

BÁO CÁO BÀI THỰC HÀNH SỐ 1

HỌC PHẦN: HỆ PHÂN TÁN

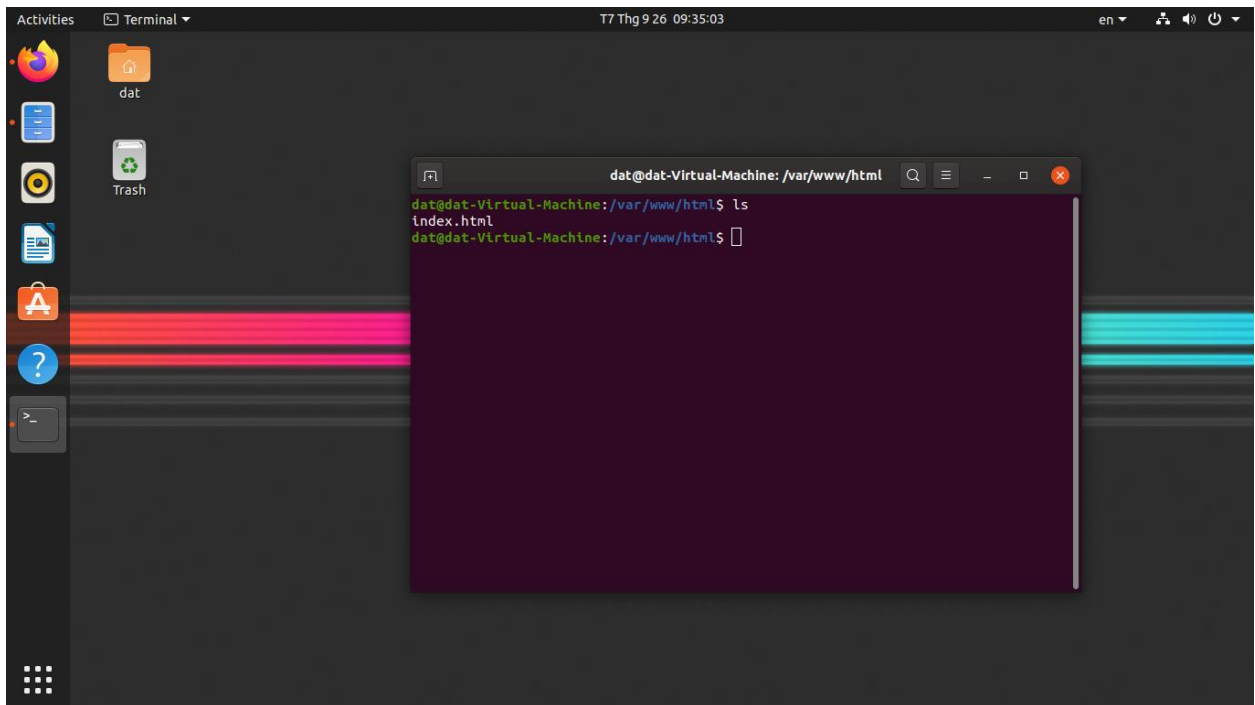
Họ và tên: Phan Thành Đạt

MSSV: 20173001

1. Web server apache2

1.1.Câu hỏi 1

Đường dẫn đến file HTML chứa nội dung mặc định của trang web là:
`/var/www/html/index.html`



1.2.Câu hỏi 2

Cổng mặc định của dịch vụ www là 80 với http và 433 với https

1.3.Câu hỏi số 3

CHMOD là lệnh xác lập thuộc tính cho thư mục hoặc files, ở đây là thư mục `/var/www`.

CHMOD 755 là :

