Q3:

```
.text:00401500              push    ebp
.text:00401501              mov     ebp, esp
.text:00401503              and     esp, 0FFFFFFF0h
.text:00401506              sub     esp, 20h
.text:00401509              call    ___main
.text:0040150E              mov     dword ptr [esp+1Ch], 64h // int a = 100
.text:00401516              jmp     loc_4015D6                  // jump to loc_4015D6
.text:0040151B ; ---------------------------------------------------------------------------
.text:0040151B
.text:0040151B loc_40151B:                          ; CODE XREF: _main+DE↓j
.text:0040151B              mov     ecx, [esp+1Ch]
.text:0040151F              mov     edx, 51EB851Fh
.text:00401524              mov     eax, ecx
.text:00401526              imul    edx
.text:00401528              sar     edx, 5
.text:0040152B              mov     eax, ecx
.text:0040152D              sar     eax, 1Fh
.text:00401530              sub     edx, eax
.text:00401532              mov     eax, edx
.text:00401534              mov     [esp+18h], eax              // [esp+18h] int b = a / 100
.text:00401538              mov     eax, [esp+18h]
.text:0040153C              imul    edx, eax, -64h
.text:0040153F              mov     eax, [esp+1Ch]
.text:00401543              lea     ecx, [edx+eax]              // temp ecx = -100y + x
.text:00401546              mov     edx, 66666667h
.text:0040154B              mov     eax, ecx
.text:0040154D              imul    edx
.text:0040154F              sar     edx, 2
.text:00401552              mov     eax, ecx
.text:00401554              sar     eax, 1Fh
.text:00401557              sub     edx, eax
.text:00401559              mov     eax, edx                    // [sep+14h] int c = temp / 10
.text:0040155B              mov     [esp+14h], eax              so int c = (-100*y+x) /10
.text:0040155F              mov     ecx, [esp+1Ch]      // ecs = a
.text:00401563              mov     edx, 66666667h
.text:00401568              mov     eax, ecx
.text:0040156A              imul    edx
.text:0040156C              sar     edx, 2
.text:0040156F              mov     eax, ecx
.text:00401571              sar     eax, 1Fh
.text:00401574              sub     edx, eax            // edx = a / 10
.text:00401576              mov     eax, edx            // eax = a / 10
```

```
.text:00401578              shl     eax, 2                  // eax = a / 10 * 4
.text:0040157B              add     eax, edx                // eax = (a / 10)*4 + a/10 =  a/10 * 5
.text:0040157D              add     eax, eax                // eax = (a/10*5) + (a/10*5)
.text:0040157F              sub     ecx, eax                // ecx = ecx - eax = a - 2 * (a/10*5)
.text:00401581              mov     eax, ecx                // eax = a-2*(a/10*5)
.text:00401583              mov     [esp+10h], eax          // int d = a-2*(a/10*5)
.text:00401587              mov     eax, [esp+18h]          // eax = b
.text:0040158B              imul    eax, [esp+18h]          // eax = b*b
.text:00401590              imul    eax, [esp+18h]          // eax = b*b*b
.text:00401595              mov     edx, eax                // edx = b*b*b
.text:00401597              mov     eax, [esp+14h]          // eax = c
.text:0040159B              imul    eax, [esp+14h]          // eax = c*c
.text:004015A0              imul    eax, [esp+14h]          // eax=c*c*c
.text:004015A5              add     edx, eax                // edx = b*b*b+c*c*c
.text:004015A7              mov     eax, [esp+10h]          // eax = d
.text:004015AB              imul    eax, [esp+10h]          // eax = d*d
.text:004015B0              imul    eax, [esp+10h]          // eax = d*d*d
.text:004015B5              add     eax, edx                // eax = b*b*b+c*c*c+d*d*d
.text:004015B7              cmp     eax, [esp+1Ch]          // comp eax to a
.text:004015BB              jnz     short loc_4015D1        // jump if a ==  b*b*b+c*c*c+d*d*d
.text:004015BD              mov     eax, [esp+1Ch]          // eax = a
.text:004015C1              mov     [esp+4], eax            // [esp+4] temp hold a
.text:004015C5              mov     dword ptr [esp], offset aD ; "%d  "
.text:004015CC              call    _printf .text:004015D1 // Prints out a
.text:004015D1 loc_4015D1:                                  ; CODE XREF: _main+BB↑j
.text:004015D1              add     dword ptr [esp+1Ch], 1 // Increment a: a = a + 1
.text:004015D6
.text:004015D6 loc_4015D6:                                  ; CODE XREF: _main+16↑j
.text:004015D6              cmp     dword ptr [esp+1Ch], 3E7h
.text:004015DE              jle     loc_40151B  // while a < 1000, goto loc_4015D6
.text:004015E4              mov     eax, 0
.text:004015E9              leave .text:004015EA                retn
.text:004015EA _main        endp
```