1. Xây dựng chương trình theo mô hình client/Server như sau:

Server:

* Mở cổng bằng cách nhập từ args.
* Lưu trữ giờ lúc mở cổng
* Nhận thông điệp từ client có dang: MSG+”@@@”+DATA, voi MSG là loại thông điệp, DATA là dữ liệu kèm theo
  + Nếu MSG=”gettime” thì trả cho client thời gian tính bằng giây từ lúc mở cổng server
  + Nếu MSG=”calculate” và DATA là một chuỗi ký tự thì hãy tìm ký tự có tần số xuất hiện lớn nhất và báo về cho client theo đúng định dang: KYTU+TANSO,
  + Nếu MSG=”stop” thì đóng server.

Client:

* Kiểm tra và Kết nối đến server
* Gửi thông điệp đến server và nhận kết quả từ server gửi về, in ra màn hình

1. Xây dựng hệ thống sau:

ServerMaster:

* Mở cổng bằng cách cách nhập từ args
* Quản lý server con
  + Cứ 1 phút nhận cổng mà server con đang mở
  + Nếu như 2 phút trở lên không nhận được thông điệp thì tự động run lại server con
* Nhân thông điệp từ client hỏi về port của server con. Trả lời port cho client

Server con:

* Mở cổng ngẫu nhiên
* Cứ 1 phút gửi cho server master port mà nó đang mở
* Nhận thông điệp từ client có dạng: MSG+”@@@”+DATA, với MSG là loại thông điệp, DATA là dữ liệu kèm theo
  + Nếu MSG=”gettime” thì trả cho client thời gian tính bằng giây từ lúc mở cổng server.
  + Nếu MSG=”calculate” và DATA là một chuỗi ký tự thì hãy tìm ký tự có tần số xuất hiện lớn nhất và báo về cho client theo đúng định dạng: KYTU+TANSO,
  + Nếu MSG=”stop” thì đóng server.

Client:

* Kết nối tới server master và hỏi server con đang mở ở cổng nào
* Kết nối tới server con
* Gửi thông điệp đến server con và nhận kết qua trả về, in ra màn hình.

1. Xây dựng ứng dụng bất kỳ dùng ngôn ngữ JSP và Servlet có các chức năng sau

* Có kết nối CSDL với tính năng thêm, chèn, xóa, sửa.
* Địa chỉ kết nối CSDL được ghi trong file config nào đó để dễ chỉnh sửa
* Nếu như ban đầu không có CSDL thì tự tạo các bảng cho phù hợp
* Có giao diện đẹp mắt