1. Please implement the following steps with PUSH and POP in the .code section: (a) get values 1 and 2 into the stack; (b) save values 2 and 1 in EAX and EBX

```
.code

main proc

            mov eax, 1

            mov ebx, 2


            push eax

            push ebx


            pop eax

            pop ebx

            invoke ExitProcess,0

      main endp

end main
```

2. To (1) get the values 6, 4, and 2 into the stack; and (2) save values 2, 4 and 6 in EAX (values can be overwritten in EAX)

, please fill out blank lines in the .code section. (assume: array WORD 2,4,6)

```
.code

main proc

            mov eax,0

            mov ecx,3


            pushLoop:

                push array[(ecx *2) - 2]

            loop pushLoop


            mov ecx, 3


            popLoop:

                pop eax

            loop popLoop


            invoke ExitProcess,0
```

```
    main endp
end main
```

3. Please predict the values in EDX in step ⓪-④. (assume: arrayVariable DWORD 3h, 6h, 9h)

```
.code
main proc
        mov eax,0
        mov ecx,3
        mov edx, arrayVariable[0]------------------⓪  ; EDX = 0000 0003
        mov edx, arrayVariable[1]------------------①  ; EDX =0600 0000
        mov edx, arrayVariable[2]------------------②  ; EDX =0006 0000
        mov edx, arrayVariable[3]------------------③  ; EDX =0000 0600
        mov edx, arrayVariable[4]------------------④  ; EDX =0000 0006

        pushLoop:
            push arrayVariable[(ecx *4) - 4]
        loop pushLoop

        mov ecx, 3
        popLoop:
              pop eax
        loop popLoop

        invoke ExitProcess,0
    main endp
end main
```

4. Reverse String. Please fill out blank lines with proper instructions.

```
.data
aName BYTE "Assembly Language",0

.code
main PROC
```

```
; Push the name on the stack.
        mov ecx, LENGTHOF aName
        mov esi,0

L1:     movzx eax, aName[esi]
        push eax  ; push on stack
        inc esi
        Loop L1

; Pop the name from the stack, in reverse,
; and store in the aName array.
        mov ecx, LENGTHOF aName
        mov esi, 0

L2:     pop eax
        mov aName[esi],al
        inc esi
        Loop L2

; Display the name.
        ……………..
    main ENDP
END main
```

5. Please use two procedures (pushProc and popProc) to rewrite Q2.

```
.code
main proc
        mov eax,0
        mov ecx,3
; Main program control procedure.
; Calls: pushProc and popProc.
        call pushProc
        mov ecx, 3
        call popProc
```

```
main ENDP


;-----------------------------------------------------
pushProc proc
;
; Push values in array into stack
;-----------------------------------------------------
   pushLoop:
                  push array[(ecx *2) - 2]
               loop pushLoop

               ret
pushProc endp
;-----------------------------------------------------
popProc proc
;
; Pop each value one by one in EAX
;-----------------------------------------------------
               popLoop:
                  pop eax
               loop popLoop

               ret
popProc endp

end main
```