Assignment Descriptions:

In this assignment, we will develop an IP-address analyzer and an IP routing table lookup*, using MIPS R-3000 assembly instruction set. IP addresses are unique identifier for each host computer connected to the Internet in the format of “N1.N2.N3.N4”, where each of N1, N2, N3, and N4, is an integer 0 through 255 (e.g., “146.163.150.56”), as shown in Figure 1. Each computer connected to the Internet is assigned a unique IP address. Since a unique IP address is assigned to a unique host computer in the Internet, other computers identify and locate each host computer in the Internet using IP addresses.

![Image of ipconfig output]

Figure 1 – IP address format

In the Internet, each individual computer belongs to a group of computers, called “domain”. For example, SIUE is a domain in the Internet, and every computer in SIUE campus belongs to SIUE domain. To reflect this design, each IP address consists of two internal address components of “domain address” and “host address”. Host addresses represent a particular domain, while host address represent the address of an individual computer in a domain. Regarding the domain address, the Internet assumes three different classes of domains, known as class-A, class-B, and class-C domains. Class-D domains are special domains, called “multicast domains”.

The three different domain classes represent domains in different sizes (number of computers that can be connected to a domain). For example, each class-A domain can have up to 2^24 host computers, while a class-B domain can have up to 2^16 computers. A class-C domain can have up to 2^8 computers. The first number (N1) in each IP address indicates the class of a domain as
shown in Figure 2. Each IP address is represented as a combination of four positive integers, where each integer can take 0 ~ 255 (as an eight-bit unsigned integer). This means that each IP address consists of a total of 32 bits (4 bytes). Thus, there are \(2^{(32-24)} = 2^8 (= 128)\) class-A domains in the Internet. Similarly, there are \(2^{(32-16)} = 65,536\) class-B domains and \(2^{(32-8)} = 2^{24}\) class-C domains in the Internet. The four domain classes of IP addresses are classified using the first number in IP addresses as follows.

<table>
<thead>
<tr>
<th>(N_1)</th>
<th>Domain Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>1~127</td>
<td>A</td>
</tr>
<tr>
<td>128~191</td>
<td>B</td>
</tr>
<tr>
<td>192~223</td>
<td>C</td>
</tr>
<tr>
<td>224~239</td>
<td>D</td>
</tr>
</tbody>
</table>

**Figure 2 – \(N_1\) for determining IP domain classes**

Hardware equipment, called “routers”, are responsible for forwarding network traffic into a correct direction in the Internet and they use IP addresses to do so. The following table shows how routers recognize IP addresses. Figure 3 shows how routers recognize each IP address (“Network” in the figure means “Domain Address” while “Host” means “Host Address”). As long as the first bit in a given IP address is ‘0’, it must be a class-A domain, while the first two bits are ‘10’ (the first ‘0’ appears in the second bit), it must belong to a class-B domain. Similarly, if the first three bits are ‘110’ (the first ‘0’ appears in the third bit), it must belong to a class-C domain. For example, SIUE’s domain address is “146.163” and Dr. Fujinoki’s PC’s (at his office at SIUE) host address is “150.56” – its IP address is “146.163.150.56”).

**Figure 3 – the bit pattern layout for the four classes of the IP addresses**

**Assignment Requirements:**

**Program 1 (30/100):** Develop a program (using MIPS R-3000 assembly instructions) that analyzes a given IP address for its domain class and displays the domain class the given IP address belongs to. Your program should prompt for an IP address and display its domain class (as shown below).
Any invalid input should be detected as soon as a number is entered and your program should repeat a prompt until a valid number is entered. See the figure below.

Once a valid number is entered, any valid input should not be repeated (as shown below).

After all the four numbers are correctly input, your program should determine its domain class and display it.
**Program 2** (70/100): On top of your Program 1, develop a program that scans an IP routing table (as shown in Table 1 in the next page) for a matching line in the table in terms of the domain addresses (the host addresses should be ignored). Your search program should satisfy the following requirements:

1. Your search program should find “the line number” for the first match. If more than one line matches, your program should display the first match.
2. If there is no match, your program should show that there is no match after all the lines in a routing table scanned.