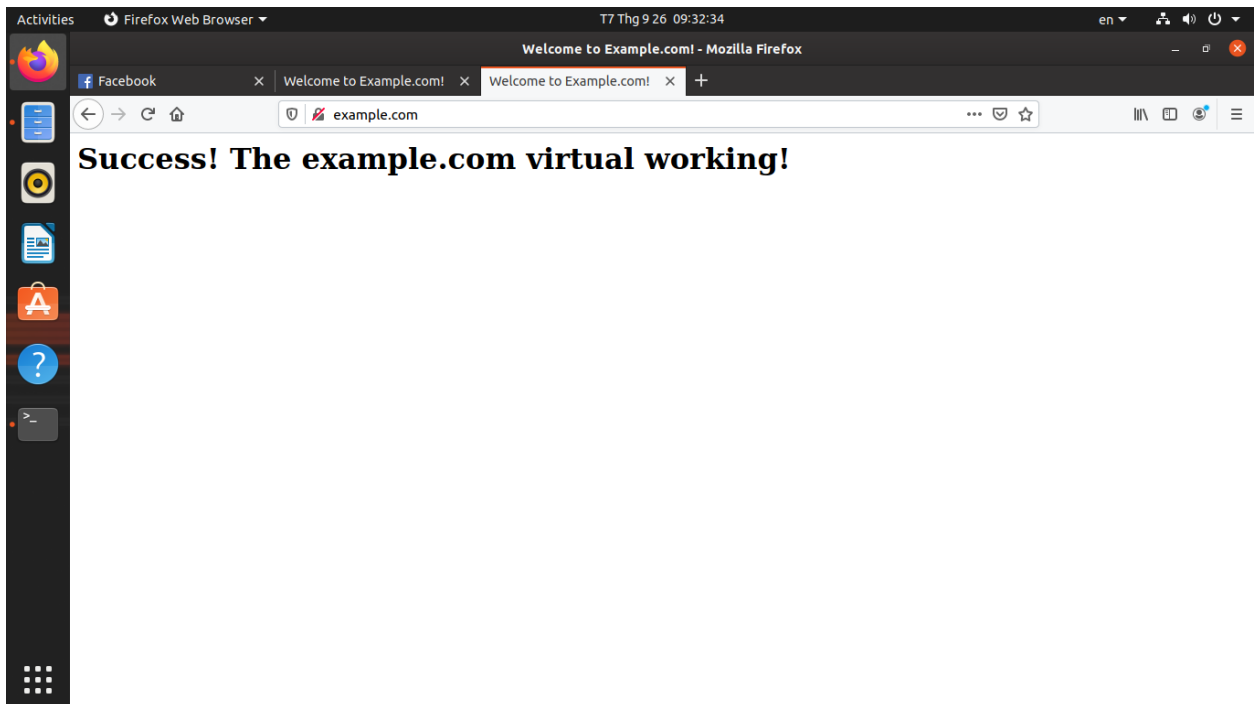
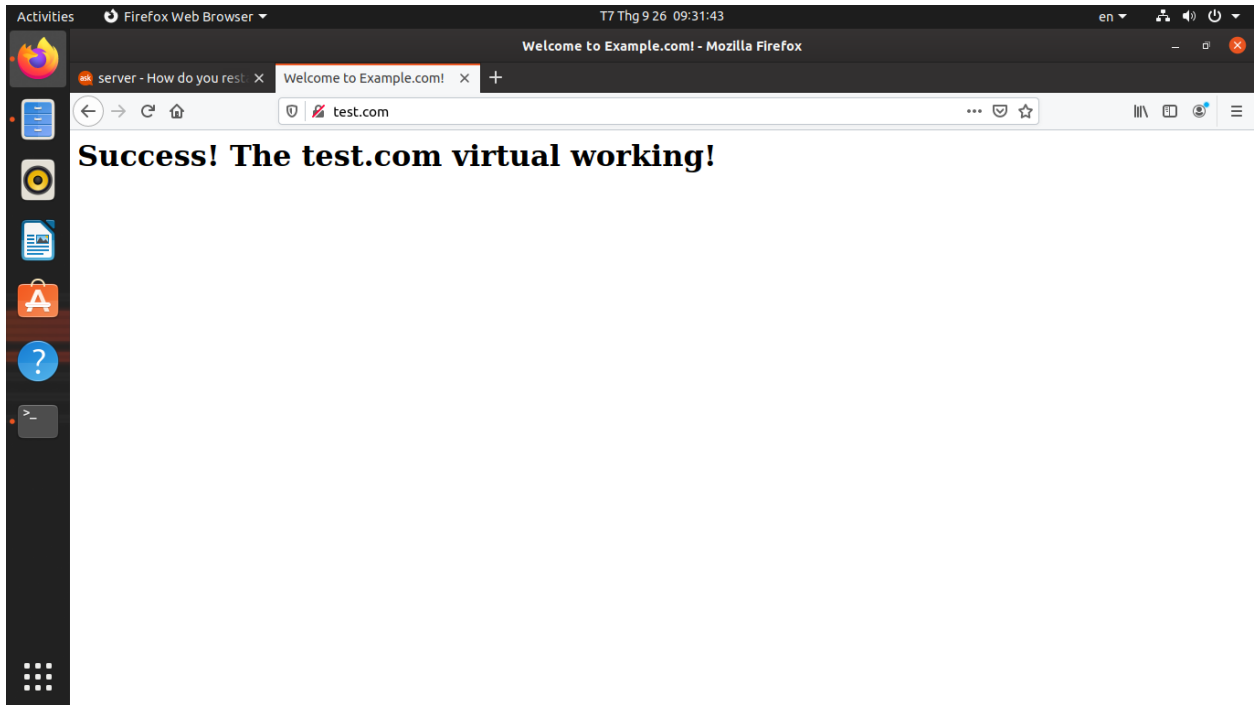
- $7=4+2+1$: dùng để cấp quyền cho Owner full permission (Read,Write,Exe)
- $5=4+0+1$: cấp quyền cho Group chỉ có read và execute (Read,Exe)
- $5=4+0+1$: cấp quyền cho Others cũng chỉ có read và execute (Read,Exe)

1.4.Câu hỏi số 4

Ta sẽ thấy nội dung trong file `index.html` tương ứng với mỗi domain.

