INTERVIEW_ASKED_CODING

(1) IN A GIVEN STRING “AAABBAAABBBAABBAA” INSTEAD OF AB PAIR PRINT A ONLY?

```
public static void main(String[] args) {
String str = "aaabbaaabbbaabbaa";
String result = replaceConsecutiveAB(str);
System.out.println(result);
}
static String replaceConsecutiveAB(String str) {
StringBuilder sb = new StringBuilder(str);
int i = 0;
while (i < sb.length() - 1) {
if (sb.charAt(i) == 'a' && sb.charAt(i + 1) == 'b') {
sb.replace(i, i + 2, "a");
i = 0; // Restart from the beginning of the string
} else {
i++;
}
}
return sb.toString();
}
```

(2) WRITE PROGRAM FOR SINGLE TON DESIGN PATTERN?

```
public class A {
      //Variable
    private static A a=null;

    //constructor
    private A() {
 }
    //a method which return type class name
    public    static A getInstance() {
          synchronized(A.class) {
          if(a==null) {
                 a=new A();
          }}
          return a;
    }}

public class AImpl {
public static void main(String[] args) {
      A a1 = A.getInstance();
      System.out.println(a1);

      A a2 = A.getInstance();
      System.out.println(a2);
    A a3 = A.getInstance();
    System.out.println(a3);
}
}
```

(3) WRITE THE CODE FOR MAX AND MIN SALARY ?

```
public static void main(String[] args) {
        int[] numbers = {5, 2, 9, 1, 7, 3};
        int min = Integer.MAX_VALUE;
        int max = Integer.MIN_VALUE;

        for (int number : numbers) {
            if (number < min) {
                min = number;
            }
            if (number > max) {
                max = number;
            }
        }

        System.out.println("Minimum number: " + min);
        System.out.println("Maximum number: " + max);    }
```

(4) WRITE THE CODE FOR PRINTING COMMON STRING FROM TWO ARRAY LIST?

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

List<String> names1 = Arrays.asList("JAVA", "HIBERNATE",
"SPRING","spring boot in console");

List<String> names2 = Arrays.asList("java", "hibernate", "spring");

for (String string : names1) {

for (String string2 : names2) {

if (string.equalsIgnoreCase(string2)) {

System.out.println(string);

}}}}
```

(5) IN GIVEN ARRAYLIST PRINT HOW MANY LETTER HOW MUCH TIMES OCCURING IN A SINGLE STRING ?

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

List<String> names = Arrays.asList("sahil","rahul","punit","delhi");

Map<String, Map<Character, Long>> letterCountMap = names.stream()

.collect(Collectors.toMap(name -> name,name -> name.chars()

.mapToObj(c -> (char) c)

.collect(Collectors.groupingBy(c -> c, Collectors.counting()))

));

System.out.println("Letter occurrences in each name:");

for (Map.Entry<String, Map<Character, Long>> entry :
letterCountMap.entrySet()) {
```

```
String name = entry.getKey();
Map<Character, Long> charCountMap = entry.getValue();
System.out.println(name + ": " + charCountMap);
}
}
```

(6) IN GIVEN STRING FIND HOW MANY TIMES WHICH LETTER OCCURE IN OVERALL?

```
public static void main(String[] args) {
List<String> names = Arrays.asList("sahil","rahul","punit","delhi");
Map<Character, Long> letterCountMap = names.stream()
.flatMap(name -> name.chars().mapToObj(c -> (char) c))
.collect(Collectors.groupingBy(c -> c, Collectors.counting()));
System.out.println("Letter occurrences:");
for (Map.Entry<Character, Long> entry : letterCountMap.entrySet()) {
char letter = entry.getKey();
long count = entry.getValue();
System.out.println(letter + ": " + count);
}
}
```

(7) WRITE THE PROGRAM FOR PENDRUM?

