx >= 1

mov ebx,[esi]

mov [edx],ebx

add edx,type samp

add esi,type arr66

dec ecx

.endw

mov esi,offset arr77

mov edx,offset arr7

mov ecx,9

.while ecx >= 1

mov ebx,[esi]

mov [edx],ebx

add edx,type samp

add esi,type arr77

dec ecx

.endw

mov esi,offset arr88

mov edx,offset arr8

mov ecx,9

.while ecx >= 1

mov ebx,[esi]

mov [edx],ebx

add edx,type samp

add esi,type arr88

dec ecx

.endw

mov esi,offset arr99

mov edx,offset arr9

mov ecx,9

.while ecx >= 1

mov ebx,[esi]

mov [edx],ebx

add edx,type samp

add esi,type arr99

dec ecx

.endw

ret

Reset endp

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

Congratulations proc

call clrscr

mov eax,red

call settextcolor

call crlf

call crlf

call crlf

call crlf

call crlf

call crlf

call crlf

call crlf

call crlf

call crlf

call crlf

mwrite" CCCCCCCC OOOOOOO N N GGGGGGGG RRRRRRR A TTTTTTT SSSSSS"

call crlf

mwrite" C O O N N N G R R A A T S "

call crlf

MWRITE" C O O N N N G RRRRRRR AAAAA T S "

call crlf

mwrite" C O O N N N G GGGG R A A T SSSSSSS"

call crlf

mwrite" C O O N N N G G R R A A T S"

call crlf

mwrite" CCCCCCCC OOOOOOO N N GGGGGGGG R R A A T SSSSSSS"

call crlf

mov eax,Yellow

call setTextColor

mwrite" YOU HAVE DONE IT!"

CALL CRLF

call showScore

call readint

call clrscr

mov eax,White

call setTextColor

ret

Congratulations endp

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

Congrats proc

mov ebx, 0

mov edx,offset arr1

mov ecx,9

.while ecx>=1

mov eax,[edx]

.if eax == 0

mov ebx,1

.endif

add edx,type samp

dec ecx

.endw

.if ebx==0

jmp cong2

;call Congratulations

.endif

cong2:

mov ebx, 0

mov edx,offset arr2

mov ecx,9

.while ecx>=1

mov eax,[edx]

.if eax == 0

mov ebx,1

.endif

add edx,type samp

dec ecx

.endw

.if ebx==0

jmp cong3

;call Congratulations

.endif

cong3:

mov ebx, 0

mov edx,offset arr3

mov ecx,9

.while ecx>=1

mov eax,[edx]

.if eax == 0

mov ebx,1

.endif

add edx,type samp

dec ecx

.endw

.if ebx==0

jmp cong4

;call Congratulations

.endif

cong4:

mov ebx, 0

mov edx,offset arr4

mov ecx,9

.while ecx>=1

mov eax,[edx]

.if eax == 0

mov ebx,1

.endif

add edx,type samp

dec ecx

.endw

.if ebx==0

jmp cong5

;call Congratulations

.endif

cong5:

mov ebx, 0

mov edx,offset arr5

mov ecx,9

.while ecx>=1

mov eax,[edx]

.if eax == 0

mov ebx,1

.endif

add edx,type samp

dec ecx

.endw

.if ebx==0

jmp cong6

;call Congratulations

.endif

cong6:

mov ebx, 0

mov edx,offset arr6

mov ecx,9

.while ecx>=1

mov eax,[edx]

.if eax == 0

mov ebx,1

.endif

add edx,type samp

dec ecx

.endw

.if ebx==0

jmp cong7

;call Congratulations

.endif

cong7:

mov ebx, 0

mov edx,offset arr7

mov ecx,9

.while ecx>=1

mov eax,[edx]

.if eax == 0

mov ebx,1

.endif

add edx,type samp

dec ecx

.endw

.if ebx==0

jmp cong8

;call Congratulations

.endif

cong8:

mov ebx, 0

mov edx,offset arr8

mov ecx,9

.while ecx>=1

mov eax,[edx]

.if eax == 0

mov ebx,1

.endif

add edx,type samp

dec ecx

.endw

.if ebx==0

jmp cong9

;call Congratulations

.endif

cong9:

mov ebx, 0

mov edx,offset arr9

mov ecx,9

.while ecx>=1

mov eax,[edx]

