Assignment # 01

Deadline: Monday, Jan 02, 2017 11:59PM

Submission Instructions:

- Submit a single 8086 assembly language program file.
- Your file name must be A1_[your-roll-no].asm (e.g., A1_BCSF15M001.asm)
- You have to submit the file through email, and the title of your email must be Assignment1_[your-section]_[your-roll-no] (e.g., Assignment1_Morning_BCSF15M001)
- You must follow the naming conventions, or you will be awarded with a **ZERO** in this assignment.
- You have to email your solution of this assignment to ahmad.muhammad@pucit.edu.pk before the deadline.
- Any submission which is after the deadline, or which doesn't follow the naming conventions, will be discarded and will not be evaluated, so you'll have to be careful about the naming conventions. (Believe me, you don't want to throw away your hard work just because you didn't follow the naming conventions, so be careful while submitting your solution)

Description and Requirement:

You've to make a replica of the dump command (d) of debug. This assignment only requires the first deliverable for this task, which includes:

- Writing a program that displays data from the memory in the segment address **073FH**, starting from the offset address **0100H**.
- The program should display the data from memory **EXACTLY** as the dump commands displays. E.g.,

```
-d
073F:0100  75 17 83 3E 76 5C 00 75-05 2B C0 99 EB 05 AE FE   u..>v\.u.+......
073F:0110  00 F0 46 74 00 00 B2 00-B2 14 99 00 2E 07 2E 07   ..Ft............
073F:0120  D2 75 04 23 C0 74 08 83-46 0C 10 83 56 0E 00 83   .u.#.t..F...V...
073F:0130  7E 12 FF 75 10 83 7E 10-FF 75 0A C7 46 10 00 00   ~..u..~..u..F...
073F:0140  C7 46 12 00 00 8B 46 10-8B 56 12 23 D2 75 04 23   .F....F..V.#.u.#
073F:0150  C0 74 08 83 46 10 10 83-56 12 00 83 7E 16 FF 75   .t..F...V...~..u
073F:0160  13 83 7E 14 FF 75 0D A1-60 1A 8B 16 62 1A 89 46   ..~..u..`...b..F
073F:0170  14 89 56 16 8B 46 14 8B-56 16 23 D2 75 04 23 C0   ..V..F..V.#.u.#.
```

- Your program should display a total of **128** bytes in **8** lines, each line containing **16** bytes.

Let's analyze the format of first row of the output:

```
073F:0100  75 17 83 3E 76 5C 00 75-05 2B C0 99 EB 05 AE FE   u..>v\.u.+......
```

There are a few things to note:

- Each value is displayed in hexa-decimal form.
- First of all, the address is displayed at the start of the line (e.g., **073F:0100**), which is in the format **SegmentAddress:OffsetAddress** i.e., **073F** is segment address, and **0100** is offset address.
- After the address, there are two spaces.
- Then **8** bytes are displayed in hexa-decimal form, each byte separated by a space. (i.e., **75 17 83 3E 76 5C 00 75**)
- And then, there's a hyphen (-).
- After the hyphen, the next 8 bytes are displayed in hexa-decimal form, each byte separated by a space (i.e., **05 2B C0 99 EB 05 AE FE**)
- Then there are 3 spaces.
- After the spaces, the ASCII dump of the **16** bytes is displayed, i.e., (**u..>v\.u.+......**). In the ASCII dump, you can easily notice that, not all of the ASCII equivalent characters of the bytes are displayed, instead a dot is displayed against them.
- If the value of the byte is from **20H** to **7EH**, then their ASCII equivalents are displayed, otherwise you'll see a dot against them. i.e., you can see the last 6 bytes of the first line (**C0 99 EB 05 AE FE**), a dot is displayed against each of them.