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

```
String str ="racecar";
boolean value = isPendrum(str);// calling static method
System.out.println(value);
}
static boolean isPendrum(String str) {
for (int i = 0; i <str.length()-1; i++) {
if(str.charAt(i)==str.charAt(str.length()-1-i)) {
return true;
}
}
return false;
}
```

(8) FROM ARRAY PRINT THE NUMBER WHICH CAN BE DEVISBLE BY 6 ONLY?

```
public static void main(String[] args) {
int[] x = {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, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60,
61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80,
81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100};
//using stream
Arrays.stream(x)
.filter(num -> num % 6 == 0)
.forEach(System.out::println);
// developing logic
for (int num : x) {
if (num % 6 == 0) {
System.out.println(num);
}
}
}
```

(9) WRITE CODE FOR PRINTING US DATE AND TIME?

```
public static void main(String[] args) {
// Get the current date and time in the default system timezone
ZonedDateTime currentDateTime = ZonedDateTime.now();
// Convert the current date and time to the USA timezone
ZoneId usaZone = ZoneId.of("America/New_York");
ZonedDateTime usaDateTime =
currentDateTime.withZoneSameInstant(usaZone);
// Format the date and time as strings
DateTimeFormatter dateFormatter =
DateTimeFormatter.ofPattern("MM/dd/yyyy");
DateTimeFormatter timeFormatter =
DateTimeFormatter.ofPattern("hh:mm:ss a");
String usaDate = usaDateTime.format(dateFormatter);
String usaTime = usaDateTime.format(timeFormatter);
// Print the USA date and time
System.out.println("USA Date: " + usaDate);
System.out.println("USA Time: " + usaTime);
}
```

(10) WRITE PROGRAM FOR BRACKET MATCHING?

```
public static void main(String[] args) {
Scanner scan = new Scanner(System.in);
System.out.println("Please enter the bracket: ");
String value = scan.nextLine();
scan.close();
Stack<Character> stack = new Stack<>();
boolean matchingValue = IntStream.range(0, value.length())
.allMatch(le -> {
char ch = value.charAt(le);
if (ch == '(' || ch == '{' || ch == '[') {
stack.push(ch);
return true;
} else if (ch == ')' || ch == '}' || ch == ']') {
if (stack.empty()) {
return false;
} else {
char retrieve = stack.pop();
return (ch == ')' && retrieve == '(') ||
(ch == '}' && retrieve == '{') ||
(ch == ']' && retrieve == '[');
}
}
return true;
});
matchingValue = matchingValue && stack.empty();
System.out.println("Bracket matching: " + matchingValue);
}
```

(11) WRITE THE CODE FOR SINGLE BRACKET MACTHING ?

```
public static void main(String[] args) {
Scanner scan = new Scanner(System.in);
System.out.println("enter opening and closing brackets");
String str = scan.next();
int count1=0;
int count2=0;
for (int i = 0; i < str.length(); i++) {
if(str.charAt(i)=='(') {
count1++;
}else if(str.charAt(i)==')'){
count2++;
}
}
if(count1==count2) {
System.out.println("No Error");
}else {
System.out.println("Error");
}
}
```

(12) WRITE THE CODE FOR COUNTING STRING CHARACTERS?

```
public static void main(String[] args) {
String str="sahill ";
int count =0;
for (int i = 0; i <str.length(); i++) {
char at = str.charAt(i);
if(Character.isLetter(at)) {
count++;
}
}
System.out.println(count);
System.out.println(str.length());
}
```

(13) WRITE CODE FOR REVERSE THE STRING ?

```
public static void main(String[] args) {
String str= " pankaj sir academy";
//splitting string in new line each word
String[] split = str.split(" ");
for (String string : split) {
System.out.println(string);
}
//trim method is using to remove both side spaces
String[] split2 = str.trim().split(" ");
for (String string : split2) {
System.out.println(string);
}
//Reverse the String
for (int i = str.length()-1; i>=0; i--) {
System.out.print(str.charAt(i));
}
}
```

(14) WRITE THE CODE FOR COUNTING CHARACTERS IN OF STRING?

```
public static void main(String[] args) {
String str = "HelloWorld";
HashMap<Character, Integer> countChars= new HashMap<>();
for (int i = 0; i < str.length(); i++) {
char c =str.charAt(i);
if(countChars.containsKey(c)) {
int count = countChars.get(c);
countChars.put(c, count+1);
}else {
countChars.put(c, 1);
}
}
for (char c : countChars.keySet()) {
int count = countChars.get(c);
System.out.println(" ' "+ c +" 'occurs' "+count+" times" );
}
}
```

