

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

.code

main proc

mov eax, 1 ; save value 1 in EAX

mov ebx, 2 ; save value 2 in EBX

push eax ; get values 1 into the stack

push ebx ; get values 2 into the stack

pop eax ; save value 2 in EAX

pop ebx ; save value 1 in EBX

invoke ExitProcess, 0

main endp

end main

2. To (1) save 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

array WORD 2, 4, 6

main proc

mov eax, 0

mov ecx, 3

pushLoop:

push array[(ecx-1) \* 2] ; locate a proper index of array

loop pushLoop

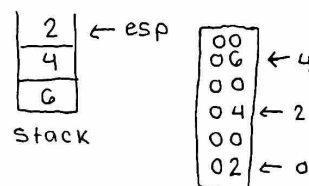
mov ecx, 3 ; configure ECX for popLoop

popLoop:

pop eax ; save values 2, 4 and 6 in EAX

loop popLoop

invoke ExitProcess, 0



```
main endp
```

```
end main
```

3. Please predict the values in EDX in step ①-④. (assume: arrayVariable DWORD 3h, 6h, 9h)

```
.code
```

```
main proc
```

```
mov eax,0
```

```
mov ecx,3
```

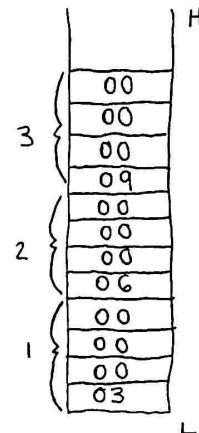
```
mov edx, arrayVariable[0]-----① ; EDX = 0000 0003
```

```
mov edx, arrayVariable[1]-----② ; EDX = 0000 0000
```

```
mov edx, arrayVariable[2]-----③ ; EDX = 0000 0000
```

```
mov edx, arrayVariable[3]-----④ ; EDX = 0000 0000
```

```
mov edx, arrayVariable[4]-----⑤ ; EDX = 0000 0006
```



```
pushLoop:
```

```
push arrayVariable[(ecx-1)*4]; proper index for DWORD
```

```
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 ; ecx = ?/Alternative to get a size
mov esi, 0 ; initialize ESI
```

```
L1:  movzx eax, aName[esi] ; get character
      push eax ; push on stack
      inc esi ; update ESI
      Loop L1
```

; Pop the name from the stack, in reverse,  
; and store in the aName array.

```
mov ecx, lengthof aName; configure ecx for loop L2 again
mov esi, 0 ; configure esi for aName again
```

```
L2:  pop eax ; get character    al
      mov aName[esi], eax ; store in string
      inc esi ; update 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 ; call pushProc procedure
```

```
mov ecx, 3
```

```
call popProc ; call popProc procedure
```

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