CS 315: Computer Networks Lab
Spring 2024-25, IIT Dharwad
Assignment-13
NS-3
April 21, 2025

Chidurala Tejaswini
(220010012 / CS22BT012)

Introduction

The ns-3 simulator is a discrete-event network simulator targeted primarily for research and educational use. The ns-3 project, started in 2006, is an open-source project developing ns-3.

In brief, *ns-3* provides models of how packet data networks work and perform, and provides a simulation engine for users to conduct simulation experiments. Some of the reasons to use *ns-3* include to perform studies that are more difficult or not possible to perform with real systems, to study system behavior in a highly controlled, reproducible environment, and to learn about how networks work.

This tutorial aims to introduce new ns-3 users to the system in a structured way. New users sometimes struggle to glean essential information from detailed manuals and convert it into working simulations. In this tutorial, we will build some example simulations, introducing and explaining key concepts and features as we go.

This assignment includes basic lessons in installation, compilation and some examples of running the ns-3 scripts. In general, follow the instructions in the assignment for a quick start.

System requirement: The ns-3 is more friendly with Linux OS. If you don't have Linux, you can use virtual machines (VMs).

NS-3 Installation Instructions:

The installation steps of NS-3 for different OS is available here: https://www.nsnam.org/wiki/Installation

Step 1: Open a Terminal
$ sudo apt update

$ sudo apt install g++ python3 python3-dev pkg-config sqlite3 cmake python3-setuptools git qtbase5-dev qtchooser qt5-qmake qtbase5-dev-tools gir1.2-goocanvas-2.0 python3-gi python3-gi-cairo python3-pygraphviz gir1.2-gtk-3.0 ipython3 openmpi-bin openmpi-common openmpi-doc libopenmpi-dev autoconf cvs bzr unrar gsl-bin libgsl-dev libgslcblas0 wireshark tcpdump sqlite sqlite3 libsqlite3-dev libxml2 libxml2-dev libc6-dev libc6-dev-i386 libclang-dev llvm-dev automake python3-pip libxml2 libxml2-dev libboost-all-dev

Step 2:
After installing the required packages, download the NS-3.36.1 on this link https://www.nsnam.org/releases/ns-allinone-3.36.1.tar.bz2 and unzip it.

```
$ wget https://www.nsnam.org/releases/ns-allinone-3.36.1.tar.bz2
$ tar -xvf ns-allinone-3.36.1.tar.bz2
```

Step 3: Go to the above folder and input these commands consecutively after each command finishes executing:

```
$ cd ns-allinone-3.36.1/
$ ./build.py --enable-examples --enable-tests
```

Building NetAnim and Running a Simulation

1. To build NetAnim, the qmake package will be utilized in the following process:

   a. Go to the NetAnim directory pasting these commands in the terminal:
   ```
   cd <ns folder name>   (ex: cd ns-allinone-3.36.1)
   cd <netanim folder name>  (ex: cd netanim-3.108)
   ```

   b. Clean make files using the command:
   ```
   make clean
   ```

   c. Make NetAnim using the commands:
   ```
   qmake NetAnim.pro
   Make
   ```

   d. Test the NetAnim installation by pasting the following command in the terminal, while within the netanim directory:
   ```
   ./NetAnim
   ```

2. If NetAnim opens, congratulations! NetAnim is now installed.

3. Building and Running simulation procedures:

   a. Copy a .cc file to the scratch directory in the ns-3.36.1 directory.
(Ex: You can get some default examples in the folder “ns-3.36.1/examples/” folder. We will review some examples in the “ns-3.36.1/examples/tutorial/” folder. However, ns-3 allows only running an experiment script from the “ns-3.36.1/scratch” folder. So you need to copy the examples (.cc files from the tutorial folder) to the “ns-3.36.1/scratch” folder.)

   b. Edit the copied .cc file to include the netanim library files by pasting in the following code to the code:

#include "ns3/netanim-module.h"

c. Instantiate the animation interface into the code.

d. Exit the scratch directory by using the following command in the terminal:

```
cd ../
```

e. Paste the following commands to the terminal:

