String Coding

Question 1) How to reverse a String?

1. Using StringBuffer class (To create a stringBuffer object)

Package stringreverse;

Public class StringReverse {

Public static void main(String[] args){

String s = “Selenium”;

StringBuffer sb = new StringBuffer(s); // s= selenium

System.out.println(sb.reverse ());

}

}

Array Coding

1. **There are 2 int type array data type. One is containing 50 elements, and another one is containing 30 elements. Can we assign the array of 50 elements to an array of 30 elements?**

Yes we can assign provided they should the same type. The compiler will check the only type of the array, not the size.

|  |  |
| --- | --- |
| 1  2  3  4  5 | int[] ab = new int[30];  int[] cd = new int[50];  a = b; //Compiler checks only type, not the size |

1. **What is the meaning of anonymous array? Explain with an example?**

Anonymous array means array without any reference.

Example

|  |  |
| --- | --- |
| 1  2  3  4  5 | //Creating anonymous arrays  System.out.println(new int[]{11, 12, 13, 14, 15}.length); //Output : 5  System.out.println(new int[]{31, 94, 75, 64, 41}[1]); //Output : 94 }} |

**3) How to sort an Array?**

It is possible using the built static method that is Arrays. sort ().

|  |  |
| --- | --- |
| 1  2  3  4  5 | Int Marks [] = {12, 5, 7, 9};  Arrays.sort(Marks);  System.out.println(Arrays.toString(Marks));//[5, 7, 9, 12] |

**4): How to get maximum or minimum numbers from an array ?**

**Ans: Example of maximum and minimum:**

int a[]={4,76,4,8,9};

int max=a[0];

for(int i=1;i<a.length;i++)

{

if(a[i]>max)

NOTE: If you just say (a[i]<max) then you will get the minimum value

{

max=a[i];

}}

syso(max);

**5) : How to get an array in a descending or ascending order ?**

**Ans:Example for ascending:**

int a[]={6,8,7,9,4};

Arrays.sort(a);  
                                           for(int i=0;i<a.length;i++)  
                                           {  
                                                 syso(a[i]);}

**Example for descending:**

int a[]={5,8,7,9,3};

Arrays.sort(a);

                                           for(int i=a.length-1;i>=0;i--)  
                                           {  
                                                   syso(a[i]);}  
**6) How to compare Two Arrays?**

If 2 arrays are of the sam size & data type then comparison can be done using “Arrays.equal ()”

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | int [] num1 = { 1, 2, 3 };  int[] num2 = { 4, 5, 6 };  System.out.println (Arrays. Equals (num1, num2)); //false  Int [] num3 = {1, 2, 3};  System.out.println (Arrays.equals (num1, num3)); //true |

**7) Can you explain different steps of declaring multidimensional arrays in Java?**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | //2D Arrays  int[][] twoArray1;  int twoArray2[][];  int[] twoArray3[];  //3D Arrays  int[][][] threeArray1;  int threeArray2[][][];  int[] threeArray3[][];  int[][] threeArray4[];  //4D Arrays  int[][][][] fourArray1;  int fourArray2[][][][];  int[] fourArray3[][][];  int[][] fourArray4[][];  int[][][] fourArray5[]; |

**8) How do we find duplicate elements in an array?**

Using Brute Force Method:

In this method, we compare each element of an array with other elements.

If any two elements found equal or same, we declare them as duplicates. Please find below code for the same

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | public class DuplicatesInArray {  public static void main(String[] args) {  String[] strArray1 = {"abc1", "def1", "mno1", "xyz1", "pqr1", "xyz1", "def1"};  for (int i = 0; i < strArray1.length-1; i++) {  for (int j = i+1; j < strArray1.length; j++) {  if( (strArray1[i].equals(strArray1[j])) && (i != j) ) {  System.out.println("Duplicates : "+strArray1[j]);  }}}} } |

Output: Duplicate Element is : def1

Duplicate Element is: xyz1

**9) How to iterate an array in java?**

1) Using normal for loop

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | public class MainClass1  {  public static void main(String[] args)  {  int[] a1 = new int[]{45, 12, 78, 34, 89, 21};  //Iterating over an array using normal for loop  for (int i = 0; i < a1.length; i++)  {  System.out.println(a1[i]);  }}} |

2) Using extended new for loop

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | public class MainClass1  {  public static void main(String[] args)  {  int[] a2 = new int[]{45, 12, 78, 34, 89, 21};  //Iterating over an array using extended for loop  for (int i: a2){  System.out.println(i);  }}} |

**10) How to find the missing element in integer array of 1 to 7?**

Solutions to solve this problem is to calculate sum of all numbers in the array

and compare with an expected sum, the difference would be the missing number.