(15) WRITE CODE FOR FINDING SECOND MAX ?

```
public static void main(String[] args) {
int[] numbers = {5, 2, 9, 1, 7, 13};
int max = Integer.MIN_VALUE;
int secondMax = Integer.MIN_VALUE;
for (int number : numbers) {
if (number > max) {
```

```
secondMax = max;
max = number;
} else if (number > secondMax && number < max) {
secondMax = number;
}
}
System.out.println("Second maximum number: " + secondMax);
}
```

(16) WRITE CODE FOR FINDING SECOND MIN SALARY?

```
public static void main(String[] args) {
int[] numbers = {5, 2, 9, 1, 7, 13};
int max = Integer.MIN_VALUE;
int secondMax = Integer.MIN_VALUE;
for (int number : numbers) {
if (number > max) {
secondMax = max;
max = number;
} else if (number > secondMax && number < max) {
secondMax = number;
}
}
System.out.println("Second maximum number: " + secondMax);
}
```

(17) FROM LIST PRINT HOW MANY CHARACTERS ARE THERE?

```
public static void main(String[] args) {
String name = "sahil satich new delhi";
int count=0;
for (int i = 0; i <name.length(); i++) {
char ch = name.charAt(i);
if(Character.isLetter(ch)) {
count++;
}
}
System.out.println(count);
System.out.println(name.length());
}
```

(18) PRINT TABLE OF ANY DIGIT?

```
public static void main(String[] args) {
Scanner scan = new Scanner(System.in);
System.out.println("enter any number");
int number = scan.nextInt();
for (int i = 1; i <=number; i++) {
```

```
int result= i* number;
System.out.println(i+"*"+number+"="+ result);
}
}
```

(19) WRITE THE CODE FOR REVERSE THE STRING?

```
public static void main(String[] args) {
String str ="sahil";
for (int i = str.length()-1; i >=0; i--) {
System.out.print(str.charAt(i));
}
}
```

(20) WRITE THE CODE FOR SERIALIZATION AND DESERIALIZATION?

```
public static void main(String[] args) {
// Object to be serialized
Person person = new Person("John Doe", 30);
String filename = "E:\\serialization.ser";
// Serialization
try {
FileOutputStream fileOut = new FileOutputStream(filename);
ObjectOutputStream out = new ObjectOutputStream(fileOut);
out.writeObject(person);
out.close();
fileOut.close();
System.out.println("Serialization complete. Serialized data is saved in " +
filename);
} catch (IOException e) {
e.printStackTrace();
}
// Deserialization
Person deserializedPerson = null;
try {
FileInputStream fileIn = new FileInputStream(filename);
ObjectInputStream in = new ObjectInputStream(fileIn);
deserializedPerson = (Person) in.readObject();
in.close();
fileIn.close();
} catch (IOException e) {
e.printStackTrace();
return;
} catch (ClassNotFoundException e) {
e.printStackTrace();
return;
}
// Print deserialized object
```

```
System.out.println("Deserialized data:");
System.out.println("Name: " + deserializedPerson.getName());
System.out.println("Age: " + deserializedPerson.getAge());
}
}
class Person implements Serializable {
private String name;
private int age;
transient private String username;
public Person(String name, int age) {
this.name = name;
this.age = age;
}
public String getName() {
return name;
}
public int getAge() {
return age;
}
```

(21) WRITE THE PROGRAM FOR STRING CASE SWAPPER?

```
public static void main(String[] args) {
String input = "Hello, World!";
String swapped = swapCase(input);
System.out.println("Swapped string: " + swapped);
}
public static String swapCase(String input) {
char[] charArray = input.toCharArray();
for (int i = 0; i < charArray.length; i++) {
char c = charArray[i];
if (Character.isUpperCase(c)) {
charArray[i] = Character.toLowerCase(c);
} else if (Character.isLowerCase(c)) {
charArray[i] = Character.toUpperCase(c);
}
}
return new String(charArray);
}
```

(22) WRITE THE CODE FOR PERMUTABLE/SUB STRING COUNT?

```
    public static void main(String[] args) {
String str = "SUB";
permute(str, 0, str.length() - 1);
}
private static void permute(String str, int l, int r) {
```

```
if (l == r) {
System.out.println(str);
} else {
for (int i = l; i <= r; i++) {
str = swap(str, l, i);
permute(str, l + 1, r);
str = swap(str, l, i); // backtrack
}
}
}
private static String swap(String str, int i, int j) {
char[] charArray = str.toCharArray();
char temp = charArray[i];
charArray[i] = charArray[j];
charArray[j] = temp;
return String.valueOf(charArray);
}
```

