**Interview Question**

**Question 1**

**Find missing element in a sequence**

**public** **class** test1 {

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

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

**int** n = arr[arr.length - 1];

**int** expectedSum = n \* (n + 1) / 2;

**int** actualSum = Arrays.*stream*(arr).sum();

**int** missingElement = expectedSum - actualSum;

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

}

}

**Question 2**

**Find longest sequence in a array**

//1,2,4,6,12,10,9,8,7

**public** **class** test1 {

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

Scanner s1 = **new** Scanner(System.***in***);

**int** n = s1.nextInt();

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

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

arr[i] = s1.nextInt();

}

HashMap<Integer, Boolean> hm = **new** HashMap<>();

**for** (**int** val : arr) {

hm.put(val, **true**);

}

**for** (**int** val : arr) {

**if** (hm.containsKey(val - 1)) {

hm.put(val, **false**);

}

}

**int** ml = 0; // max length

**int** msp = 0; // max starting point

**for** (**int** val : arr) {

**if** (hm.get(val) == **true**) {

**int** tl = 1; // temporary length

**int** tsp = val; //temporary starting point

**while**(hm.containsKey(tsp+tl)) {

tl++;

}

**if**(tl>ml) {

ml=tl;

msp=tsp;

}

}

}

**for**(**int** i=0;i<ml;i++) {

System.***out***.println(msp+i);

}

}

}

**// output**

6

7

8

9

10

**Question 3**

**Matrix with sorted element . Get position of any specific element**

// 3 3

//input 1 2 3 4 5 6 7 8 9 and element to search 7

//output 2 0

**public** **class** test1 {

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

Scanner s1 = **new** Scanner(System.***in***);

**int** r = s1.nextInt();

**int** c = s1.nextInt();

**int**[][] arr = **new** **int**[r][c];

**for** (**int** i = 0; i < r; i++) {

**for** (**int** j = 0; j < c; j++) {

arr[i][j] = s1.nextInt();

}

}

**int** element = s1.nextInt();

**int** i = 0;

**int** j = arr[0].length - 1;

**while** (i < arr.length && j >= 0) {

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

System.***out***.println(i + " " + j);

**return**;

} **else** **if** (element < arr[i][j]) {

j--;

} **else** {

i++;

}

}

}

}

**Question 4**

**Reverse an array.**

//input 1,2,3,4,5

//output 5,4,3,2,1

**public** **class** test1 {

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

Scanner s1 = **new** Scanner(System.***in***);

**int** n = s1.nextInt();

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

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

arr[i] = s1.nextInt();

}

*reverse*(arr);

*display*(arr);

}

**private** **static** **void** display(**int**[] arr) {

**for** (**int** val : arr) {

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

}

}

**private** **static** **void** reverse(**int**[] arr) {

**int** i = 0;

**int** j = arr.length - 1;

**while** (i < j) {

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

arr[i] = arr[j];

arr[j] = temp;

i++;

j--;

}

}

}

**Question 5**

**sort an array in ascending order**

//input 1,2,3,4,5

//output 5,4,3,2,1

**public** **class** test1 {

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

Scanner s1 = **new** Scanner(System.***in***);

**int** n = s1.nextInt();

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

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

arr[i] = s1.nextInt();

}

*sort*(arr);

*display*(arr);

}

**private** **static** **void** display(**int**[] arr) {

**for** (**int** val : arr) {

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

}

}

**private** **static** **void** sort(**int**[] arr) {

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

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

**if** (arr[i] > arr[j]) {

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

arr[i] = arr[j];

arr[j] = temp;

}

}

}

}

}

**Question 6**

**Occurrence of each character in a string. Find maximum occurrence**

**public** **class** test1 {

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

String str = "saurabh";

Character key = str.chars().mapToObj(ch -> (**char**) ch)

.collect(Collectors.*groupingBy*(x -> x, Collectors.*counting*())).entrySet().stream()

.max(Map.Entry.*comparingByValue*()).get().getKey();

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

}

}

**Using java 7**

**public** **class** test1 {

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

String str = "saurabh";

**char**[] ch = str.toCharArray();

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

**for** (**char** val : ch) {

**if** (hm.containsKey(val)) {

hm.put(val, hm.get(val) + 1);

} **else** {

hm.put(val, 1);

}

}

Set<Entry<Character, Integer>> entrySet = hm.entrySet();

**int** maxCount = 0;

**char** maxChar = 0;

**for** (Entry<Character, Integer> entry : entrySet) {

**if** (entry.getValue() > maxCount) {

maxCount = entry.getValue();

maxChar = entry.getKey();

}

}

System.***out***.println(maxCount + " " + maxChar);

}

}

**Question 8**

**Search an element in an array**

**public** **class** test1 {

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

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

**int** searchElement = 3;

**boolean** anyMatch = Arrays.*stream*(arr).anyMatch(x -> x == searchElement);

System.***out***.println(searchElement + "found in array " + anyMatch);

}

}

**Question 9**

**Permutation of a string**

**public** **class** test1 {

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

String str = "abc";

*printPermutation*(str, "");

}

**private** **static** **void** printPermutation(String ques, String asf) {

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

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

**return**;

}

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

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

String qlpart = ques.substring(0, i);

String qrpart = ques.substring(i + 1);

String rques = qlpart + qrpart;

*printPermutation*(rques, asf + ch);

}

}

}