Trả lời câu hỏi Java

Câu 1

**public** **static** **void** main(String[] args) {

*cau1*("chien");

}

**private** **static** **void** cau1(String str1) {

String str2="";

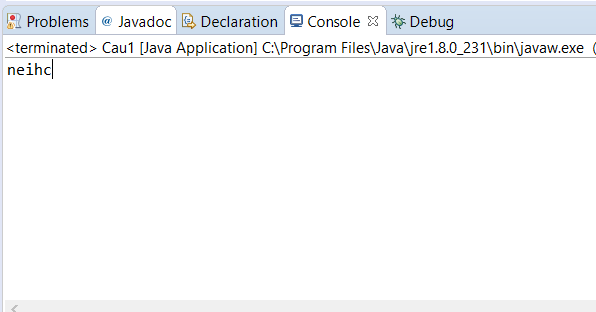
**for** (**int** i = str1.length()-1; i >=0; i--) {

str2 = str2 + str1.charAt(i);

}

System.***out***.println(str2);

}



Câu 2:

**public** **static** **void** main(String[] args) {

System.***out***.println(*cau2*("chien"));

System.***out***.println(*cau2*("minh chien"));

}

**public** **static** **boolean** cau2(String str) {

Set<Character> result = **new** HashSet<>();

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

Character word = str.charAt(i);

**if** (!result.add(word)) {

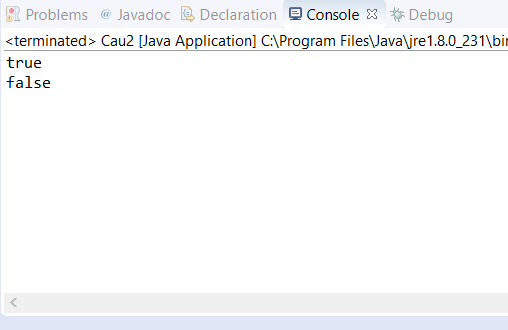
**return** **false** ;

}

}

**return** **true**;

}



Câu 3 :

**public** **static** **void** main(String[] args) {

System.***out***.println(*cau3*("minh chien", "chienminh"));

}

**public** **static** **boolean** cau3(String input1 , String input2) {

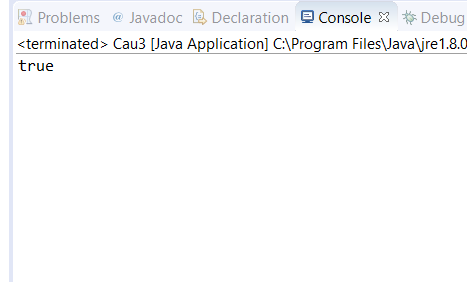
**if** ((input1+ input1).contains(input2)) {

**return** **true**;

}

**return** **false**;

}



Câu 4

**public** **static** **void** main(String[] args) {

*cau4*("dang minh chien");

}

**public** **static** **void** cau4(String str) {

Map<Character, Integer> result = **new** HashMap<>();

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

**char** word = str.charAt(i);

**if** (result.containsKey(word)) {

result.put(word, Integer.*parseInt*(result.get(word).toString())+1);

}

**else** {

result.put(word , 1);

}

}

**for** (Object x : result.keySet()) {

Integer number = Integer.*parseInt*(result.get(x).toString());

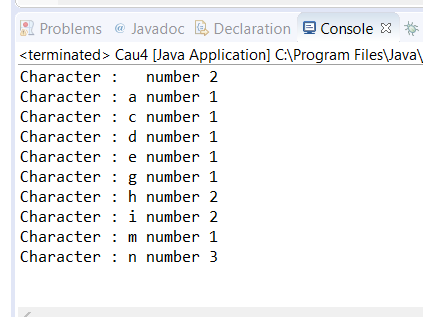
**if** (number>=1) {

System.***out***.println("Character : "+ x +" number "+ number );

}

}

}



Câu 5:

**public** **static** **void** main(String[] args) {

System.***out***.println(*cau5*("ninhchien"));