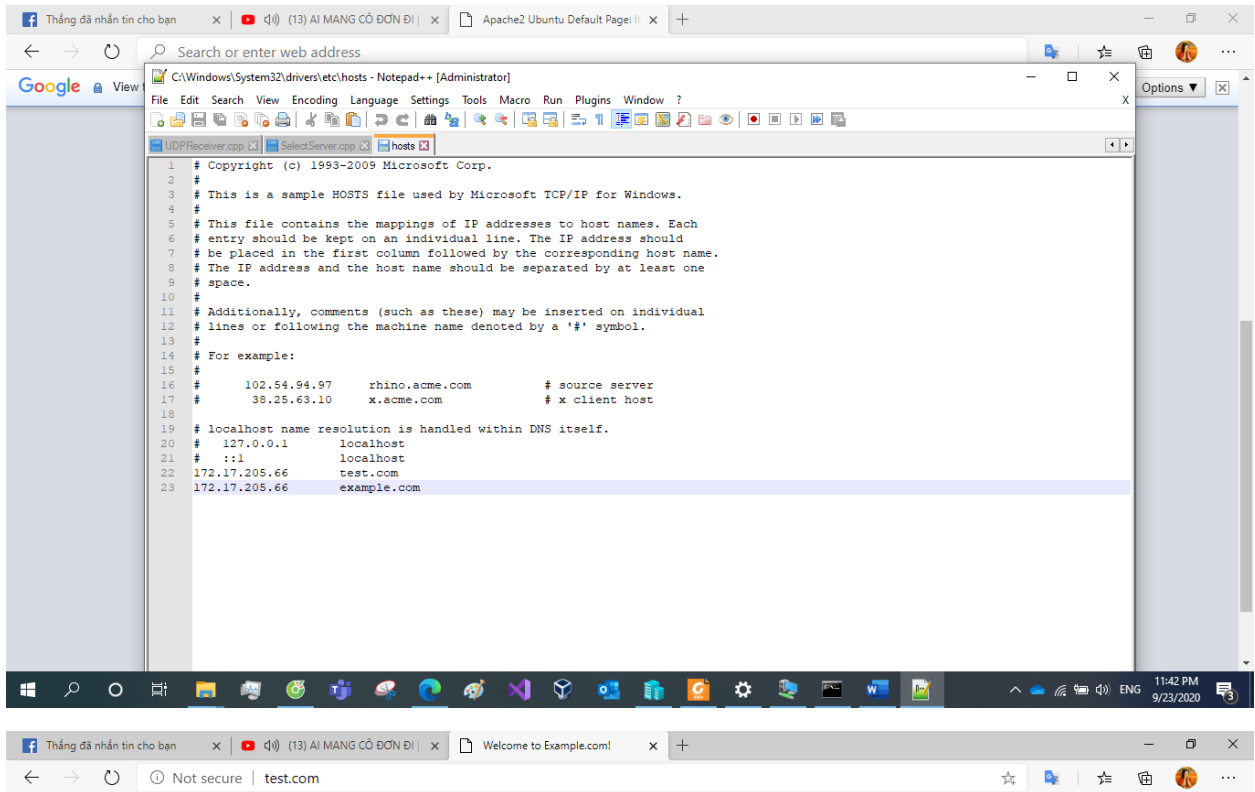
Khi ta gõ `example.com` hoặc `test.com`, ban đầu máy client sẽ thực hiện quá trình phân giải tên miền, client kiểm tra trong file `/etc/hosts` trước và ta đã thực hiện việc này thủ công bằng việc thêm vào file `/etc/hosts` (trong trường hợp ta chưa thêm thì client sẽ thực hiện quá trình phân giải tên miền qua các máy chủ online).

Sau đó client gửi http request lên server apache, nên nhớ rằng apache đang chia sẻ 2 web server trên cùng 1 IP (127.0.0.1) vì thế server sẽ phân tích http header từ client yêu cầu để ánh xạ đến đúng website được chỉ định theo tên miền (việc khai báo này nằm trong file .conf trong thư mục sites-available).



