ID No 14306

Semester 5th

Subject Assembley language

Assignment No 5

Submitted to Sir Amin

Page ① Assignment NO 5

Q1 Which register (in 32-bit mode) manages the Stack?

Ans Extended Stack Pointer.

Q2 Why is the Stack called a LIFO Structure?

Ans It is called LIFO Structure because it contains set of memory blocks, in which data is retrieved in order i.e. The last value pushed into the Stack is the first value popped out from the Stack.

Q3 When a 32-bit value is pushed on the Stack, what happens to ESP?

Ans ESP is Decremented by 4.

Q4 What would happen if the RET instruction was amitted from

a Procedure?

Ans. Execution would continue beyond the End of the Procedure, Possibly into the beginning of another Procedure.

Q5 How are the word Recive and Returns used in the Suggest Procedure documentation?

Ans A list of input Parameter and their usage, labeled by a word Such as Receives. A description of any values returned by the Procedure, labeled by a word Such as Returns.

Q6 which Procedure in the link library generates a random integer within a Selected range?

Ans Random Range Procedure.

Q7 which procedure in the Library display "Press [Enter] to Continue" and waits for the user to Press the Enter Key?

Ans Wait Msg Procedure.

Q8 write Statement that Cause a Program to Pause for 700 milli Seconds.

Ans Code example mov eax, 700
Call Delay.

Q9 which procedure from the link library writes an unsigned integar to the Console window in decimal format?

Ans Write Dec procedure.

Q10 which procedure from the link library Place the cursor at a Specific Console window.

location?

Ans GotoXy.

Q11 What are the required input parameters for the DumpMem procedure?

Ans Imagine two possible way of calling the DumpMem.

Require input parameters.

- Pushed
- mov esi, OFFSET array
- mov ecx, LENGTHOF array
- mov ebx, TYPE array
- Call DumpMem
- Popad.

OR

- Push OFFSET array

- Push LENGTHOF array
- Push TYPE array
- Call DumpMem.

Q12 what are the required input Parameter for the readString Produre?

Ans EDX Contain the offset of an array of bytes, and ECX Contain the maximum number of Characters to read.

Q13 which Procedure in the link library generates a random integer within a Selected range?

Ans Random Range.

Q14 what will be the final value in EAX after these instructions execute?

```
Push 5
Push 6
Pop eax
Pop eax
```

Ans Final value in EAX after these instruction.
. Exection (5)

Q.15 which Statement is true about what will happen when the example code run?

```
1. main PROC
2. Push 10
3: Push 20
4: Call EX2Sub
5: Pop eax
6: INVOKE ExitProcess,0
7. main ENDP
8:
9: EX2Sub PROC
```

```
10: Pop eax
11: ret
12: Ex2Sub ENDP
```

Ans This one I would belive it would equal 10 because of Lifo (last in first out). EAX would equal 10 because it was the last one in.

Q16 which Statement is true about what will happen when the example code runs?

```
1: main PROC
2: mov eax, 30
3: Push eax
4: Push 40
5: Call Ex3Sub
6: INVOKE ExitProcess, 0
7: main ENDP
8:
9: ex3Sub PROC
```

10. Pusha
11. mov eax, 80
12. Popa
13. Ret
14. EX3Sub ENDP

Ans BY: looking at this I think that EAX would still equal 30 at the end of line 6 because eax was just pushed on the Stack not a change in value.

Q17 which Statement is true: about what will happen when the example code runs?

1: main PROC
2: mov eax, 40
3: Push offset Here
4: jump Ex4Sub
5: Here:
6: Mov eax, 30

```
7: INVOKE ExitProcess,0
8: main ENDP
9:
10. EX4Sub PROC
11: ret
12: EX4Sub ENDP
```

Ans. C. EAX will be equal 30 on line 6.

Q18. which Statment is true about what will happen when the example code runs?

```
1: main PROC
2: mov edx,0
3: mov eax,40
4: Push eax
5: Call EX5Sub
6: INVOKE ExitProcess,0
7: main ENDP
8:
9: EXSub PROC
```

10. Pop eax
11. Pop edx
12. Push eax
13. Get
14. Ex5 Sub eax

Ans A. EDX will be equal 40 on line 6.

Q19 write a Sequence of Statements that use only PUSH and POP instructions to exchange the value in the EAX and EBX registers.

Ans
```
Push ebx
Push eax
Pop ebx
Pop eax
```

Q21 Create a Procedure that generate a random String of length L, containing all Capital letters. when Calling the Procedure, Pass the value of L in EAX, and Pass a pointer to an array of byte that will hold the random String. write a test program that Calls your Procedure 20 times and displays the Strings in the Console window.