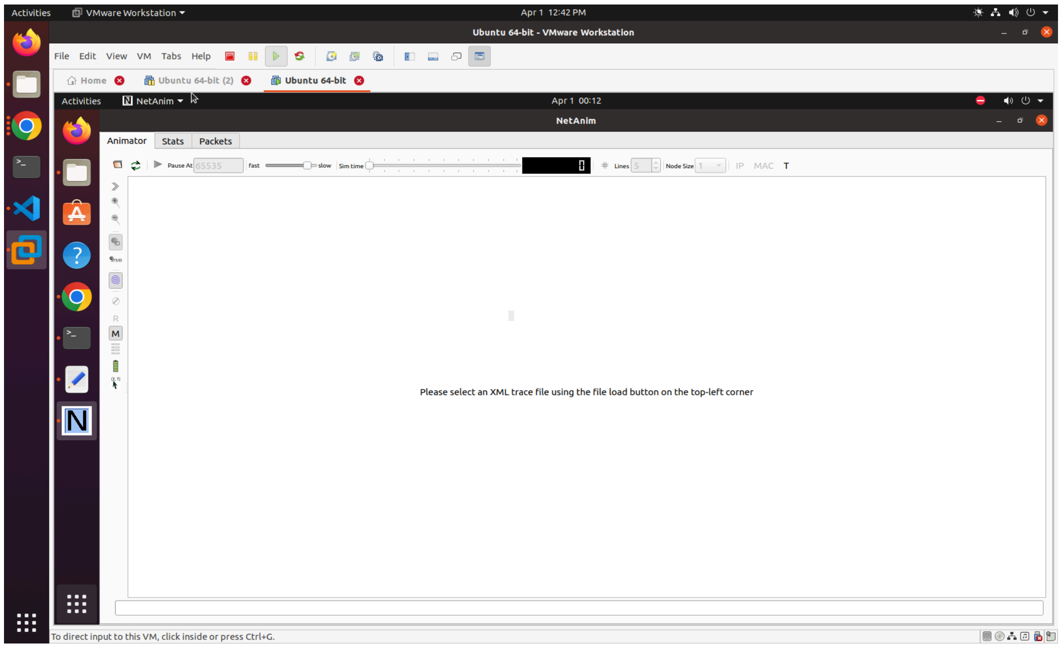
```
./ns3 run scratch/myfirst.cc
```

f. Run NetAnim by entering the netanim directory in the terminal and use the code:

```
./NetAnim
```

g. Select the created xml file and run the resulting simulation by following these procedures:

i. Click on the folder icon in the top left side, shown below.



Please select an XML trace file using the file load button on the top-left corner

ii. Go into the ns-3.36.1 folder and select the xml file you built.

iii. Run the simulation by pressing the play icon.