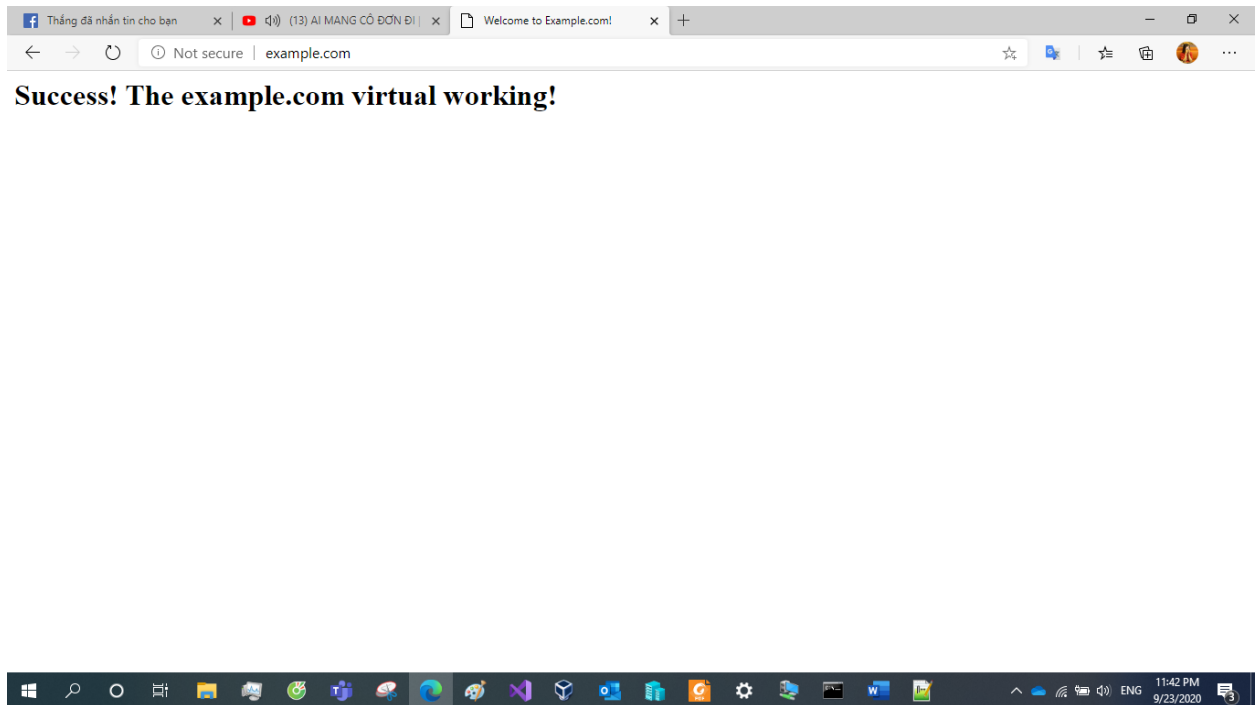
1.5.Câu hỏi số 5

Truy cập từ máy khác cùng mạng LAN vào 2 web đó hoàn toàn được, nếu ta sửa file hosts của máy đó, phân giải tên miền 2 web đó là địa chỉ IP của máy chạy apache.



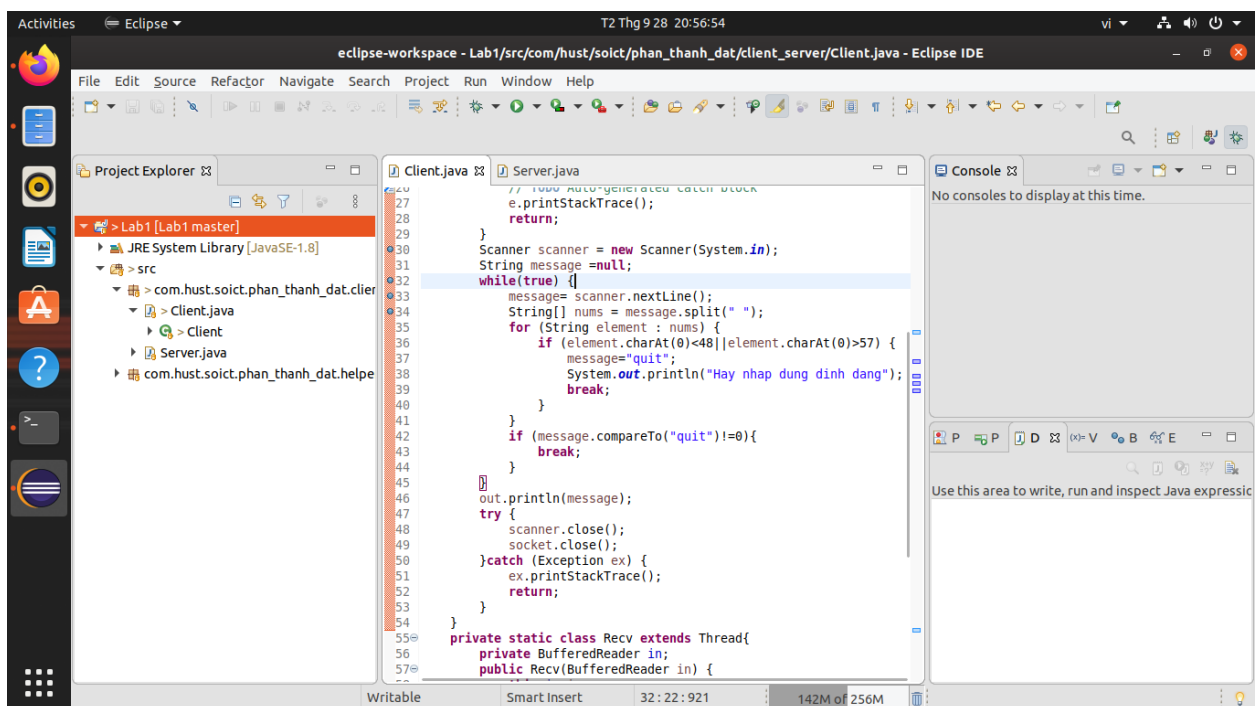
Success! The test.com virtual working!





