Q2:

```
.text:00401500              push    ebp
.text:00401501              mov     ebp, esp
.text:00401503              and     esp, 0FFFFFFF0h
.text:00401506              sub     esp, 40h
.text:00401509              call    ___main
.text:0040150E              mov     dword ptr [esp+18h], 0Ch // set memory to 12
.text:00401516              mov     dword ptr [esp+1Ch], 0Fh // set memory to 15
.text:0040151E              mov     dword ptr [esp+20h], 0DDh // set memory to 221
.text:00401526              mov     dword ptr [esp+24h], 3      // set memory to 3
.text:0040152E              mov     dword ptr [esp+28h], 1B0h // set memory to 432
.text:00401536              mov     dword ptr [esp+2Ch], 36h // set memory to 54
.text:0040153E              mov     dword ptr [esp+30h], 10h // set memory to 16
.text:00401546              mov     dword ptr [esp+34h], 43h  // set memory to 67
.text:0040154E              mov     dword ptr [esp+3Ch], 0     // set memory to 0
.text:00401556              mov     dword ptr [esp+38h], 0      // set memory to 0
// Note_start
The first 8 memory address assignments are all 4 bits apart in address so I created an array
called “arr” to hold all 8 values (which yielded the same assembly code)
I created an int max (will represent maximum value) set to 0 at [esp+3Ch]
I created an int x (an index essentially) set to 0 at [esp+38h]
// Note_end
.text:0040155E              jmp     short loc_40157F            // jump to loc_40157F
.text:00401560 ; ---------------------------------------------------
.text:00401560
.text:00401560 loc_401560:                          ; CODE XREF: _main+84↓j
.text:00401560              mov     eax, [esp+38h]              // eax = x
.text:00401564              mov     eax, [esp+eax*4+18h]        // eax = arr[x]
.text:00401568              cmp     eax, [esp+3Ch]              // if(arr[x] <=max)
.text:0040156C              jle     short loc_40157A            // jump to loc_40157A
.text:0040156E              mov     eax, [esp+38h]              // eax = x
.text:00401572              mov     eax, [esp+eax*4+18h]        // eax = arr[x]
.text:00401576              mov     [esp+3Ch], eax              // max = arr[x]
.text:0040157A
.text:0040157A loc_40157A:                          ; CODE XREF: _main+6C↑j
.text:0040157A              add     dword ptr [esp+38h], 1      // increment x
.text:0040157F
.text:0040157F loc_40157F:                          ; CODE XREF: _main+5E↑j
.text:0040157F              cmp     dword ptr [esp+38h], 7      // if(x <= 7)
.text:00401584              jle     short loc_401560            // jump to loc_401560
// Note_start
The above two lines are a while(x <= 7) loop
// Note_end
```

```
.text:00401586                mov     eax, [esp+3Ch]                  // set eax to max
.text:0040158A                mov     [esp+4], eax                    // temp memory for print func
.text:0040158E                mov     dword ptr [esp], offset aD ; "%d"
.text:00401595                call    _printf                         // print value of max
.text:0040159A                mov     eax, 0
.text:0040159F                leave
.text:004015A0                retn                                    // return 0
.text:004015A0 _main          endp
```