(23) swapping of two numbers without using temp variable?

```
public class Practice {
public static void main(String[] args) {
int a=10;
int b=20;
```

```
System.out.println("values of a and b befor swapping: a="+a+"and b= "+b);

a=a+b; //a=30

b=a-b; //b=30-20=10

a=a-b; //30-10=20

System.out.println("values of a and b After swapping: a="+a+"and b= "+b);
```

(24) Reversing of Integer?

```
int num=1234;

int rev=0;

while (num!=0) {

rev=rev*10+num%10;

num=num/10;

}

System.out.println("Reversed number: "+rev);
```

(25) Reversing of String?

```
String name="Prashant";

String revString="";

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

revString=revString+name.charAt(i);
}
System.out.println("reversed string is: " +revString);
```

(26) String to charArray and Reversing of given string by converting that to charArray?

```
String name="prashantGani";
String revString="";
char[] charArray = name.toCharArray();
for (int i = name.length()-1; i>=0; i--) {
revString=revString+charArray[i];
}
System.out.println(revString);
```

(27) sum of digits are present ?

```
int num=1234;
int sum=0;
while (num>0) {
sum=sum+num%10;
num=num/10;
}
System.out.println("sum of number is: "+sum);
```

(28) Count number of words?

```
String nameString="my name is prashant";
  String[] split = nameString.trim().split(" ");
  System.out.println(split.length);
```

(29) Find the biggest of the three numbers?

```
int n1=255;
int n2=355;
int n3=566;
if (n1>n2 && n1>n3) {
System.out.println("biggest among three number is: "+n1);
}else if (n2>n3 && n2>n3) {
System.out.println("biggest among three number is: "+n2);
}else {
System.out.println("biggest among three number is : "+n3);
}
```

(30) Find even and odd numbers in variables?

```
                int num=2121212;
int evenCount=0;
int oddCount=0;
while (num>0) {
int rem=num%10;
if (rem%2==0) {
evenCount++;
}else {
oddCount++;
}
num=num/10;
}
System.out.println("number of even count is: "+evenCount);
System.out.println("number of even count is: "+oddCount);
```

(31) chech wether given number is fibanocci or not?

```
   System.out.println("enter the number to check number is fibanocci or nor:
");
Scanner scanner=new Scanner(System.in);
```

```
int num = scanner.nextInt();
if (num<0) {
System.out.println("please enetr the positve number to check wether the
number is
fibanocci or not");
}else {
int a=0,b=1,c=0;
while (c<num) {
c=a+b;
a=b;
b=c;
}
if (c==num) {
System.out.println("the give number "+num+" is fibanocci number");
}else {
System.out.println("the give number "+num+" is NOT fibanocci number");
}
}
```

(32) inding the given two arrays are equal or not?

```
int[] num1= {1,2,3,4,5,6,7,8,9,10,456};
```

```
int[] num2= {1,2,3,4,5,6,7,8,9,10,456};
boolean status=Arrays.equals(num1, num2);
if(status==true) {
System.out.println("both the arrays are equal");
}else {
System.out.println("bot the arrays are not same");
}
```

(33) sorting of integer array?

ONE WAY:

```
                    int[] num= {16,28,346,499,554,654,17,845,659,1240,456};
for(int i=0;i<num.length;i++){
for(int j=i+1;j<num.length;j++) {
if(num[i]>num[j]) {
int temp=num[i];
num[i]=num[j];
num[j]=temp;
}
}
}
for (int value : num) {
```

```
System.out.print(" "+ value);
}
```

SECOND WAY:

```
                    int[] num= {16,28,346,499,554,654,17,845,659,1240,456};
Arrays.sort(num);
for (int value : num) {
System.out.print(" "+value);
}
```

(34) finding duplicate in integer array?

ONE WAY:

```
                                 int []num= {1,8,2,3,4,5,6,7,8,2,9};
for(int i=0;i<num.length;i++) {
for(int j=i+1;j<num.length;j++) {
if (num[i]==num[j]) {
//System.out.print("duplicate ements is: ");
System.out.print(" "+num[j]);
}
 }
}
```