EECS 349

Tianxin Jiang

10/24/2019

Question 1

```
# allocate stack memory
push ebp
mov ebp,esp
and esp,0FFFFFFF0h
sub esp,20h
call ___main

# initialize variables
mov dword ptr [esp+1Ch],3
mov dword ptr [esp+18h],5
mov dword ptr [esp+14h],0

# eax = [esp+1Ch]*[esp+18h]
mov eax,[esp+1Ch]
imul eax,[esp+18h]

# edx = eax
mov edx,eax
```

```
# eax = [esp+1Ch]
mov eax,[esp+1Ch]

# ecx = eax unsigned right shifts 31
mov ecx,eax
shr ecx,1Fh

# eax = eax + ecx
add eax,ecx

# eax = eax signed right shifts 1
sar eax,1

# edx = edx - eax
sub edx,eax

# [esp+14h] = edx
mov eax,edx
mov [esp+14h],eax

# [esp+4] = [esp+14h]
mov eax,[esp+14h]
mov [esp+4],eax
```

```
# printf("%d", [esp+4])
mov dword ptr [esp], offset aD;"%d"00404000
call _printf

# return 0
mov eax,0
leave
retn
_main endp
```

Question 2

```
# allocate stack memory
push ebp
mov ebp,esp
and esp,0FFFFFFF0h
sub esp,40h
call ___main

# initialize variables
mov dword ptr [esp+18h],0Ch
mov dword ptr [esp+1Ch],0Fh
mov dword ptr [esp+20h],0DDh
mov dword ptr [esp+24h],3
mov dword ptr [esp+28h],1B0h
mov dword ptr [esp+2Ch],36h
mov dword ptr [esp+30h],10h
mov dword ptr [esp+34h],43h
mov dword ptr [esp+3Ch],0
mov dword ptr [esp+38h],0

# jump to loc_40157F
jmp short loc_40157F

loc_401560:
# eax = [esp+38h] ([esp+38h] like a index of array)
mov eax,[esp+38h]

# eax = [esp+18h+eax*4]
mov eax,[esp+eax*4+18h]

# if (eax <= [esp+3Ch]) then jump to loc_40157A
cmp eax,[esp+3Ch]
jle short loc_40157A

# eax = [esp+38h]
mov eax,[esp+38h]
```

```
# [esp+3Ch] = [esp+18h+eax*4]
mov eax,[esp+eax*4+18h]
mov [esp+3Ch],eax

loc_40157A:
# [esp+38h] = [esp+38h] + 1
add dword ptr [esp+38h],1

loc_40157F:
# if([esp+38h] <= 7) then jump to loc_401560
cmp dword ptr [esp+38h],7
jle short loc_401560

# printf("%d", [esp+3Ch])
mov eax,[esp+3Ch]
mov [esp+4],eax
mov dword ptr [esp],offset aD;"%d"
call _printf

# return 0
mov eax,0
leave
retn    c3
_main endp
```

Question 3

```
# allocate stack memory
push ebp
mov ebp,esp
and esp,0FFFFFFF0h
sub esp,20h
call ___main

# [esp+1Ch] = 64h = 0x64 = 100
mov dword ptr [esp+1Ch],64h

# jump to loc_4015D6
jmp loc_4015D6

loc_40151B:
# ecx = [esp+1Ch]
mov ecx,[esp+1Ch]

# edx = 51EB851Fh = 0x51EB851F
mov edx,51EB851Fh

# eax = ecx
```

```
mov eax,ecx

# edx:eax = eax * edx = [esp+1Ch] * 0x51EB851F
imul edx

# edx = edx signed right shifts 5
sar edx,5

# eax = ecx
mov eax,ecx

# eax = eax signed right shifts 0x1F
sar eax,1Fh

# edx = edx - eax
sub edx,eax

# [esp+18h] = edx
mov eax,edx
mov [esp+18h],eax

# eax = [esp+18h]
mov eax,[esp+18h]
```

```
# edx = eax * (-0x64)
imul edx,eax,-64h

# eax = [esp+1Ch]
mov eax,[esp+1Ch]

# ecx = (edx + eax)
lea ecx,[edx+eax]

# edx = 66666667h = 0x66666667
mov edx, 66666667h

# eax = ecx
mov eax,ecx

# edx:eax = eax*edx
imul edx

# edx = edx signed right shifts 2
sar edx,2

# eax = ecx
mov eax,ecx
```

```
# eax = eax signed right shifts 0x1F
sar eax,1Fh

# edx = edx - eax
sub edx,eax

# eax = edx
mov eax,edx

# [esp+14h] = eax
mov [esp+14h],eax

# ecx = [esp+1Ch]
mov ecx,[esp+1Ch]

# edx = 0x66666667
mov edx, 66666667h

# eax = ecx
mov eax,ecx

# edx:eax = eax*edx
imul edx
```

```
# edx = edx signed right shifts 2

sar edx,2


# eax = ecx

mov eax,ecx


# eax = eax signed right shifts 0x1F

sar eax,1Fh


# edx = edx - eax

sub edx,eax


# eax = edx

mov eax,edx


# eax = eax unsigned right shifts 2

shl eax,2


# eax = eax + edx

add eax,edx


# eax = eax + eax

add eax,eax
```

```
# ecx = ecx - eax

sub ecx,eax


# eax = ecx

mov eax,ecx


# [esp+10h] = eax

mov [esp+10h],eax


# eax = [esp+18h]

mov eax,[esp+18h]


# eax = eax * [esp+18h]

imul eax,[esp+18h]


# eax = eax * [esp+18h]

imul eax,[esp+18h]


# edx = eax

mov edx,eax


# eax = [esp+14h]

mov eax,[esp+14h]
```

```
# eax = eax * [esp+14h]
imul eax,[esp+14h]

# eax = eax * [esp+14h]
imul eax,[esp+14h]

# edx = edx + eax
add edx,eax

# eax = [esp+10h]
mov eax,[esp+10h]

# eax = eax * [esp+10h]
imul eax,[esp+10h]

# eax = eax * [esp+10h]
imul eax,[esp+10h]

# eax = eax + edx
add eax,edx

# if (eax != [esp+1Ch]) then jump to loc_4015D1
cmp eax,[esp+1Ch]
jnz short loc_4015D1
```

```
# printf("%d ", [esp+1Ch])
mov eax,[esp+1Ch]
mov [esp+4],eax
mov dword ptr [esp],offset aD;"%d "
call _printf


loc_4015D1:
#   [esp+1Ch] = [esp+1Ch] + 1
add dword ptr [esp+1Ch],1


loc_4015D6:
# if( [esp+1Ch] <= 0x3E7 ) then jump to  loc_40151B
cmp dword ptr [esp+1Ch],3E7h
jle loc_40151B


# return 0
mov eax,0
leave
retn
_main endp
```