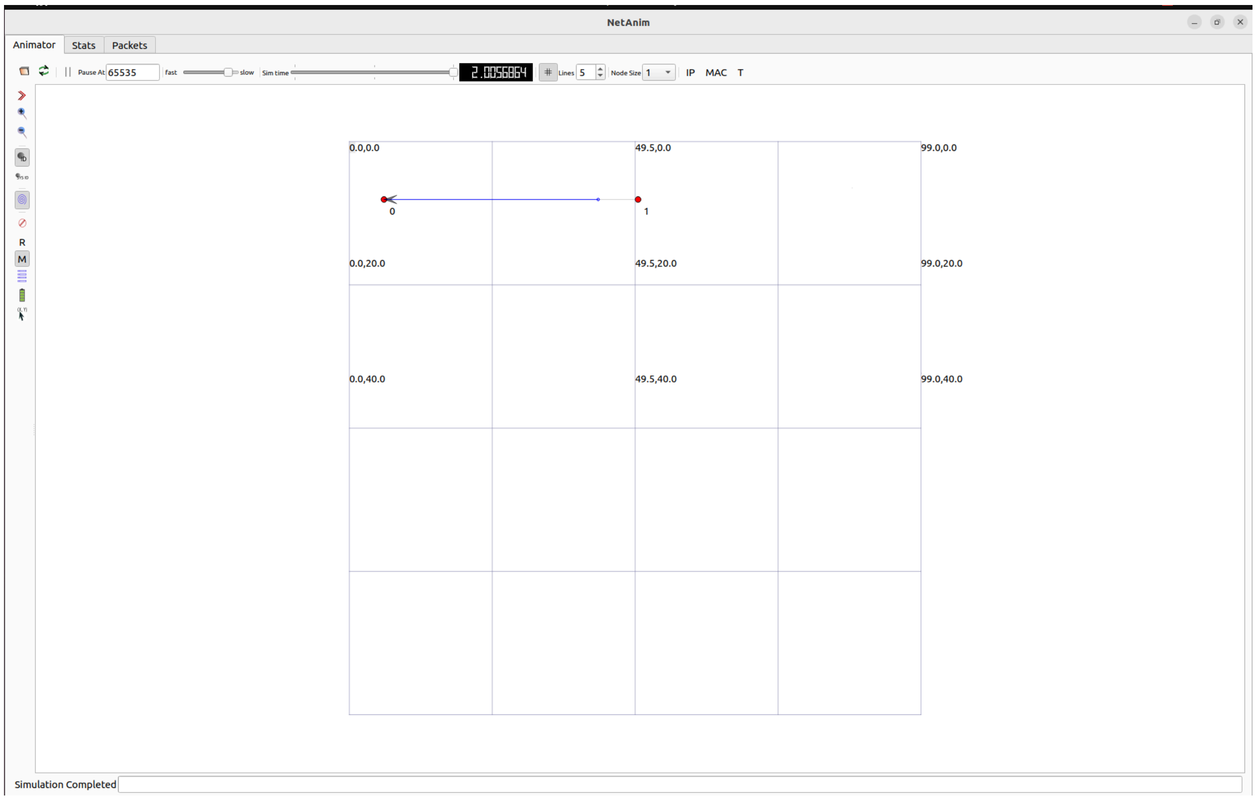
In this lab, create a simple topology of two nodes – Node1 and Node2, separated by a point-to-point link. Setup a UdpClient on Node1 and UdpServer on Node2. Start the client application, and measure the transmission and receiving rate of each node while varying the latency of the link. (this can be referenced from the first.cc code available in "ns-3.36.1/examples/tutorial/" location

Part-1

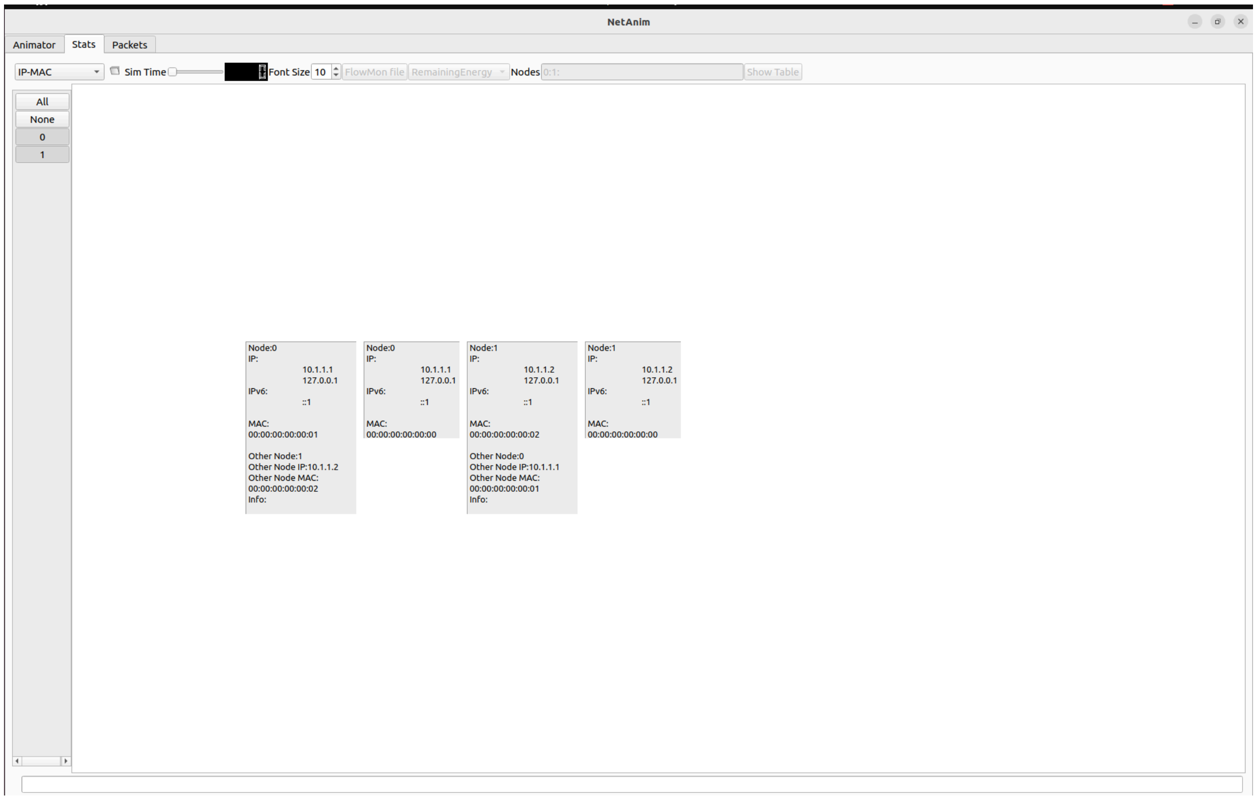
Based on the above topology, you need to add some lines of the code to the first.cc such that you can able to visualize the transmission and receiving of the packets in the Netanim tool. The details of the usage of the Netanim tool are mentioned in the above document for the reference.