The following two figures show a successful and an unsuccessful search results.

---

### Console

```
enter an IP address
first: 148
second: 146
third: 1
fourth: 1
the IP address you entered: 148.146.1.1

class B address
Matching domain found at: 0
program successfully completed ...
```

---

### Console

```
enter an IP address
first: 149
second: 234
third: 45
fourth: 45
the IP address you entered: 149.234.45.45

class B address
Matching domain was NOT found ...
program successfully completed ...
```
Each routing table is defined using “.word” PC-SPIM directive as shown below. The size of an IP routing table is defined using the label “IP_ROUTING_TABLE_SIZE:”. The beginning of the routing table is defined using the label “IP_ROUTING_TABLE”. Each number declared using “.word” directive is a four-byte integer in memory. The following sample IP routing will be posted to CS286 course home and you should use (you can modify it as you like) for your debugging. When we (Dr. Fujinoki and the course TA) grade your program after your final submission, another IP routing table (but in the same format) will be used for testing. Each of you should copy the whole IP routing table (include “IP_ROUTING_TABLE_SIZE:”) at the beginning of your *.asm source code file (in its “data section”).

<table>
<thead>
<tr>
<th>IP_ROUTING_TABLE_SIZE:</th>
<th>.word 10</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IP_ROUTING_TABLE</strong>:</td>
<td># line #, x.x.x.x -----------------------------------</td>
</tr>
<tr>
<td>.word 0, 146, 163, 255, 255 # 146.163.255.255</td>
<td></td>
</tr>
<tr>
<td>.word 1, 147, 163, 255, 255 # 147.163.255.255</td>
<td></td>
</tr>
<tr>
<td>.word 2, 201, 88, 88, 90 # 201.88.88.90</td>
<td></td>
</tr>
<tr>
<td>.word 3, 182, 151, 44, 56 # 182.151.44.56</td>
<td></td>
</tr>
<tr>
<td>.word 4, 24, 125, 100, 100 # 24.125.100.100</td>
<td></td>
</tr>
<tr>
<td>.word 10, 146, 163, 140, 80 # 146.163.170.80</td>
<td></td>
</tr>
<tr>
<td>.word 11, 146, 163, 147, 80 # 146.163.147.80</td>
<td></td>
</tr>
<tr>
<td>.word 12, 146, 164, 147, 80 # 146.164.147.80</td>
<td></td>
</tr>
<tr>
<td>.word 20, 148, 146, 170, 80 # 148.146.170.80</td>
<td></td>
</tr>
<tr>
<td>.word 30, 193, 77, 77, 10 # 193.77.77.10</td>
<td></td>
</tr>
</tbody>
</table>

Table 1 – a sample IP routing table

**Requirement 1**: This project is an individual project. Absolutely no “source code sharing” (both “partial sharing” and “whole sharing” are strictly prohibited).

**Requirement 2**: In your source code file, clearly show a reference you used to complete this project (book, web site in the Internet, or any other source), if you used any resource(s) other than those available in the course textbook.
Submissions:

- Extra-credit deadline: 23:59:59 on November 27th
- Free-feedback deadline: 23:59:59 on November 29th
- Your source code file should be e-mailed to the TA (ksenthi@siue.edu) by 11:00 A.M. on December 4th.
- Your program hardcopy is due at the beginning of the class on the due date (5 points penalty to your project #3 for failing to make a hardcopy submission on the due).

Note: In your program source code, include your student ID (the last 3 digits) in your program header.

Late Submission:

- Penalty of -10% will be given for every 24 hours after the due (i.e., -10% for a submission within the first 24 hours after the due).
- Submission more than 48 hours after the due will not be accepted.

Grading Criteria:

- **Assembler error**: If the TA can not assemble your assembler source code file due to any assembler error: 0 ~ 20% of the credit for this assignment phase will be given.

- **Run-time error**: After your assembly source code file is successfully assembled to machine codes, if your program does not satisfy any requirement: 5% penalty for each minor problem. For any major failure, penalty depends on each such major problem.