// Câu 1

const LongestIncrease = (arr) => {

// Gán mặc định dãy dài nhất là 1 và vị trí bắt đầu là 0 (đầu dãy)

let resultMax = 1;

let resultIndex = 0;

for(let i = 0; i < arr.length - 1; i++){

// Khởi tạo biến đếm với từng phần tử của arr và lưu index của phần tử đó

let count = 1;

let index = i;

// Chạy vòng for để đếm

for(let j = i + 1; j < arr.length; j++){

// Nếu phần tử phía sau tiếp tục tăng

if(arr[j] >= arr[j-1] )

{

// Tăng biến đếm

count++;

// Xử lý trường hợp phần tử tăng và là phần tử cuối của arr

if(j === (arr.length - 1) && count > resultMax){

resultMax = count;

resultIndex = index;

}

}

// Trường hợp phần tử không tăng

else if(arr[j] < arr[j-1] )

{

// Nếu biến đếm cao hơn biến đếm của result thì đổi biến đếm và index

if(count > resultMax){

resultMax = count;

resultIndex = index;

}

// Break vòng for để tiếp tục với i khác

break;

}

}

}

return [resultMax,resultIndex];

}

// Câu 2

const CompareString = (strA,strB) => {

// Lấy độ dài của chuỗi ngắn hơn

let minLength = (strA.length < strB.length)?strA.length:strB.length;

// Mặc định count giống nhau = 0

let count = 0;

// Vị trí để bắt đầu tìm ở strB, mặc định = 0

let tempIndex = 0;

// Chạy vòng for qua từng kí tự của strA

for(let i = 0; i < strA.length; i++){

// Chạy vòng for từ tempIndex ở strB

for(let j = tempIndex; j < strB.length; j++ ){

// Nếu kí tự giống nhau

if(strB[j] === strA[i]){

// Tăng biến đếm

count++;

// Gán lại vị trí bắt đầu tìm ở string B

tempIndex = j + 1;

// Bắt đầu với i mới ở str A

break;

}

}

}

// Trả về tỉ lệ giống nhau, chuyển thành số nguyên

return parseInt((count/minLength)\*100);

}

// Câu 3

const E = (str,number) => {

// Array mã hóa

let tempStr = [];

let charCode = null;

for(let i = 0; i < str.length; i++){

// Tìm char code của kí tự

charCode = str.charCodeAt(i);

// Tăng char code lên theo number

charCode += number;

// Push kí tự mới vào Array

tempStr.push(String.fromCharCode(charCode));

}

// Trả về chuỗi

return tempStr.join("");

}

const D = (str,number) => {

// Array giải mã

let tempStr = [];

// Cho N = 11

let tempNumber = 11 \* number + 1;

let charCode = null;

for(let i = 0; i < str.length; i++){

// Tìm char code của kí tự

charCode = str.charCodeAt(i);

// Giảm char code để trở về char code ban đầu

charCode -= tempNumber;

// Push kí tự mới vào array

tempStr.push(String.fromCharCode(charCode));

}

// Trả về chuỗi mã hóa

return tempStr.join("");

}