



Assignment # 05
Microprocessor & Assembly Language
BS (CS)
Fall semester 2018

- Q.1** Which register (in 32-bit mode) manages the stack?
- Q.2** Why is the stack called a LIFO structure?
- Q.3** When a 32-bit value is pushed on the stack, what happens to ESP?
- Q.4** What would happen if the RET instruction was omitted from a procedure?
- Q.5** How are the words *Receives* and *Returns* used in the suggested procedure documentation?
- Q.6** Which procedure in the link library generates a random integer within a selected range?
- Q.7** Which procedure in the link library displays “Press [Enter] to continue. . .” and waits for the user to press the Enter key?
- Q.8** Write statements that cause a program to pause for 700 milliseconds.
- Q.9** Which procedure from the link library writes an unsigned integer to the console window in decimal format?
- Q.10** Which procedure from the link library places the cursor at a specific console window location?
- Q.11** What are the required input parameters for the DumpMem procedure?
- Q.12** What are the required input parameters for the ReadString procedure?
- Q.13** Which procedure in the link library generates a random integer within a selected range?
- Q.14** What will be the final value in EAX after these instructions execute?

```
push 5
push 6
pop eax
pop eax
```

- 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

Q.16 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

```

- 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

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

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
- Q.19** Write a sequence of statements that use only PUSH and POP instructions to exchange the values in the EAX and EBX registers.
- Q.21** 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.
- 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.
- 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.
- 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.