Program:-

```
; Random Strings.

INCLUDE Irvine32.inc
TAB = 9                    ; ASCII Code for Tab
Strlen = 10                ; length of the String
.386
.model flat, stdcall
.stack 4096
```

```
ExitProcess PROTO, dwExitCode:
DWORD

.code
main PROC
  mov edx, offSET Str1
  ;"The C20 random String are:"
  Call WriteString
  ; writes String.
  Call CrLF
  ; writes on end-of-line
  Sequence to the console window.
  mov ecx, 20      ; Create 20 String

L1: mov edx, OFFSET addr1
  mov eax, Strlen
  ; EAX : String : length
  Call RandomString
  ; EAX : String length
  Call Display
  ; generates the random String
```

```
mov al, TAB
Call writeChar
exit
main ENDP ; leave a tab space

Randomstring PROC USES eax
esi mov ecx, eax ; ECX = String
length

L1:
mov eax, 26
Call RandomRange
add eax, 65 ; EAX gets ASCII
of a Captile letter.
mov axx1 [esi], eax
mov
inc esi
loop L1
Randomstring EXDP

Display proc USES eax esi
```

```
                    ; Display the
                    Generate random strings

mov ecx, eax        ; ECX = String length
L1:
mov eax, ascii[esi] ; EAX = ASCII value
Call write Char     ; write the letter
inc esi
loop L1
Display ENDP
Call dumpregs

INVOKE Exit : Process, 0
END main.
```

Q22 write a program that display a Single Character at 100 random Screen location, using a timing delay of 100 millseconds, Hint: use the GetMax XY produce to determine the Current Size of the Consolve window.

Ans Title Random Characters (Source.cpp).

```
// This Program display a single
Character at 100 random Screen
locations.
Include Irvine 32.Inc
.data
row WORD? ; // rows variable
                to hold num of rows.

COLS WORD? ; // Cols variable to
                hold num of Columns.
.Code
main PROC
Call ClrScr ; // Set Cursor top left
mov ecx, 100 ; // loop for 100 times.

L1: Cal GetMaxXY; // Size Console window
    mov rows, ax ; // return rows
    mov Cols, dx ; // return Columns
```

```
movZx eax, rows ; // moving rows eax
Call RandowRange ;// generate integer
mov dh,al        ;// Setting range boundaries

movZx eax,cols ;// moving Columns to eax
Call RandomRange;// generate integers
mov dl ,al      ;// Setting range boundaries

Call Gotoxy ;// Cursor alocation
Call writechar;// write random Charater
mov eax,100 ;// time 100 milliseconds
Call delay ;// Pauses Program,100 millisecond

loop L1     ; looping
exit
main ENDP
END main.
```

Q23 write a Program that display a Single Character in all possible Combinations of foreground and back ground colours (16-16-256).. The Colors are numbered from 0 to 15, So you Can use a nested loop to generate all possible Combinations.

Ans

```
; Color Matrix
; write a Program that displays
a Single Characters. in all possible
; Combination of foreground and
background Color (16x16=256). The
; Color are numbered from 0 to 15,
So you Can use a nested
loop to
; generate all possible Combination.

; last. update:
```

```
INCLUDE Irvine32.inc

CR = 0Dh ; Carriage return
LF = 0Ah ; line feed
.data
prompt1 BYTE "Please type a
Character: ",0
dispChr DWORD?
.Code
 main PROC
; Set text color to white text
on black background.
; even though these are the default
Colours.
; Each colour constant is defined
in Irvine32-inc
    mov eax, white + (black * 16)
    Call SetTextColor
    Call Clrscr ; Clear the Screen
; Get the user to type Some
 Character for our display:
```

```
mov edx, OFFSET Prompt1: Please type a...
Call writeString
Call ReadChar

; One now has the desired characters in AL,
; but we can only used it letter so
; we save it in variable dispChar:

mov dispChar, eax
Call CrLF      ; new line

; Generating all the possible color
Combination.
; in a nested loop require a
bit more work
; than simply a single loop.
; ECX will be preserved on the
Stack across the inner loop:

mov ecx; outer loop counter.
```

```
L1: Push ecx ; Save outer loop
                Counter.

; As the text attributes above
illustrate
; to set Colors we need to
Compute.
; EAX in a way that Combine both
attributes.
; Set EAX using the Stack Copy
of EAX.
; Value so the background will
Vary
; Slowest (with the outer loop Counter)

    mov ecx,16 ; inner loop Counter

L2: Pop eax  ; get the outer
               loop Counter in
               EAX
```

```
Push eax      ; Saves it back again
Sub eax,1     ; Shift range from 1-16
                to 0-15

rol eax,4     ; Sets AL for the background
                Color
rol eax,ecx   ; add the inner loop
                counter of ECX
Sub eax,1     ; Set AL for the
                foreground Color
Call SetTextColor

; EAX is now available for
restoring the display character.

mov eax,dispch:
Call writeChar

Loop L2

; Return the cursor to begining
```

of matrix

```
Call CrlF       ; new line
Pop ecx         ; restore outer loop
                  Counter
loop L1

; Show the result on Screen
until user hit enter, then exit

mov eax, white + (black * 16)
Call SetTextColor ; otherwise we
                    leave black on
                    black

Call CrlF       ; new line
Call waitMsg    ; "Press [Enter]

exit
main ENDP
END main
```