2. Interface trong Java

2.1. Câu hỏi số 6

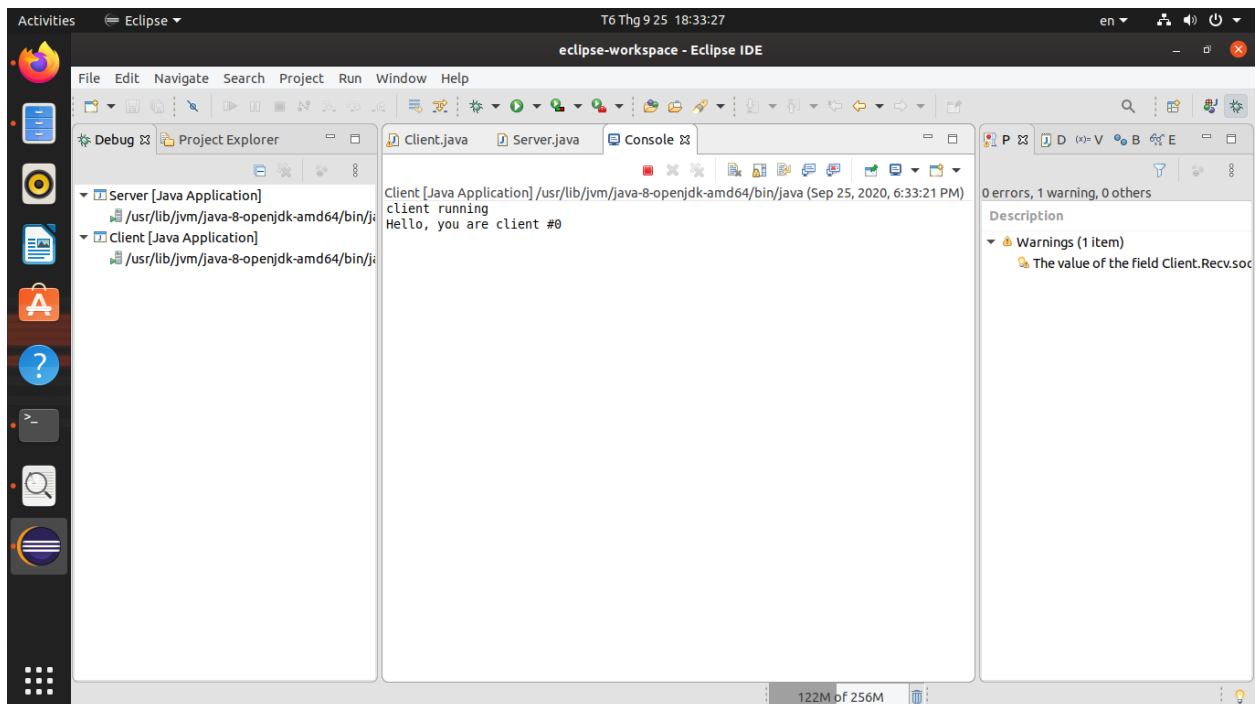
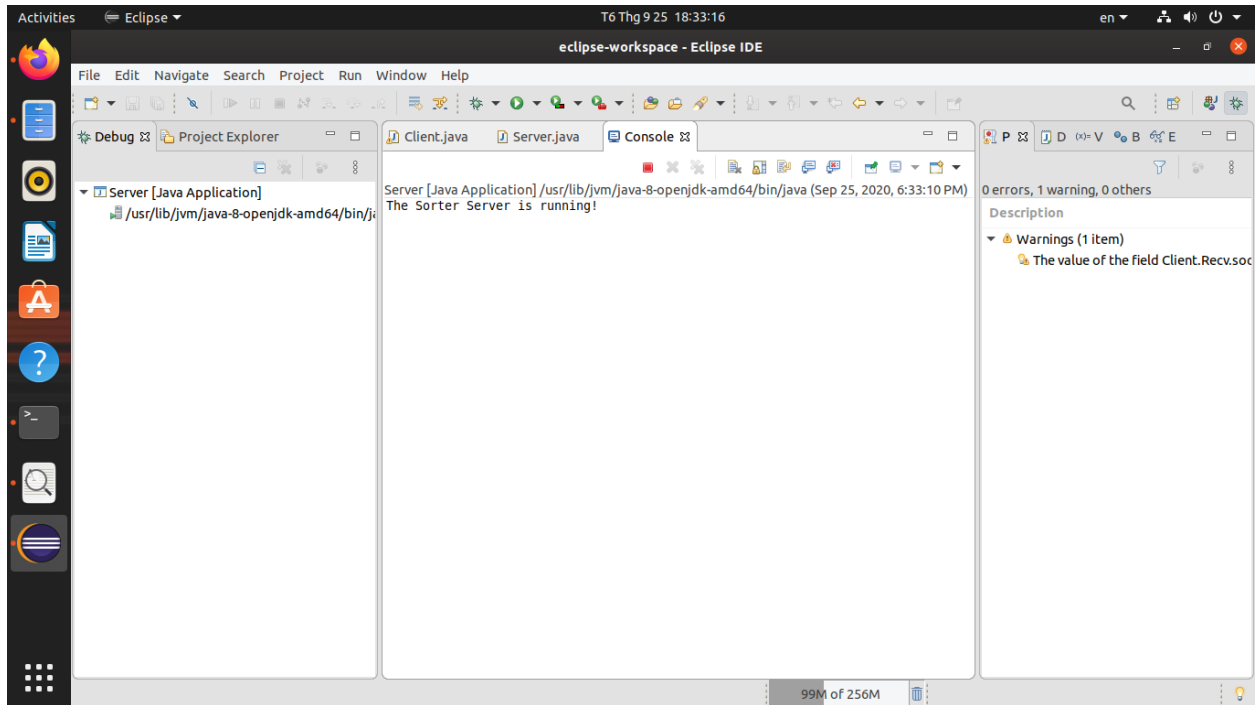


