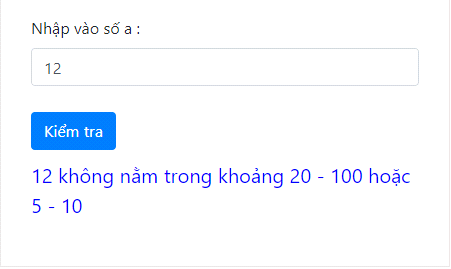
# Bài tập buổi 7

## Bài tập 1: Xây dựng chức năng như hình sau, yêu cầu

Nhập vào 1 số bất kỳ, kiểm tra xem số đó có nằm trong phạm vi từ 1 – 10 hoặc 50 – 100 hay không. Kết quả sẽ được hiển thị như trong hình

Không cần thiết phải trang trí như hình dưới



Yêu cầu: Kiểm tra tính hợp lệ của dữ liệu

* Sử dụng thuộc tính required của HTML5 để kiểm tra
* Vượt qua thuộc tính required bằng cách nhập dấu cách thì sẽ kiểm tra xem trường có để trống hay không, kiểm tra bằng hàm trim()

trim(): Loại bỏ dấu trắng ở đầu và cuối chuỗi

trim(‘abc ‘) => abc

* Kiểm tra xem thông tin nhập vào có phải là số hay không, sử dụng hàm Number.isNaN(number) để kiểm tra

VD: Number.isNaN(‘abc’) => true

Number.isNaN(123) => false

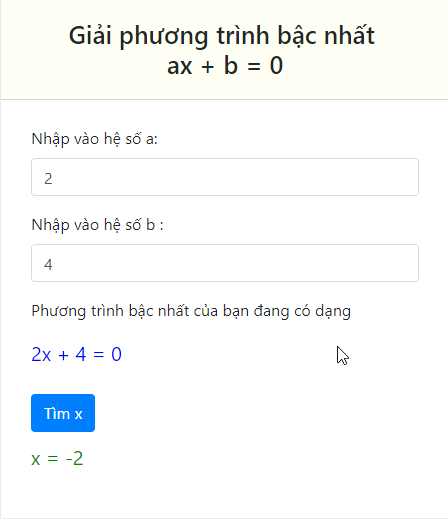
* Hàm ép kiểu từ chuỗi sang số nguyên: parseInt(<chuỗi-cần-ép>)**,** hàm này trả về số nguyên tương ứng sau khi ép kiểu thành công

parseInt(‘123’) => 123

## Bài tập 2: Xây dựng chức năng như hình sau, yêu cầu

Giải phương trình bậc nhất 1 ẩn với giao diện như sau, bắt buộc validate với trường hơp không nhập số và nhập không phải số

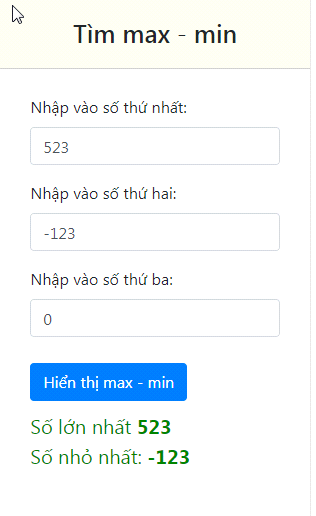
Không cần thiết phải trang trí như hình



Yêu cầu: bắt buộc validate với trường hơp không nhập số và nhập không phải số

## Bài tập 3: Xây dựng giao diện như hình sau, yêu cầu:

* Nhập vào 3 số bất kỳ
* Tìm số lớn nhất, nhỏ nhất trong 3 số vừa nhập, hiển thị thông báo như trong hình

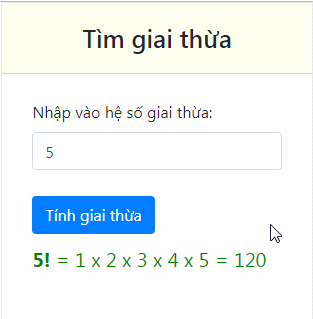


## Bài tập 4: Xây dựng giao diện như sau, yêu cầu:

* Nhập hệ số giai thừa, tính giai thừa của hệ số đó

Tips: Giai thừa của n là tích của n số nguyên dương đầu tiên, ký hiệu viết tắt n!

VD: 5! = 1 \* 2 \* 3 \* 4 \* 5 = 120



## Bài tập 5: Cho giao diện đầy đủ như sau



Yêu cầu:

Mô tả giao diện:

* Tất cả các input ở dạng nhập text bắt buộc phải có type=text
* Phần select sẽ có các thông tin như sau

Giá trị -1, text Select-Country

Giá trị 1, text Viet Nam

Giá trị 2, text Japan

Giá trị 3, text USA

Bắt validate sử dụng Javascipt, với các yêu cầu sau

* Các trường name, email không được để trống, nếu để trống cần báo lỗi màu đỏ “Không được để trống trường này”, đồng thời focus con trỏ chuột vào ô input tương ứng đó
* Nếu không chọn Country, báo lỗi “Cần phải chọn Country”, và focus con trỏ chuột vào vị trí của select đó
* Nếu nhập sai định dạng email, cần báo lỗi “Email chưa đúng định dạng”, điều kiện check email hợp lệ là
  + Trước dấu ‘@’ được phép nhập các chữ hoa, thường, số, dấu gạch ngang và dấu được xuất hiện nhiều lần. Dấu gạch ngang (\_) và dấu (.) không được phép xuất hiện cạnh nhau, tức là không có \_\_ và ..
  + Trước dấu “@” không được có dấu chấm. Ví dụ [abc.@insta.com](mailto:abc.@insta.com) là không hợp lệ
  + Từ sau dấu “@” đến trước dấu chấm được phép nhập các chữ hoa, thường, số với số lượng kí tự >=1
  + Cuối cùng phần đuôi sau dấu chấm – ví dụ “.com” chỉ được phép nhập từ

2->3 kí tự

* Nếu có dấu . của subdomain, thì subdomain thứ 2 chỉ cho phép có 2 kí tự (có thể viết hoa, thường, số): VD: [abc@asdas.**co**.in](mailto:abc@asdas.co.in) (Bôi đen)
* Nếu có domain cuối thì chỉ cho phép nhập chữ thường (a-z) xuất hiện từ 2-3 lần. Ví dụ: [abc@asdas.co.](mailto:abc@asdas.co.in)**[in](mailto:abc@asdas.co.in)** (bôi đen)
  + Để kiểm tra regex có hợp lệ hay không, kiểm tra ở đây. Sau đó chép regex này vào khu vực khai báo regex để kiểm tra. Trong này đã có regex hợp lệ và không hợp lệ <https://regex101.com/r/IPJE9Y/6>
  + Khi nhập các thông tin hợp lệ, thì hiển thị các thông tin user vừa nhập, cách nhau bằng môt dấu xuống dòng