}

**public** **static** Character cau5(String str) {

Map<Character, Integer> countCharacter = **new** LinkedHashMap<>();

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

Character c = str.charAt(i);

**if** (!countCharacter.containsKey(c)) {

countCharacter.put(c, 1);

} **else** {

countCharacter.put(c, countCharacter.get(c) + 1);

}

}

**for** (Entry<Character, Integer> e : countCharacter.entrySet()) {

**if** (e.getValue() == 1) {

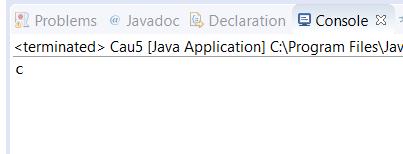
**return** e.getKey();

}

}

**return** **null**;

}



Câu 6:Chưa làm được

Câu 7 :

String chuoi = ‘lap trinh java web’

chuoi.lenght();

Câu 8

**static** **void** cau8(String str, String ans)

{

**if** (str.length() == 0) {

System.***out***.print(ans + " ");

**return**;

}

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

**char** ch = str.charAt(i);

String ros = str.substring(0, i) +

str.substring(i + 1);

*cau8*(ros, ans + ch);

}

}

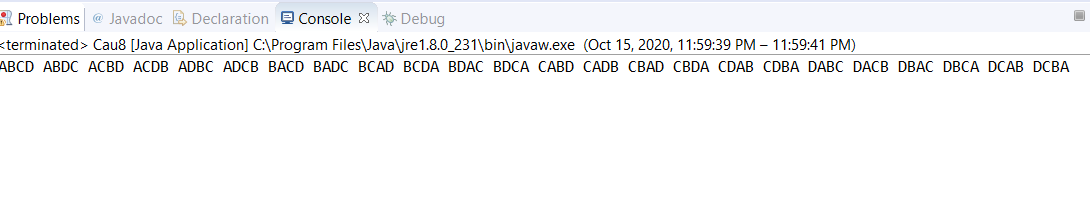
**public** **static** **void** main(String[] args)

{

String s = "ABCD";

*cau8*(s, "");

}



Câu 9 :

**public** **class** Cau9 {

**public** **static** **int** nhap() {

Scanner input = **new** Scanner(System.***in***);

**boolean** check = **false**;

**int** n = 0;

**while** (!check) {

System.***out***.print(" ");

**try** {

n = input.nextInt();

check = **true**;

} **catch** (Exception e) {

System.***out***.println("Ban phai nhap so! hay nhap lai...");

input.nextLine();

}

}

**return** (n);

}

**public** **static** **int** tinhTong(**long** i) {

**int** sum = 0;

**long** n;

**while** (i != 0) {

n = i % 10;

sum += n;

i /= 10;

}

**return** (sum);

}

**public** **static** **void** main(String[] args) {

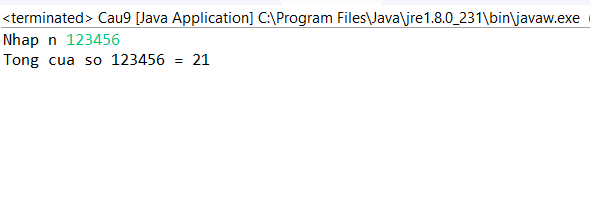
System.***out***.print("Nhap n");

**int** n = *nhap*();

System.***out***.println("Tong cua so " + n + " = " + *tinhTong*(n));

}

}



Câu 10

Giống câu 4

Câu 11

**public** **static** **void** main(String[] args) {

*cau11*();

}

**public** **static** **void** cau11() {

**int** numbers[] = **new** **int**[] { 23, 55, 66, 14, 24, 33, 99 };

**int** min = numbers[0];

**int** max = numbers[0];

**for** (**int** i = 0; i < numbers.length; i++) {

**if** (numbers[i] < min) {

min = numbers[0];

}

**if** (numbers[i] > max) {

max = numbers[i];

}

}

System.***out***.println("Min" + min);