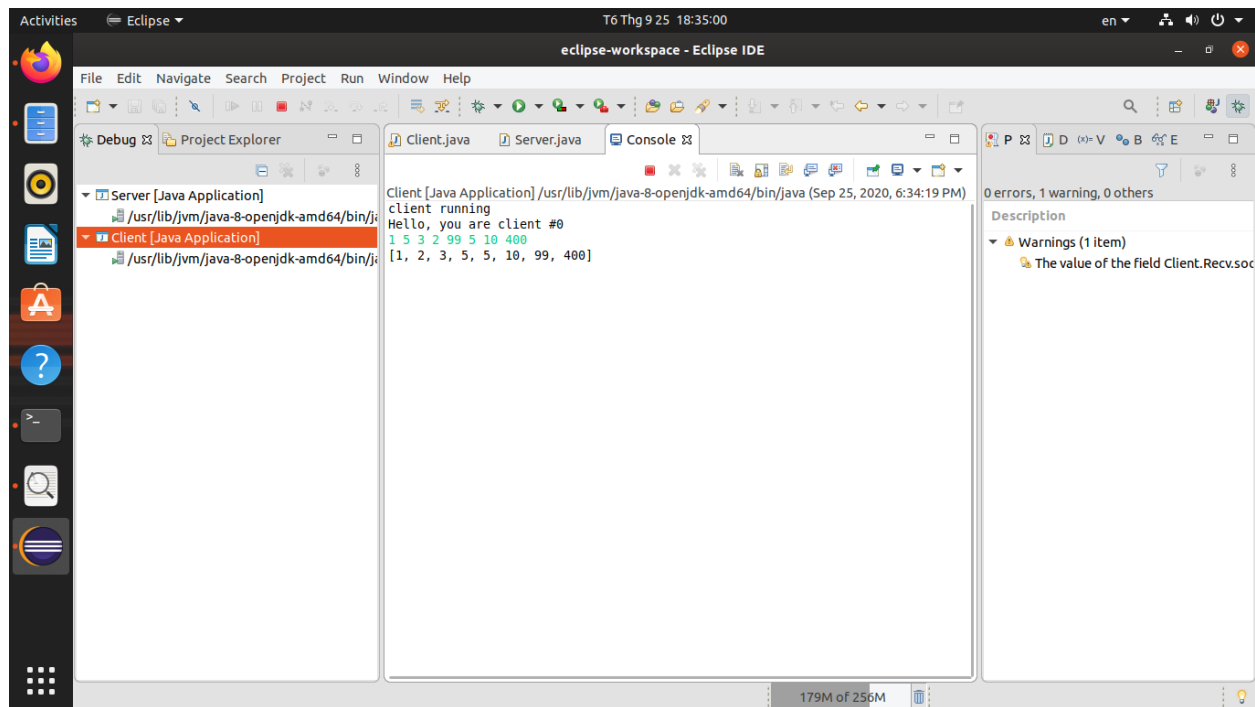
2.2. Câu hỏi số 7

Phương thức **run** thực hiện quá trình nhận dữ liệu từ client, tách dữ liệu thành mảng int, sắp xếp rồi gửi lại cho client.

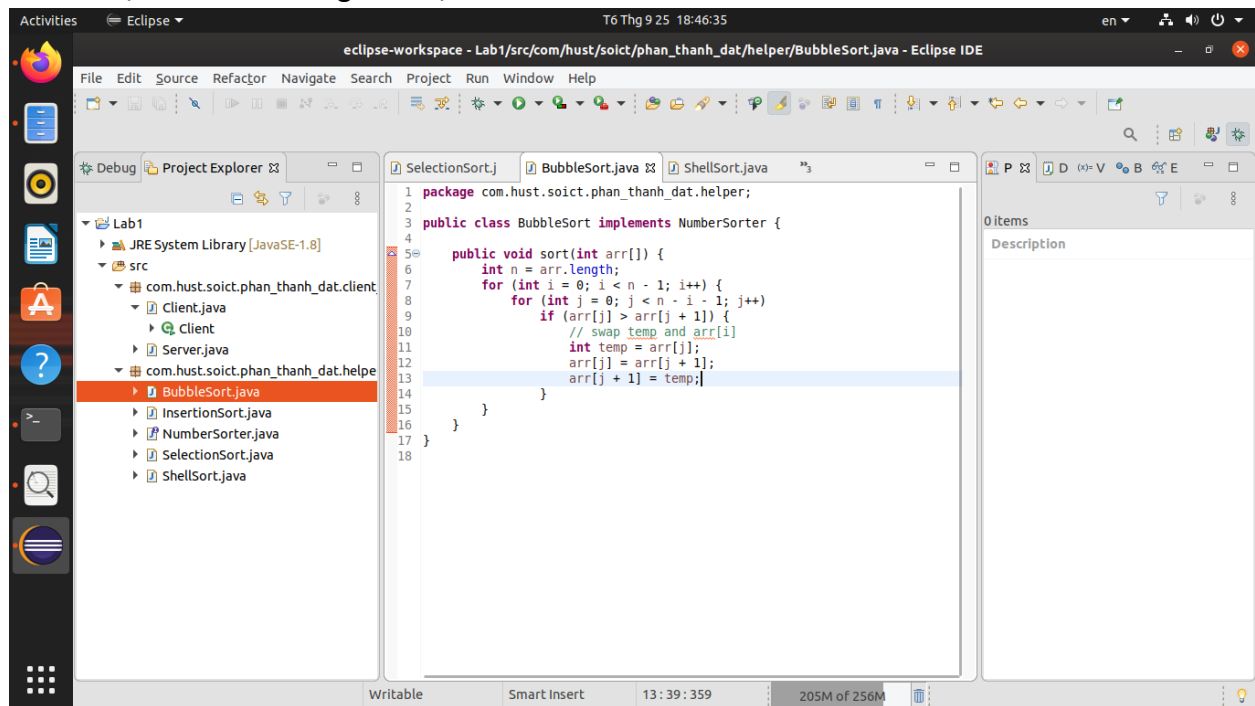
Hàm **run** được thực hiện khi có một kết nối mới tới server (hàm main chạy vòng while chờ client kết nối rồi accept).