1. Submit the code after updating the lines for the Netanim.

2. Attach a screenshot of the animator tool after uploading your XML file



3. **Attach a screenshot of the statistics (or Stats, i.e. available in the animator tool) obtained after running your XML file.**



```
Node:0
IP:
            10.1.1.1
            127.0.0.1
IPv6:
            ::1

MAC:
00:00:00:00:00:01

Other Node:1
Other Node IP:10.1.1.2
Other Node MAC:
00:00:00:00:00:02
Info:
```

```
Node:0
IP:
            10.1.1.1
            127.0.0.1
IPv6:
            ::1

MAC:
00:00:00:00:00:00
```

```
Node:1
IP:
            10.1.1.2
            127.0.0.1
IPv6:
            ::1

MAC:
00:00:00:00:00:02

Other Node:0
Other Node IP:10.1.1.1
Other Node MAC:
00:00:00:00:00:01
Info:
```

```
Node:1
IP:
            10.1.1.2
            127.0.0.1
IPv6:
            ::1

MAC:
00:00:00:00:00:00
```

Part-2

In this section, you will familiarize yourself with the flow monitor module available in the NS-3.

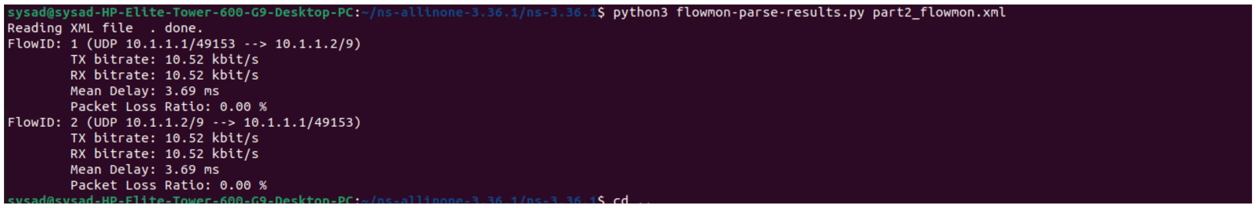
The Flow Monitor module's goal is to provide a flexible system to measure the performance of network protocols. The module uses probes installed in network nodes to track the packets exchanged by the nodes, and it will measure several parameters. Packets are divided according to the flow they belong to, where each flow is defined according to the probe's characteristics (e.g., for IP, a flow is defined as the packets with the same {protocol, source (IP, port), destination (IP, port)} tuple. The statistics collected for each flow can be exported in XML format. Moreover, the user can access the probes directly to request specific stats about each flow.

Based on the above code you have used to solve in part -1, you need to add a few lines of code to it to obtain the flow data rate in terms of each node's transmission and receiving rate and the mean delay(in ms).

