COAL

Assignment #2

Name: Ritesh Kumar
Roll no: 21K-3961
Sec: BSCCS)-3E

Question # 1:

RESULT is not ~~defe~~ declared so MOV RESULT, EAX will give an error.

Before "ret" statement, : pop esi should be there

i.e:

```
POP ESI
ret
Addition ENDP
```

(i)

11500000
5
6

← 00001ffECh
← 00001ff0h
← 00001ff4h
← 00001ff8h (initially)

(ii)

25	← 00001FE8h
11500000	← 00001FECh
5	← 00001FF0h
6	← 00001FF4h
	← 00001FF8h (initially)

The push offset [x1+4] had pushed 27h into the stack making ESP's value as 00001FE4.
But the it was poppeed into ESI so the ESP became 00001FE8h and the value of ESI is 27 but after add [ESI], EAX there will be some garbage value as EAX has nothing moved in it.

Question #2:

```
00100101
00001001
00000001   Line 2 ZF=0

00100100
00001001
00000000   Line 4 Zf=1
```

Question #3:

```
include irvine32.inc
.data
arr1 sdword 40,-90,-67,98,78,-45,0,32
arr2 sdword 5 DUP(?)
.code
main proc
mov esi, offset arr1
mov ecx, lengthof arr1
pushad
call positivevalues
popad
exit
main endp
```

21k-3961

```
positivevalues proc
mov edi, offset arr2
L:
mov eax, [esi]
cmp eax, 0
jL dontmove
mov [edi], eax
add edi, 4

dontmove:
add esi, 4
loop L
ret
positivevalues endp
end main
```

21K-3961

Question #1:

```
include irvine32.inc
.data
A sdword 10
B sdword 5          ; initialized random values
N sdword 3
.code
_while:
mov eax, N
cmp eax, 0
jle _endwhile
cmp eax, 3
je _else
mov eax, A
cmp N, eax
jnl checkB
sub N, 2
jmp _while
checkB:
mov eax, B
cmp N, eax
jng _else
sub N, 2
jmp _while

_else:
dec N
jmp _while
_endwhile:
exit
main endp
end main
```

Question #5:

```
include irvine32.inc
.data
prompt byte "Enter any number between
            0 and 5: ", 0
prompt1 byte "0", 0
prompt2 byte "e", 0
.code
main proc
mov edx, offset prompt
call writestring
call readdec
cmp al, 0
jng ended
cmp al, 5
jnl enddd
cmp al, 1
jne checkfor3
mov edx, offset prompt1
call writestring
jmp enddd
checkfor3:
cmp al, 3
jne checkfor2
```

```
mov edx, offset prompt1
call writestring
jmp endd
checkfor2:
cmp al, 2
jne checkfor4
mov edx, offset prompt2
call writestring
jmp endd
checkfor4:
cmp al, 4
jne endd
mov edx, offset prompt2
call writestring
endd:
exit
main endp
end main
```

Question #6:

```
include irvine32.inc
.data
a dword 100
b dword 200
cC dword ?
i dword ?
j dword ?
.code
main proc
mov i, 5
_outer:
cmp i, 0
jng endd
mov eax, b
add eax, a
mov b, eax
call writedec
call crlf
mov j, 5
_inner:
cmp j, 0
jng endinner
dec a
add cC, 10
mov eax, a
call writedec
call crlf
dec j
jmp _inner
endinner:
dec i
jmp _outer
endd:
exit
main endp
end main
```

Question #7:

```
include irvine32.inc
.data
prompt byte "Enter any number: ", 0
space byte " ", 0
.code
mov edx, offset promp
call writestring
call readdec
mov ebx, eax
mov ecx, eax
L1:
push ecx
L2:
call writedec
mov edx, offset space
Call writestring
dec eax
loop L2
call crlf
mov eax, ebx
pop ecx
loop L1
exit
main endp
end main
```

21K-3961

Question #8:

```
include irvine32.inc
.data
msg byte 01110101b
prompt1 byte "Message has even parity!", 0
prompt2 byte "Message has odel parity!", 0
.code
main proc
mov eax, 0
mov al, msg
add al, 00000010b
jp notevenn
jnp even
noteven:
mov edx, offset prompt2
call writestring
jmp endd
even:
mov edx, offset prompt1
call writestring
endd:
exit
main endp
end main
```