* Get the sum of numbers
* total = n\*(n+1)/2
* Subtract all the numbers from sum and
* you will get the missing number.
* According to below logic sumOfNnumberss is 7\*(7+1)/2=28

sumOfElements = 1+2+3+5+6+7=24

missing element is = 28-24=4

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | int n = ar.length+1;  int total = n\*(n+1)/2;  for(int i =0;i<ar.length;i++){  total -=ar[i];}  System.out.println(total); |

**11) How to find the duplicate in an array?**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | String str = "HI RAHUL I AM FINE RAHUL"; // String with a duplicate word.  String[] words = str.split(" "); // Splitting and converting to Array .  for(int i = 0; i < words.length; i++){ //iterating array inside for loop  for(int j = i+1; j < words.length; j++){ //iterating same array inside another for loop  if (words[i].equals(words[j])){ // Using equal method i am verifying which word is repeating . System.out.println( words[i]); // here it will print duplicate .  }}} |

**12) How to check array contains value or not?**

Here is a String[] with values

public static final String[] myNames = new String[] {“B”,”A”,”K”,”C”};

If myNames contains that value then it will return true otherwise false.

Here is two methods isExists() and contains()

both the methods return true if the value is available otherwise false.

First Method

It is converting an array to ArrayList.

After that it will test if an array contains any value then it will return true otherwise false.

|  |  |
| --- | --- |
| 1  2  3 | public static <T> boolean isExists(final T[] array, final T object) {  return Arrays.asList(array).contains(object); } |

Second Method

This method loop through an array and use equal() method to search element.

This actually performs a linear search over an array in Java. It will return true if an array has provided value.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | public static <T> boolean contains(final T[] array, final T v) {  for (final T e : array) { if (e == v || v != null && v.equals(e)) {  return true;  } }  return false;  } } |

**13) How to get largest and smallest number in an array?**

Ans: Logic of this problem:

* We use two variables to store largest and smallest number.
* First, we initialize largest with Integer.MIN\_VALUE and
* Next, we initialize smallest with Integer.MAX\_VALUE.
* In each iteration of the for loop, we will compare present number with largest and smallest number, and we will update
* If a number is larger than largest, then it cannot be smaller than smallest. That means no need to check if the first condition is true,

We will use the if-else code block, where else part will only execute if the first condition is false means not true.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | import java.util.Arrays;  public class MaximumMinimumArrayExample{  public static void largestAndSmallest(int[] numbers) {  int largest = Integer.MIN\_VALUE;  int smallest = Integer.MAX\_VALUE;  for (int number : numbers) {  if (number > largest) {  largest = number;  }  else if (number < smallest) {  smallest = number;  } }  System.out.println("Largest is : " + largest);  System.out.println("Smallest is : " + smallest); } } |

**14) How to do the intersection of two sorted arrays?**

Ans: Logic: print the element if the element is present or available in both the arrays.

* Use two index variables i and j, after that initial the values i = 0, j = 0
* If arr01 [i] is smaller than arr02 [j] then increment i.
* If arr01 [i] is greater than arr02 [j] then increment j.

If both are same then print any of them and increment both i and j.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39 | public static void getIntersection(int arr01[], int arr02[], int m, int n){  int i = 0, j = 0;  while (i < m && j < n){  if (arr01[i] < arr02[j])  i++;  else if (arr02[j] < arr01[i])  j++;  else{  System.out.print(arr02[j++]+" ");  i++;  }  }  }  public static void main(String args[]){  int arr01[] = {1, 2, 4, 5, 6};  int arr02[] = {2, 3, 5, 7};  int m = arr01.length;  int n = arr02.length;  getIntersection(arr01, arr02, m, n);  } |

**15) How to get top two numbers from an array?**

Ans: Logic is:

* We will assign first variable max01 with Integer.MIN\_VALUE

Moreover, second variable max02 with same as max01 that is Integer.MIN\_VALUE.