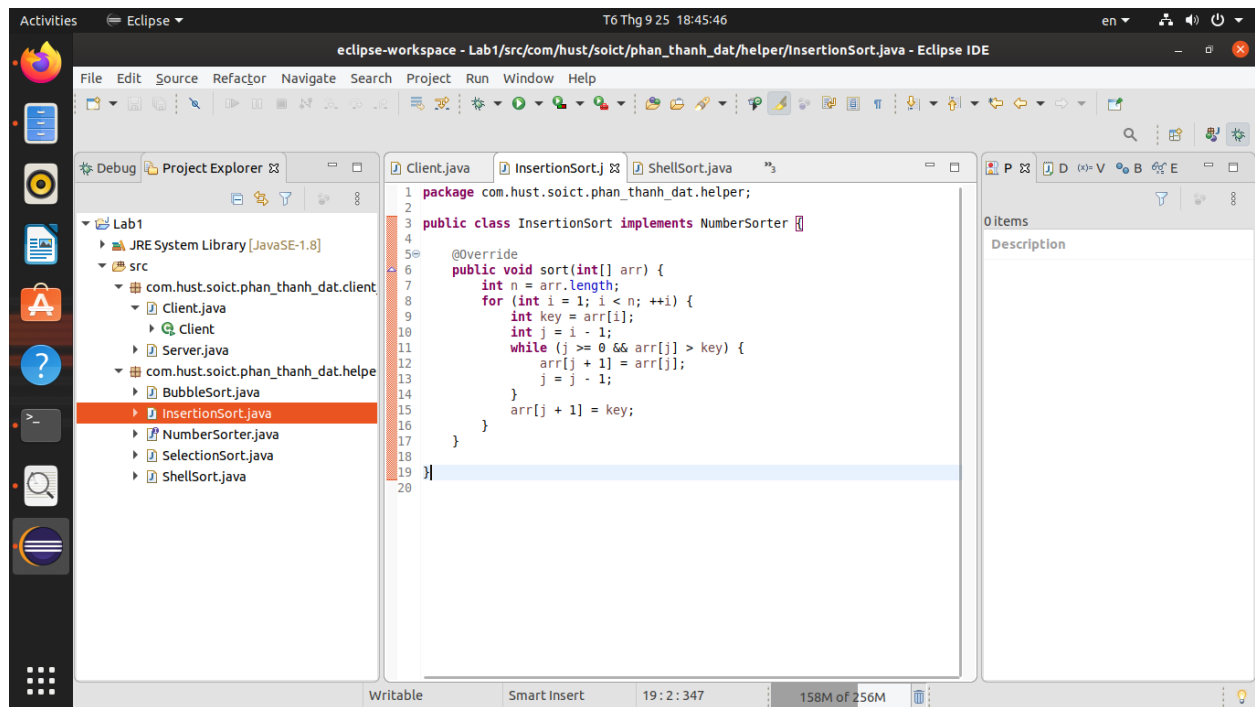
Chạy thử:





