COAL Assignment 2

21K-3153

(Q1) Error: Eax should' be initailized to 0 to prevent garbage values being added

The "ktum" variable does not exist

"add [esi],eax" should be "add eax,[esi]" as all addition is reliant on eax and the contents of eax are moved into the result variable

The ret instruction pops OFFSET X1 when it should have popped 11500000h

1FE0h →		
0000 1FE8h →		
0000 1FEC →	OFFSET X1+4	→ popped
00001FF0h →	OFFSET X1	
00001FF4h →	11500000h	
0000FF8h →	5	
	6	

~~MOV result, eax~~	~~ESP = 0000 1FEC~~
MOV result, eax	ESP = 0000 1FEC h
PUSH OFFSET [X1+4]	ESP = 0000 1FE4 h
ADD [esi],eax	~~ESP = 0000 1FE8 h~~
	[esi] = 0027h → if ear initialized to 0

(Q2)

```
Mov al, 00100101 b
Test al, 00001001 b
```

00100101
00001001
00000001

Zero flag=0

```
mov al, 00100100 b
test al, 00001001 b
```

00100100 b
00001001 b
00000000

ZF=1

```
(Q3)      mov ecx, lengthof arr1                arr2  8 db DUP(0)
          mov esi, offset arr1
          mov edi, offset arr2

func        proc uses ecx edi esi

   L1:
        mov eax,0
        mov eax, [esi]
       cmp eax, 0
       JL  not
       mov [edi], eax
       ~~add esi,4~~
       add edi,4
        not:
             add esi,4
      loopL1
      ret
   func endp

   arr2:   40  98  78  0  32  0  0  0
```

(Qn)

```
N SDWORD ?
A sdword ?
B sdword ?

.main
while:
  mov eax,N
  cmp eax,0
  JLE endwhile
  cmp eax,3
  JNE false
  cmp eax,A
  JL trueor
  cmp eax,B
  JLE falseor
   trueor:
   sub n,2
   JMP while
   falseor:
    sub n,1
   JMP while
 endwhile:
  exit
main endp
END main
```

(Q5)

```
.data
        O  byte "o",0
           E  byte "e",0
        invalid byte "invalid",0

.code
Main proc
call readint
cmp al,1
JE L1
cmp al,3
 JNE L2
L1:
   mov edx, offset O
   call writestring
   exit
L2:
   ~~Mov edx, off~~   cmp al,2
                      JEL3
                      cmp al,4
                      JNE L4
L3:
      mov edx, offset E
      call write string
      exit
```

```
L4:
   mov edx, offset invalid
   call write string
   exit
```

Date: M T W T F S S

(Q6) .data

A dword 100

B dword 200

c dword ?

i dword 5

J dword 5

.code:

mov ecx,i

mov ebx,b

mov eax,a

L1:

push ecx

add ebx, eax

push eax

mov eax,ebx

call writedec

pop eax

mov ecx,5

L2:

~~sub eax,1~~ sub eax,1

add c, 10

loop l2

pop ecx

call writedec

~~Mov eax,c~~

~~call writedec~~

push eax

mov eax, c

call writedec

pop eax

loop l1

(Q7)

```
        .code
~~code~~ mov eax,0
call ~~io~~ readdec
 mov   ecx, ~~0~~ eax
  L1:
   ~~mov eax~~
     mov  B,ecx
  L2:
   cell writedec
   sub eax,1
   loop L2
~~Mode~~ mov ecx,B
         cell crlf
   loop1
  exit
  main endp
  end main
```

(Q8)

```
 .data
   notset  "parity not set", 0
   set     "parity set", 0

.code
  mov eax,0
  clc                          clear carry
  mov al, 01110101b
   lahf
              parity bit
      bt ax,5
      Jc l1
      mov edx, offset notset
      call writestring
      exit
      l1:
       mov edx, offset set
       call write string
       exit
```