* We will iterate this array and compare each number with max01 and max02,
* If current number is greater than max1 then assign max01 = number and max02 = max1.
* Else if it is only greater than max02 then we will only update max02 with the current number.
* At the end of an iteration, max01 and max02 points to top two numbers from given array.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | import java.util.Arrays;  public class TopTwoMaximumNumbers{  public static void main(String args[]) {  topTwoNumbers(new int[]{20, 34, 21, 87, 92, Integer.MAX\_VALUE});  topTwoNumbers(new int[]{0, Integer.MIN\_VALUE, -2});  topTwoNumbers(new int[]{Integer.MAX\_VALUE, 0, Integer.MAX\_VALUE});  topTwoNumbers(new int[]{1, 1, 0}); }  public static void topTwoNumbers(int[] numbers) {  int max01 = Integer.MIN\_VALUE;  int max02 = Integer.MIN\_VALUE; for (int number : numbers) {  if (number > max01) { max02 = max01; max01 = number; }  else if (number > max02) { max02 = number; } }  System.out.println("First largest number is : " + max01);  System.out.println("Second largest number is : " + max02); } } |

**16) How to cut or remove an element from the array?**

Ans: Logic is: We can remove or cut an element using Apache Commons ArrayUtils based on an index.

ArrayUtils has several overloaded remove() method.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | import java.util.Arrays;  import org.apache.commons.lang.ArrayUtils;  public class RemoveFromArray{  public static void main(String args[]) {  //let's create an array for demonstration purpose  int[] testArr = new int[] { 10, 102, 13, 14, 105};  System.out.println("Array size : " + testArr.length );  System.out.println("Find Contents : " + Arrays.toString(testArr));  //let's remove or delete an element from Array using Apache Commons ArrayUtils  testArr = ArrayUtils.remove(testArr, 2); //removing element at index 2  //Size of array must be 1 less than original array after deleting an element  System.out.println("Size of array after removing : " + testArr.length);  System.out.println("Content of Array : "+ Arrays.toString(testArr));  } } |

**17) What is the logic to reverse the array?**

Ans: Logic: print the element in reverse order.

* Declare a String Array String[] s = new String[]{“My”,”Leg”,”is”,”cut”};
* Iterate it using for loop get all elements in reverse order means end point to start point.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | public static void main(String[] args)  {String[] s = new String[]{"My","Leg","is","cut"};  for(int i=s.length-1; i>=0; i--){  System.out.println("reverse "+s[i]);}} |

**18) How do you find the second largest element in an array of integers?**

Ans: Logic is:

* Iterate the given array using for loop.
* (first if condition arr[i] > largest):
* If current array value is greater than largest value then
* Move the largest value to secondLargest and make current value as largest
* (second if condition arr[i] > secondLargest )
* If the current value is smaller than largest and

greater than secondLargest then the current value becomes

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | public static void main(String[] args) {  int myArr[] = { 14, 46, 47, 86, 92, 52, 48, 36, 66, 85 };  int largest = myArr[0];  int secondLargest = myArr[0];  System.out.println("The given array is:" );  for (int i = 0; i < myArr.length; i++) {  System.out.print(myArr[i]+"\t");  }  for (int i = 0; i < myArr.length; i++) {  if (myArr[i] > largest) {  secondLargest = largest;  largest = myArr[i];  } else if (myArr[i] > secondLargest) {secondLargest = myArr[i];  }}  System.out.println("\nSecond largest number is:" + secondLargest);  } |

**19) Write a program to sum values of given array.**

Ans: Logic is:

* Declare and assign int variable that is sum =0.
* Iterate the given array using for loop.

