IQRA NATIONAL UNIVERSITY

INU

DEFINING POSSIBILITIES, CREATING OPPORTUNITIES

Department of Computer Science

Microprocessor & Assembly Language

Assignment Number: 05

Name _______ Mohammad Basir

ID _______ 13142

INSTRUCTOR Name _______ Mohammad Amin

Semester _______ 8th

Q_1

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

ANSWER :

ESP .

Q_2

Why is the stack called a LIFO structure?

ANSWER :

LIFO is short for “Last In First Out”. The last element pushed onto the **stack** will be the first element that gets popped off. ... If you were to pop all of the elements from the **stack** one at a time then they would appear in reverse order to the order that they were pushed on.

Q_3

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

ANSWER:

The **ESP** register holds a **32-bit** offset into some location on the **stack**. - **ESP** always points to the last **value** to be added to, or **pushed** on, the top of **stack**. - A **stack** is also called a LIFO structure (Last-In, First-Out) because the last **value** put into the **stack** is always the first **value** taken out.

Q_4

What would happen if the RET instruction was omitted from a procedure?

ANSWER:

The program would not return to the point where it was before the procedure was called.

Q_5

How are the words Receives and Returns used in the suggested procedure documentation?

ANSWER:

Receives: state of the registers before the procedure and what they will be used for in the procedure. (Input parameters)
Returns: indicates what value, if any, the procedure produces when it returns it to its caller.

Q_6

Which procedure in the link library generates a random integer within a selected range?

ANSWER:

RandomRange procedure

Q_7

Which procedure in the link library displays “Press [Enter] to continue. . .” and waits for the user to press the Enter key?

ANSWER

WaitMsg procedure .

Q_8

Write statements that cause a program to pause for 700 milliseconds.

ANSWER

```
mov eax,700
call Delay
```

Q_9

Which procedure from the link library writes an unsigned integer to the console window in decimal format?

ANSWER:

WriteDec procedure .

Q_10

Which procedure from the link library places the cursor at a specific console window location?

ANSWER:

Gotoxy procedure .

Q_11

What are the required input parameters for the DumpMem procedure?

ANSWER:

ESI contains the data's starting address, ECX contains the number of data units, and EBX contains the data unit size (byte, word, etc)

Q_12

What are the required input parameters for the ReadString procedure ?

ANSWER:

EDX contains the offset of an array of bytes, and ECX contains the maximum number of characters to read .

Q_13

Which procedure in the link library generates a random integer within a selected range?

ANSWER:

RandomRange **procedure**.

Q_14

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

push 5

push 6

pop eax

pop eax

ANSWER:

eax = 5

Q_15

Which statement is true about what will happen when the example code runs? 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

a. EAX will equal 10 on line 6
b. The program will halt with a runtime error on Line 10
c. EAX will equal 20 on line 6
d. The program will halt with a runtime error on Line 11

ANSWER

“ D “

The program will halt with a runtime error on Line 11

Q_16

Which statement is true about what will happen when the example code runs?

```
main PROC
mov eax,30
push eax
push 40
call Ex3Sub
INVOKE ExitProcess,0
main ENDP
Ex3Sub PROC
pusha
mov eax,80
popa
ret
Ex3Sub ENDP
```

a. EAX will equal 40 on line 6
b. The program will halt with a runtime error on Line 6

c. EAX will equal 30 on line 6

d. The program will halt with a runtime error on Line 13

ANSWER :

“ C ”

. EAX will equal 30 on line 6

Q_17

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: jmp Ex4Sub
 5: Here:
 6: mov eax,30
 7: INVOKE ExitProcess,0
 8: main ENDP
9: 10: Ex4Sub PROC
11: ret
12: Ex4Sub ENDP
```

a. EAX will equal 30 on line 7

b. The program will halt with a runtime error on Line 4

c. EAX will equal 30 on line 6

d. The program will halt with a runtime error on Line 11

ANSWER:

“C”

EAX will equal 30 on line 6 .

Q_18

Which statement 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: Ex5Sub PROC
10: pop eax
11: pop edx
12: push eax
13: ret
14: Ex5Sub ENDP
```

a. EDX will equal 40 on line 6

b. The program will halt with a runtime error on Line 13

c. EDX will equal 0 on line 6

d. The program will halt with a runtime error on Line 11

ANSWER :

“ A”

EDX will equal 40 on line 6 .

Q_19

Write a sequence of statements that use only PUSH and POP instructions to exchange the values in the EAX and EBX registers.

ANSWER:

```
push ebx
 push eax
pop ebx
pop eax
```

Q_20

Create a procedure that generates 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.

ANSWER :

```
INCLUDE Irvine32.inc
strLen=10
.data
arr BYTE strLen DUP(?)
```

```
.code
   main PROC
      call Clrscr
      mov esi, offset arr
      mov ecx, 20
      L1:
         call GenerateRandomString
      Loop L1
      call WaitMsg
      exit
   main ENDP

   GenerateRandomString PROC USES ecx
      mov ecx, lengthOf arr
      L2:
         mov eax, 26
         call RandomRange
         add eax, 65
         mov [esi], eax
         call WriteChar        ; write character
      loop L2
      call Crlf
      ret
   GenerateRandomString ENDP

END main
```

Q_22

Write a program that displays a single character at 100 random screen locations, using a timing delay of 100 milliseconds. Hint: Use the GetMaxXY procedure to determine the current size of the console window.

ANSWER:

```
INCLUDE Irvine32.inc

.data
rows WORD ?
cols WORD ?


.code
    main PROC
        call Clrscr
        mov ecx, 100
        L1:
            call GetMaxXY
            mov rows, ax
            mov cols, dx

            movzx eax, rows
            call RandomRange
            mov dh, al

            movzx eax, cols
            call RandomRange
            mov dl, al

            call Gotoxy         ; locate cursor

            mov al,'H'
            call WriteChar

            mov eax,100
            call Delay

        Loop L1

        call WaitMsg
        exit
    main ENDP
```

```
END main .
```

Q_23

Write a program that displays a single character in all possible combinations of foreground and background colors (16 _ 16 _ 256). The colors are numbered from 0 to 15, so you can use a nested loop to generate all possible combinations.

ANSWER:

```
INCLUDE Irvine32.inc

.data
count DWORD ?

.code
main PROC

    mov eax, 0 + (0 * 16)
    mov ecx, 16
    L1:
        mov count, ecx
        push eax
        mov ecx, 16
        L2:
            call SetTextColor
            push eax
            mov al,'H'
            call WriteChar
            pop eax
            inc eax
        LOOP L2
        call crlf
        pop eax
        add eax, 16
        mov ecx, count
    LOOP L1

    call crlf
    call WaitMsg
    exit
main ENDP

END main
```

Q_24

Write a program that displays a string in all possible combinations of foreground and background colors (16 x 16 =256). The colors are numbered from 0 to 15, so you can use a nested loop to generate all possible combinations.

ANSWER:

```
INCLUDE Irvine32.inc

.data
count DWORD ?

.code
main PROC

    mov eax, 0 + (0 * 16)
    mov ecx, 16
    L1:
        mov count, ecx
        push eax
        mov ecx, 16
        L2:
            call SetTextColor
            push eax
            mov al,'H'
            call WriteChar
            pop eax
            inc eax
        LOOP L2
        call crlf
        pop eax
        add eax, 16
        mov ecx, count
    LOOP L1

    call crlf
    call WaitMsg
    exit
main ENDP

END main
```