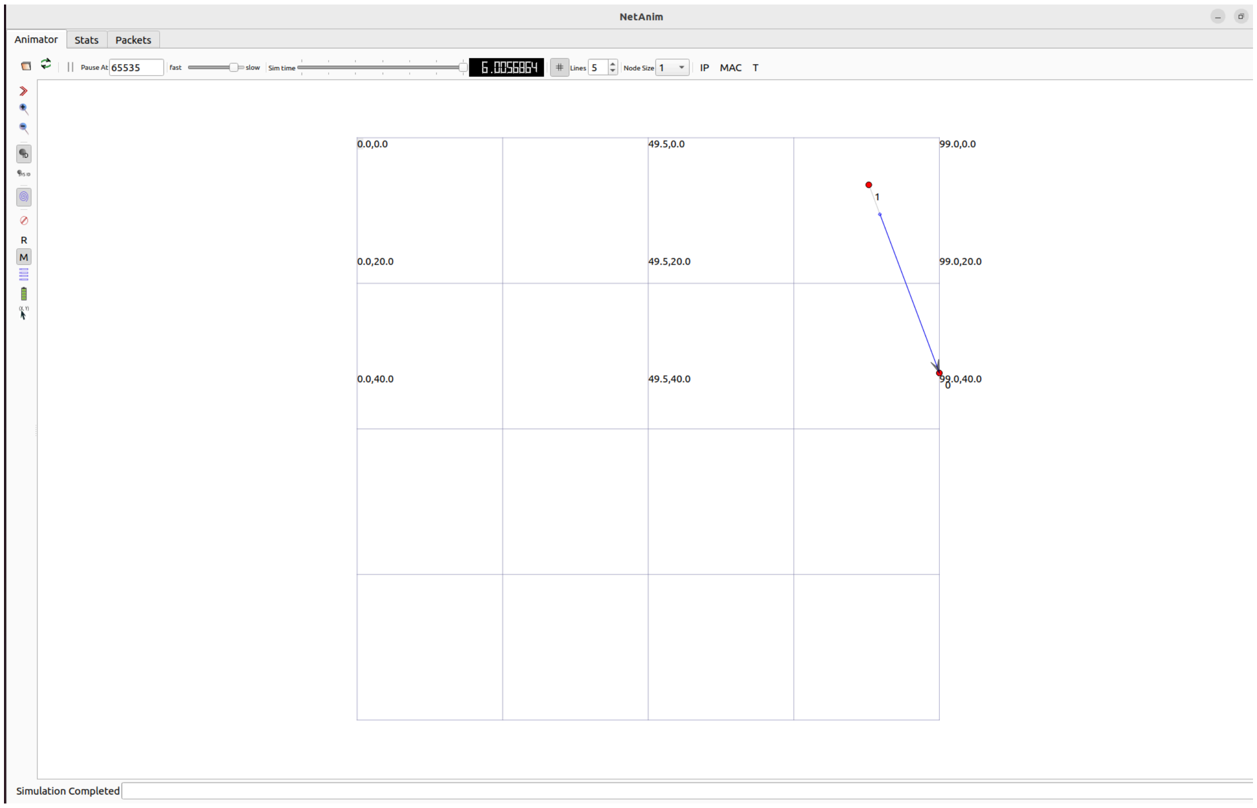
1. **Submit your code after updating the lines for the flow monitor.**