Add all the array element and keep in this variable that is the sum.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | Public static void main(String[] args) {  int my\_array[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  int sum = 0;  for (int i: my\_array)  sum += i;  System.out.println("The sum is " + sum);  } |

**20) How do you separate zeros from non-zeros in an array?**

Ans: Logic is:

* Initializing counter to 0
* Traversing inputArray from left to right
* If inputArray[i] is non-zero
* Assigning inputArray[i] to inputArray[counter]
* Incrementing the counter by 1

Assigning zero to remaining elements

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | Public class SeparateZerosFromNonZeros  { static void moveZerosToEnd(int inputArray[]) { int counter = 0;  for (int i = 0; i < inputArray.length; i++)  {  if(inputArray[i] != 0) {  inputArray[counter] = inputArray[i];  counter++;  } }  while (counter < inputArray.length) {  inputArray[counter] = 0;  counter++; }  System.out.println(Arrays.toString(inputArray));  }  public static void main(String[] args) {  moveZerosToEnd(new int[] {12, 0, 7, 0, 8, 0, 3});  moveZerosToEnd(new int[] {1, -5, 0, 0, 8, 0, 1});  moveZerosToEnd(new int[] {0, 1, 0, 1, -5, 0, 4});  moveZerosToEnd(new int[] {-4, 1, 0, 0, 2, 21, 4});  } } |

**21) Write a program to insert an element and in the specific position in the array?**

Ans: Logic is:

* Insert an element in 3rd position of the array (index->2, value->5)
* Iterate the given array in reverse order using for loop.

Now insert given position and value after for loop.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | Public static void main(String[] args) {  int[] my\_array = {25, 14, 56, 15, 36, 56, 77, 18, 29, 49};  int Index\_position = 2;  int newValue = 5;  System.out.println("Original Array : "+Arrays.toString(my\_array));  for(int i=my\_array.length-1; i > Index\_position; i--){  my\_array[i] = my\_array[i-1];  }my\_array[Index\_position] = newValue;  System.out.println("New Array: "+Arrays.toString(my\_array));  } |

**22) How to get the index of an array element?**

Ans: Logic is:

* In each step, it checks the input key value with the value of the middle element of an array.
* If the keys match then it will return the position. In another case, if the key is less than the middle element’s key,

Then it will repeat the while loop. If the remaining array searched and it reduced to zero it will return -1 means not found

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | public static int findIndex (int[] my\_array, int t) {  int startPoint = 0;  int endPoint = my\_array.length - 1;  while (startPoint <= endPoint) {  int mid = (startPoint + end) / 2;  if (key == my\_array[mid]) {  return mid;  }  if (key < my\_array[mid]) {  endPoint = mid - 1;  } else {  startPoint = mid + 1;  }  }  return -1;  }  Public static void main(String[] args) {  int[] arr = {2, 4, 6, 8, 10, 12, 14, 16};  System.out.println("Key 14's position: "findIndex(arr, 14));  int[] arr1 = {6,34,78,123,432,900};  System.out.println("Key 432's position: "+findIndex(arr1, 432));  } |

**23) Can we extend an array after initialization?**

Ans: Logic is: We can extend an array after initialization by adding a new array.

Please find below example.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | Public static void main(String[] args) {  String [] names = new String[] { "A", "B", "C" };  String [] extended = new String[5];  extended [3] = "D";  extended [4] = "E";  System.arraycopy(names, 0, extended, 0, names.length);  //here we are extending with new array extended[3] and extended[4].  for (String str: extended){  System.out.println(str);  }  } |

output is A,B,C,D,E.

**24) How to fill element (initialize at once) in an array?**

Ans: This example fill(initialize all the elements of the array in one short) an array by using

Array.fill(array name,value) method and

Array.fill(array name, starting index, ending index, value)

the method of Java Util class.

Java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | import java.util.\*;  public class FillTest {  public static void main(String args[]) {  int array[] = new int[6];  Arrays.fill(array, 100);  for (int i = 0, n = array.length; i < n; i++) {  System.out.println(array[i]);  }  System.out.println();  Arrays.fill(array, 3, 6, 50);  for (int i = 0, n = array.length; i< n; i++) {  System.out.println(array[i]);}}} |

output is

100,100,100,100,100,100

100,100,100,50,50,50