.if eax == 0

mov ebx,1

.endif

add edx,type samp

dec ecx

.endw

.if ebx==0

call Congratulations

.endif

ret

congrats endp

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

Welcome proc

mov eax,yellow

call settextcolor

call crlf

call crlf

call crlf

call crlf

call crlf

call crlf

call crlf

call crlf

call crlf

call crlf

call crlf

mwrite " SSSSSSS UU UU DDD OOOOOOOO KK KK UU UU |||"

call crlf

mwrite " SSSSSSS UU UU DDDDD OOOOOOOO KK KK UU UU ||| "

call crlf

mwrite " SS S UU UU DD DD OO OO KK KK UU UU |||"

call crlf

mwrite " SS UU UU DD DD OO OO KK KK UU UU |||"

call crlf

mwrite " SS UU UU DD DD OO OO KK KK UU UU |||"

call crlf

mwrite " SS UU UU DD DD OO OO KK KK UU UU |"

call crlf

mwrite " S SS UU UU DD DD OO OO KK KK UU UU "

call crlf

mwrite " SSSSSSS UUUUUUUUUU DDDDDD OOOOOOOO KK KK UUUUUUUUUU @"

call crlf

call crlf

call crlf

mov eax, Blue (Yellow)

call setTextColor

mwrite " Do you want to continue ? "

call readint

call clrscr

call crlf

mov eax,WHITE

call settextcolor

ret

Welcome endp

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

Thankyou PROC

call clrscr

mov eax,red

call settextcolor

call crlf

call crlf

call crlf

call crlf

call crlf

call crlf

call crlf

call crlf

call crlf

call crlf

call crlf

mwrite " TTTTTTTT H H A N N K K U U "

call crlf

mwrite " T H H A A N N N K K U U"

call crlf

mwrite " T H H A A N N N K K U U "

call crlf

mwrite " T HHHHHHHH AAAAAAA N N N K K U U"

call crlf

mwrite " T H H A A N N N K K U U "

call crlf

mwrite " T H H A A N N K K UUUUUUUUU"

call crlf

call readint

call clrscr

mov eax,White

call setTextColor

ret

Thankyou endp

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

Instructions proc

mWrite " I N S T R U C T I O N S "

call crlf

;mov edx , offset instruction

;call CreateOutputFile

;mov Instruct\_Handler , eax

;mov edx , offset instruction\_buffer

;mov ecx , buSize

;call ReadString

;mov eax , Instruct\_Handler

;mov edx , offset instruction\_buffer

;mov ecx , buSize

;call WriteToFile

;mov eax , Instruct\_Handler

;call closeFile

mov edx , offset instruction

call OpenInputFile

mov Instruct\_Handler , eax

call crlf

mov edx , offset instruction\_buffer

mov ecx , buSize

mov eax , Instruct\_Handler

call ReadFromFile

mov edx , offset instruction\_buffer

call WriteString

mov eax , Instruct\_Handler

call CloseFile

call crlf

ret

Instructions endp

;------------------- M A I N -----------------

main proc

call Welcome

call LoadHighScore

START:

;--------------------------Staring menu

call Instructions

call readint

call clrscr

mwrite " 0. Exit 1.Play"

call crlf

mwrite " Enter Your Choice: "

call readint

.if eax == 0

jmp ENDING

.elseif eax ==1

call Print

jmp Play

.else

call clrscr

call wrongChoice

jmp START

.endif

PLAY:

mov ecx, 1

.while ecx > 0

mwrite " 0. Exit 1. Enter 2.Reset 3.Read Instructions"

call crlf

mwrite " Enter Your Choice: "

call readint

.if eax == 0

JMP ENDING

.elseif eax == 1

call TakeIndex

call Comparison

call clrscr

call print

JMP PLAY

.elseif eax == 2

call Reset

call clrscr

call print

JMP PLAY

.elseif eax == 3

call instructions

.else

call clrscr

call Print

call wrongChoice

JMP PLAY

.endif

.endw

ENDING:

call Thankyou

; Exit the program

invoke ExitProcess, 0

exit

main endp

end main