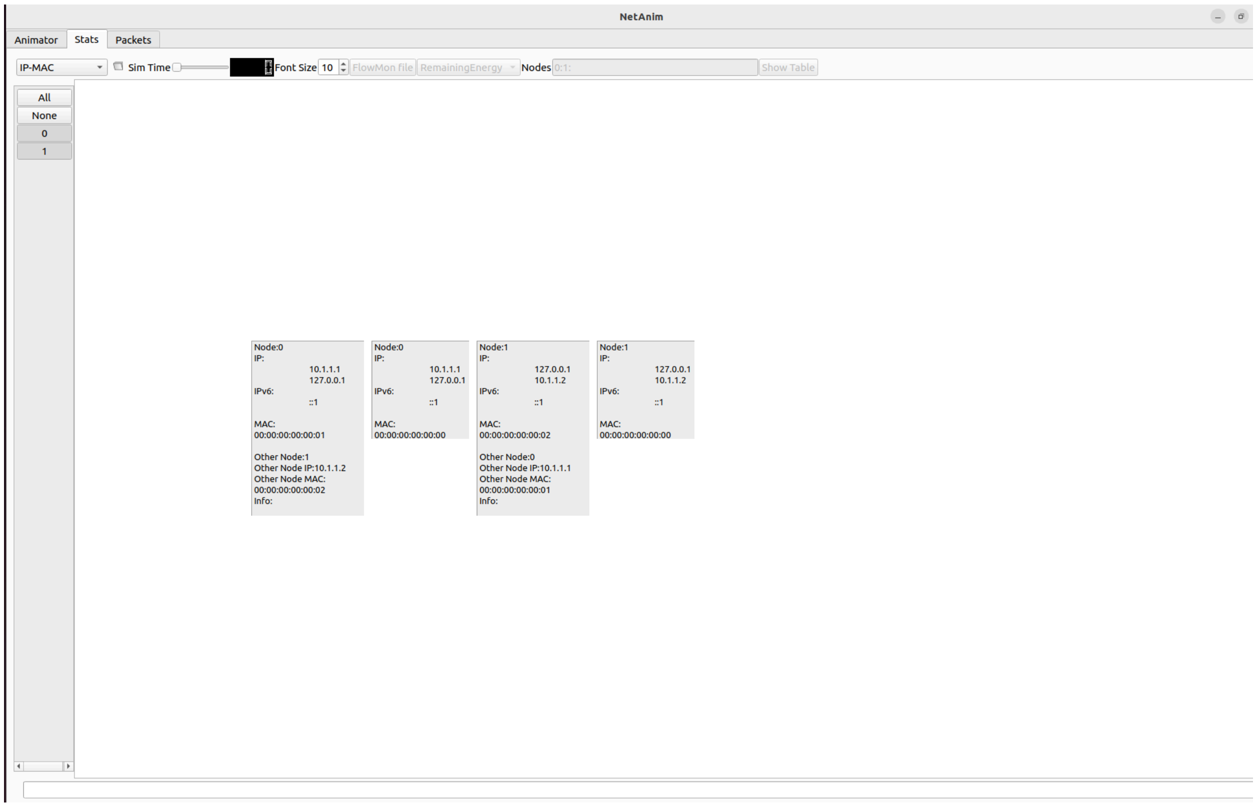
2. **Attach a screenshot of the results obtained after executing the code.**

Note: On compiling the above code, you will obtain another xml file other than that used in Part-1. To obtain the results of transmission and receiving rate after adding the code of flow monitor, first copy the python file “flowmon-parse-results.py” from this “ns-allinone-3.36.1/ns-3.36.1/src/flow-monitor/examples” location to this “ns-allinone-3.36.1/ns-3.36.1” location. After that, you can use this command “python3 flowmon-parse-results.py part-2_file_name.xml” to obtain the results.



```
sysad@sysad-HP-Elite-Tower-600-G9-Desktop-PC:~/ns-allinone-3.36.1/ns-3.36.1$ python3 flowmon-parse-results.py part2_flowmon.xml
Reading XML file  . done.
FlowID: 1 (UDP 10.1.1.1/49153 --> 10.1.1.2/9)
        TX bitrate: 10.52 kbit/s
        RX bitrate: 10.52 kbit/s
        Mean Delay: 3.69 ms
        Packet Loss Ratio: 0.00 %
FlowID: 2 (UDP 10.1.1.2/9 --> 10.1.1.1/49153)
        TX bitrate: 10.52 kbit/s
        RX bitrate: 10.52 kbit/s
        Mean Delay: 3.69 ms
        Packet Loss Ratio: 0.00 %
sysad@sysad-HP-Elite-Tower-600-G9-Desktop-PC:~/ns-allinone-3.36.1/ns-3.36.1$ cd ..
```





```
Node:0
IP:
            10.1.1.1
            127.0.0.1
IPv6:
            ::1

MAC:
00:00:00:00:00:01

Other Node:1
Other Node IP:10.1.1.2
Other Node MAC:
00:00:00:00:00:02
Info:
```

```
Node:0
IP:
            10.1.1.1
            127.0.0.1
IPv6:
            ::1

MAC:
00:00:00:00:00:00
```

```
Node:1
IP:
            127.0.0.1
            10.1.1.2
IPv6:
            ::1

MAC:
00:00:00:00:00:02

Other Node:0
Other Node IP:10.1.1.1
Other Node MAC:
00:00:00:00:00:01
Info:
```

```
Node:1
IP:
            127.0.0.1
            10.1.1.2
IPv6:
            ::1

MAC:
00:00:00:00:00:00
```

Submission Details

- Write your answers in a single doc/tex file, and submit its PDF named after your IIT Dharwad roll number, which contains all answers (with screenshots). Also, submit your code with `roll_no_part1.cc` and `roll_no_part2.cc`.