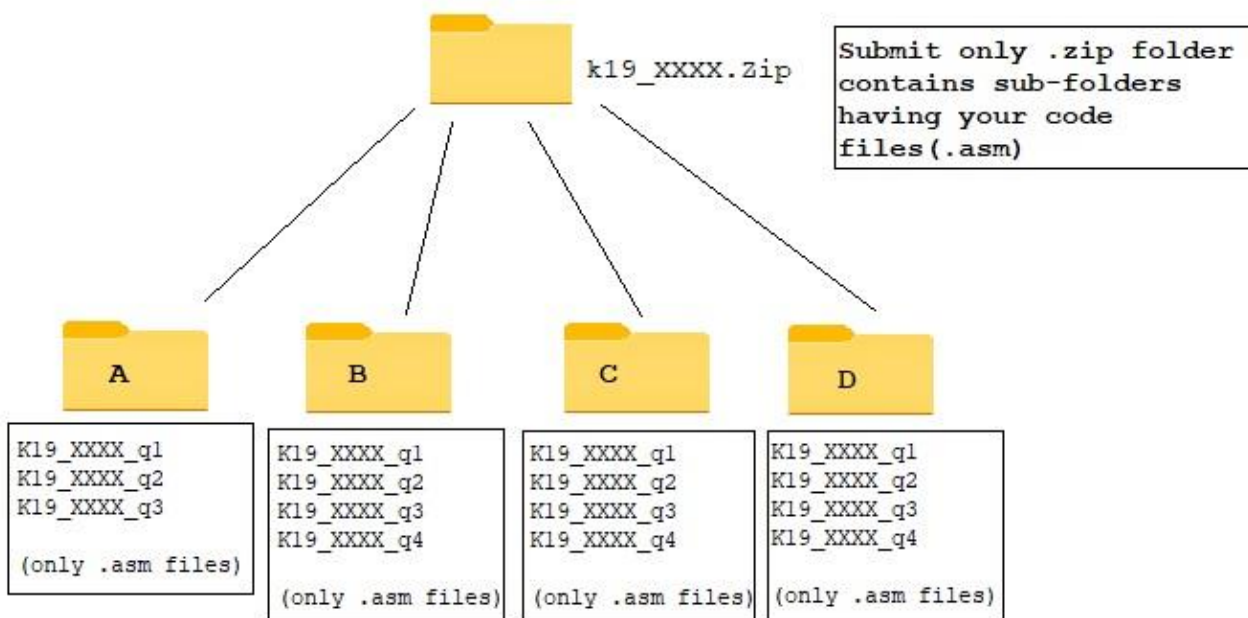


COAL Assignment: 02

Instruction:

Marks: 75 Point (Weightage: 08)

- You just need to submit only a ZipFile, with the name **k19_XXXX**.
- ZipFile contains **FOLDER FOR EACH SECTION** of assignment, that folder contains only solution file(.asm) for each question of particular section.
- Naming Convention for a solution file should be **K19_XXXX_q1**, **K19_XXXX_q2**, **K19_XXXX_q3** and so on.
- Deadline: **Friday December 11, 2020 (before midnight 11:45)** on google classroom.



A. Conditional Processing

1. Using recursion, calculate and display the **6th** positioned to **10th** positioned elements of Fibonacci Series, also draw out the stack (stack frame) for your code.

0,1,1,2,3,**5,8,13,21,34**

2. Translate the following pseudo-code to Assembly Language

```
Swap_Count = 0  
for all elements of list  
    if list[i] > list[i+1]  
        swap(list[i], list[i+1])  
        Swap_Count = Swap_Count + 1  
    end if  
end for  
Print Swap_Count
```

3. Write a program for sequential search. Take an input from the user and find if it occurs in the following array

arr WORD 10, 4, 7, 14, 299, 156, 3, 19, 29, 300, 20

B. Integer Arithmetic:

1. Using shift and add instructions multiply a decimal number X_{10} by 23_{10} . Assume that the result does not exceed the range of a **16-bit** register.
2. The **DATE_OF_BIRTH** field of a file directory entry uses bits 0 to 7 for the **DAY**, bits 8 to 15 for the **MONTH**, and bits 16 to 31 for the **YEAR**. Write instructions to copy the **YEAR** to a word variable **wYEAR**.
3. Given that multi-level encryption is implemented in the following code. Provide encrypted values in **AX/AL** after each encryption and write down a working **decryption key** for each encryption.

```
MOV AL, 04h
MOV CL, AL
; encryption1
CBW
INC AH
XOR AX, 04h                ; AX ?
; encryption2
ROL AH, CL
DEC CL
SHL AX, CL
TEST AX, 0FFFFh           ; AX?
```

4. Write a procedure that should swap the values of **AX** in and **DX** in such a way that whatever is stored in **AX**, after swapping **DX** would hold reverse of it, and vice versa. (For instance, given that **AX** = **ABCDh**, **DX** = **7654h**, after swapping: **AX** = **4567h** and **DX** = **DCBA**). Make use of stack, XCHG and Shift/Rotate instructions only. (5)

C. Procedures:

1. Write a program that reads 6 inputs from the user and prints the lowest and highest input.
2. Recode the logic using `cmp` and conditional jump statements (Do not use `.IF` / `.ENDIF` directives)

```
if(marks==100){
    printf(string1);
}
else if(marks>=60 && marks<100){
    printf(string2);
}
else{
    printf("You Failed!");
}
```

3. Write a simple menu based program to perform basic arithmetic operations. Take 2 operands as user input then perform user desired operation over them. Create a separate procedure to perform each operation.
4. Write a program that reads a single character '0' to '9', 'A' to 'F', or 'a' to 'f', and then converts it and displays it as a number between 0 and 15 (Hexadecimal conversion). Use the `ReadChar` procedure to read the character. Do not use the `ReadHex` procedure. Display an error message if the input character is invalid.

D. Strings and Arrays:

1. Write a program that takes a string of up to 50 characters as input from the user and stores it in an array. It should then convert each lowercase letter to uppercase, leaving every other character unchanged. The program should output the modified string.
2. Store your first and last name in two separate arrays named `string1` and `string2`. Replace the first three characters of `string1` with the last three characters of `string2`.
3. Palindromes are words that can be read both forward and backward. For example, level, civic, racecar and Bob. Write a program that verifies if a string is a palindrome or not. Your program should work for both character and integer strings.
4. Define a 2-dimensional array of integer values with 3 logical rows and 3 logical columns. Write a program that updates the value of each element in the main diagonal of the array to zero, then calculates the sum of each row.