System.***out***.println("Max" + max);

}

Câu 12

**public** **static** **void** main(String[] args) {

*cau12*();

}

**public** **static** **void** cau12() {

**int**[] numbers = { 11, 6, 5, 7, 0, 1 };

Arrays.*sort*(numbers);

HashSet<Integer> set = **new** HashSet<>();

**for** (**int** i = numbers[0]; i < numbers[numbers.length - 1]; i++) {

set.add(i);

}

**for** (**int** i = 0; i < numbers.length; i++) {

set.remove(numbers[i]);

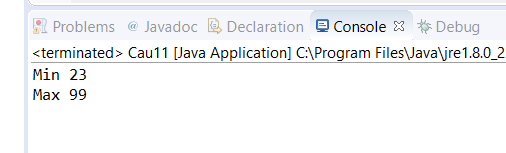
}

**for** (**int** x : set) {

System.***out***.print(x + " ");

}

}



Câu 13

Sắp xếp mảng lớn đến nhỏ rồi duyệt mảng vào Set và lấy ra phần tử thứ 2

Câu 14

**static** **int** cau14(**int** arr[], **int** arr\_size) {

**int** i;

**for** (i = 0; i < arr\_size; i++) {

**int** count = 0;

**for** (**int** j = 0; j < arr\_size; j++) {

**if** (arr[i] == arr[j])

count++;

}

**if** (count % 2 != 0)

**return** arr[i];

}

**return** -1;

}

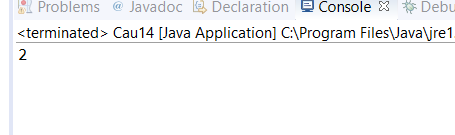
**public** **static** **void** main(String[] args) {

**int** arr[] = **new** **int**[] { 2, 3, 3, 4, 3, 2, 4, 3, 3, 2, 4, 4 };

**int** n = arr.length;

System.***out***.println(*cau14*(arr, n));

}



Câu 15

**static** **void** cau15(**int** arr[], **int** sum)

{

HashSet<Integer> s = **new** HashSet<Integer>();

**for** (**int** i = 0; i < arr.length; ++i) {

**int** temp = sum - arr[i];

**if** (s.contains(temp) ) {

System.***out***.println(

"Phần tử đã liên tiếp có tổng "

+ sum + " là(" + arr[i]

+ ", " + temp + ")");

}

s.add(arr[i]);

}

}

// Driver Code

**public** **static** **void** main(String[] args)

{

**int** A[] = { 1, 4, 45, 6, 10, 8 };

**int** n = 49;

*cau15*(A, n);

}

Câu 16

**static** **void** cau16(**int** arr1[], **int** arr2[], **int** n, **int** m) {

**for** (**int** i = 0; i < n; i++)

**for** (**int** j = 0; j < m; j++)

**if** (arr1[i] == arr2[j]) {

System.***out***.println("Số giống nhau giữa hai mảng là : " + arr1[i] );

}

}

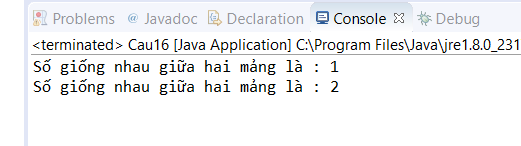
**public** **static** **void** main(String[] args) {

**int**[] a = {1,2,3,4,5,6};

**int**[] b = {1,2,8,9,10,11};

*cau16*(a,b ,6 , 6);

}



Cau 17

**static** **int** cau17(**int** arr[], **int** n) {

**if** (n==0 || n==1)

**return** n;

**int**[] temp = **new** **int**[n];

**int** j = 0;

**for** (**int** i=0; i<n-1; i++)

**if** (arr[i] != arr[i+1])

temp[j++] = arr[i];

temp[j++] = arr[n-1];

**for** (**int** i=0; i<j; i++)

arr[i] = temp[i];