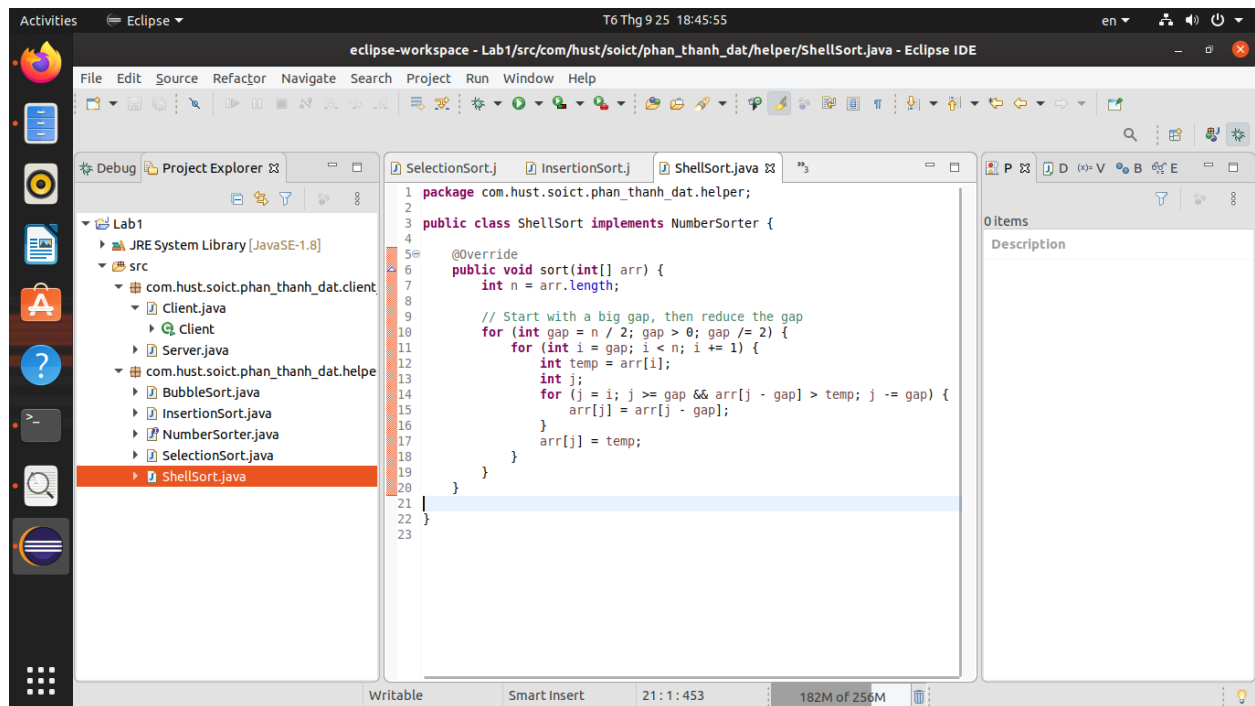
Ta lần lượt triển khai các giải thuật khác nhau:





The screenshot shows the Eclipse IDE interface. The title bar indicates the workspace is 'eclipse-workspace - Lab1/src/com/hust/soict/phan_thanh_dat/helper/InsertionSort.java - Eclipse IDE'. The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations and development. The Project Explorer on the left shows a project structure with 'Lab1' containing 'JRE System Library [JavaSE-1.8]' and 'src'. Under 'src', there are two packages: 'com.hust.soict.phan_thanh_dat.client' and 'com.hust.soict.phan_thanh_dat.helper'. The 'helper' package contains 'BubbleSort.java', 'InsertionSort.java' (selected), 'NumberSorter.java', 'SelectionSort.java', and 'ShellSort.java'. The main editor displays the code for 'InsertionSort.java', which implements the 'NumberSorter' interface. The code defines a 'sort' method that uses a standard insertion sort algorithm. The status bar at the bottom shows 'Writable', 'Smart Insert', '19:2:347', and '158M of 256M'.

```
1 package com.hust.soict.phan_thanh_dat.helper;
2
3 public class InsertionSort implements NumberSorter {
4
5     @Override
6     public void sort(int[] arr) {
7         int n = arr.length;
8         for (int i = 1; i < n; ++i) {
9             int key = arr[i];
10            int j = i - 1;
11            while (j >= 0 && arr[j] > key) {
12                arr[j + 1] = arr[j];
13                j = j - 1;
14            }
15            arr[j + 1] = key;
16        }
17    }
18 }
19
20 }
```



The screenshot shows the Eclipse IDE interface. The title bar indicates the workspace is 'eclipse-workspace - Lab1/src/com/hust/soict/phan_thanh_dat/helper/ShellSort.java - Eclipse IDE'. The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations and development. The Project Explorer on the left shows the same project structure as the first screenshot, with 'ShellSort.java' selected in the 'helper' package. The main editor displays the code for 'ShellSort.java', which implements the 'NumberSorter' interface. The code defines a 'sort' method that uses a Shell sort algorithm, starting with a large gap and reducing it. The status bar at the bottom shows 'Writable', 'Smart Insert', '21:1:453', and '182M of 256M'.

```
1 package com.hust.soict.phan_thanh_dat.helper;
2
3 public class ShellSort implements NumberSorter {
4
5     @Override
6     public void sort(int[] arr) {
7         int n = arr.length;
8
9         // Start with a big gap, then reduce the gap
10        for (int gap = n / 2; gap > 0; gap /= 2) {
11            for (int i = gap; i < n; i += 1) {
12                int temp = arr[i];
13                int j;
14                for (j = i; j >= gap && arr[j - gap] > temp; j -= gap) {
15                    arr[j] = arr[j - gap];
16                }
17                arr[j] = temp;
18            }
19        }
20    }
21 }
22
23 }
```