**return** j;

}

**public** **static** **void** main (String[] args)

{

**int** arr[] = {1, 2, 2, 3, 4, 4, 4, 5, 5};

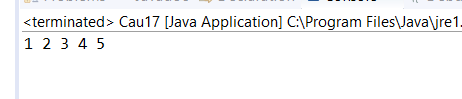
**int** n = arr.length;

n = *cau17*(arr, n);

**for** (**int** i=0; i<n; i++)

System.***out***.print(arr[i]+" ");

}



Cau 18

**static** **void** cau18(**int** inputArray[], **int** inputNumber)

{

System.***out***.println("Các cặp phần tử có tổng bằng "+inputNumber+" là : ");

**for** (**int** i = 0; i < inputArray.length; i++)

{

**for** (**int** j = i+1; j < inputArray.length; j++)

{

**if**(inputArray[i]+inputArray[j] == inputNumber)

{

System.***out***.println(inputArray[i]+" + "+inputArray[j]+" = "+inputNumber);

}

}

}

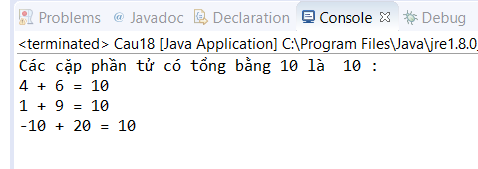
}

**public** **static** **void** main(String[] args)

{

*cau18*(**new** **int**[] {4, 1, 5, -10, 9,6, 20}, 10);

}



Cau 19

**public** **static** **void** main(String[] args) {

**int**[] inp = {12,17,70,15,22,65,21,90 };

**int** n = inp.length;

**int**[] a = **new** **int**[(n + 1)/2];

**int**[] b = **new** **int**[n - a.length];

**for** (**int** i = 0; i < n; i++)

{

**if** (i < a.length)

a[i] = inp[i];

**else**

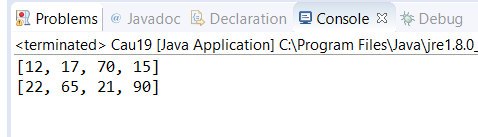
b[i - a.length] = inp[i];

}

System.***out***.println(Arrays.*toString*(a));

System.***out***.println(Arrays.*toString*(b));

}



Cau 20

Cau 21

Câu 22

Câu 23

**private** **int** cau23(**int** arr[], **int** n)

{

**int** min = arr[0];

**for** (**int** i = 1; i < n; i++)

**if** (arr[i] < min)

min = arr[i];

**return** min;

}

**private** **int** getMaximum(**int** arr[], **int** n)

{

**int** max = arr[0];

**for** (**int** i = 1; i < n; i++)

**if** (arr[i] > max)

max = arr[i];

**return** max;

}

**public** **static** **boolean** checkArrayContainsConsecutiveElements(**int** arr[], **int** n)

{

**if** ( n < 1 )

**return** **false**;

**int** min = getMinimum(arr, n);

**int** max = getMaximum(arr, n);

**if** (max - min + 1 == n)

{

**boolean**[] visited=**new** **boolean**[arr.length];

**for** (**int** i = 0; i < n; i++)

{

**if** ( visited[arr[i] - min] != **false** )

**return** **false**;

visited[arr[i] - min] = **true**;

}

**return** **true**;

}

**return** **false**;

}

**public** **static** **void** main(String args[])

{

**int** arr[]= {6, 7, 5, 6};

**if**(checkArrayContainsConsecutiveElements(arr, arr.length))

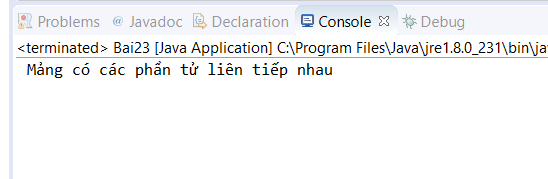
System.***out***.println(" Mảng có các phẩn tử liên tiếp nhau ");

**else**

System.***out***.println(" Mảng không có các phẩn tử liên tiếp nhau");

**return**;

}



Câu 24

Câu 25

Câu 26