

Top 20 Java Interview Programs For Programming And Coding Interview

Q #1) Write a Java Program to reverse a string without using String inbuilt function.

Answer: Here, we are initializing a string variable str and making use of the string builder class.

The object of the string builder class str2 will be further used to append the value stored in the string variable str.

Thereafter, we are using the inbuilt function of the string builder (reverse()) and storing the new reversed string in str2. Finally, we are printing str2.

Following program code explains this:

```
public class FinalReverseWithoutUsingStringMethods {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        String str = "Automation";  
        StringBuilder str2 = new StringBuilder();  
        str2.append(str);  
        str2 = str2.reverse(); // used string builder to reverse  
        System.out.println(str2);  
    }  
}
```

Output:

noitamotua

Q #2) Write a Java Program to reverse a string without using String inbuilt function reverse().

Answer: There are several ways with which you can reverse your string if you are allowed to use the other string inbuilt functions.

Method 1:

In this method, we are initializing a string variable called str with the value of your given string. Then, we are converting that string into a character array with the toCharArray() function. Thereafter, we are using for loop to iterate between each character in reverse order and printing each character.

```
public class FinalReverseWithoutUsingInbuiltFunction {  
    public static void main(String[] args) {  
        String str = "Saket Saurav";  
        char chars[] = str.toCharArray(); // converted to character array and printed in reverse order  
        for(int i= chars.length-1; i>=0; i--) {  
            System.out.print(chars[i]);  
        }  
    }  
}
```

Output:

varuaS tekaS

Method 2:

This is another method in which you are declaring your string variable str and then using Scanner class to declare an object with a predefined standard input object.

This program will accept the string value through the command line (when executed).

We have used nextLine() which will read the input with the spaces between the words of a string. Thereafter, we have used a split() method to split the string into its substrings(no delimiter given here). Finally, we have printed the string in reverse order using for loop.

```

import java.util.Scanner;

public class ReverseSplit {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        String str;

        Scanner in = new Scanner(System.in);

        System.out.println("Enter your String");

        str = in.nextLine();

        String[] token = str.split(""); //used split method to print in reverse order
        for(int i=token.length-1; i>=0; i--)
        {
            System.out.print(token[i] + "");
        }

    }

}

```

Output:

```

Enter your String
Softwaretestinghelp
plehgnitseterawtfoS

```

Method 3:

This is almost like method 2, but here we did not use the split() method. We have used the scanner class and nextLine() for reading the input string. Then, we have declared an integer length which has the length of the input string.

Thereafter, we have printed the string in the reverse order using for loop. However, we have used the `charAt(index)` method which will return the character at any specific index. After each iteration, the character will be concatenated to reverse the string variable.

Finally, we have printed the reverse string variable.

```
import java.util.Scanner;
```

```
public class Reverse {
```

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

```
        // TODO Auto-generated method stub
```

```
        String original, reverse = "";
```

```
        System.out.println("Enter the string to be reversed");
```

```
        Scanner in = new Scanner(System.in);
```

```
        original = in.nextLine();
```

```
        int length = original.length();
```

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

```
            reverse = reverse + original.charAt(i); //used inbuilt method charAt() to reverse the string
```

```
        }
```

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

```
    }
```

```
}
```

Output:

Enter the string to be reversed

automation testing

gnitset noitamotua

Q #3) Write a Java Program to swap two numbers using the third variable.

Answer: In this example, we have made use of the Scanner class to declare an object with a predefined standard input object. This program will accept the values of x and y through the command line (when executed).

We have used nextInt() which will input the value of an integer variable 'x' and 'y' from the user. A temp variable is also declared.

Now, the logic of the program goes like this – we are assigning temp or third variable with the value of x, and then assigning x with the value of y and again assigning y with the value of temp. So, after the first complete iteration, the temp will have a value of x, x will have a value of y and y will have a value of temp (which is x).

```
import java.util.Scanner;
```

```
public class SwapTwoNumbers {
```

```
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        int x, y, temp;  
        System.out.println("Enter x and y");  
        Scanner in = new Scanner(System.in);  
        x = in.nextInt();  
        y = in.nextInt();  
        System.out.println("Before Swapping" + x + y);  
        temp = x;  
        x = y;  
        y = temp;  
        System.out.println("After Swapping" + x + y);  
    }
```

```
}
```

Output:

Enter x and y

45

98

Before Swapping4598

After Swapping9845

Q #4) Write a Java Program to swap two numbers without using the third variable.

Answer: Rest all things will be the same as the above program. Only the logic will change. Here, we are assigning x with the value $x + y$ which means x will have a sum of both x and y.

Then, we are assigning y with the value $x - y$ which means we are subtracting the value of y from the sum of $(x + y)$. Till here, x still has the sum of both x and y. But y has the value of x.

Finally, in the third step, we are assigning x with the value $x - y$ which means we are subtracting y (which has the value of x) from the total $(x + y)$. This will assign x with the value of y and vice versa.

```
import java.util.Scanner;
```

```
class SwapTwoNumberWithoutThirdVariable
```

```
{
```

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

```
    {
```

```
        int x, y;
```

```
        System.out.println("Enter x and y");
```

```
        Scanner in = new Scanner(System.in);
```

```
        x = in.nextInt();
```

```
        y = in.nextInt();
```

```
        System.out.println("Before Swapping\nx = "+x+"\ny = "+y);
```

```

    x = x + y;

    y = x - y;

    x = x - y;

    System.out.println("After Swapping without third variable\nx = "+x+"\ny = "+y);
}
}

```

Output:

Enter x and y

45

98

Before Swapping

x = 45

y = 98

After Swapping without a third variable

x = 98

y = 45

Q #5) Write a Java Program to count the number of words in a string using HashMap.

Answer: This is a collection class program where we have used HashMap for storing the string.

First of all, we have declared our string variable called str. Then we have used split() function delimited by single space so that we can split multiple words in a string.

Thereafter, we have declared HashMap and iterated using for loop. Inside for loop, we have an if-else statement in which wherever at a particular position, the map contains a key, we set the counter at that position and add the object to the map.

Each time, the counter is incremented by 1. Else, the counter is set to 1.

Finally, we are printing the HashMap.

Note: The same program can be used to count the number of characters in a string. All you need to do is to remove one space (remove space delimited in split method) in String[] split = str.split("");

```
import java.util.HashMap;
```

```
public class FinalCountWords {
```

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

```
        // TODO Auto-generated method stub
```

```
        String str = "This this is is done by Saket Saket";
```

```
        String[] split = str.split(" ");
```

```
        HashMap<String,Integer> map = new HashMap<String,Integer>();
```

```
        for (int i=0; i<split.length; i++) {
```

```
            if (map.containsKey(split[i])) {
```

```
                int count = map.get(split[i]);
```

```
                map.put(split[i], count+1);
```

```
            }
```

```
            else {
```

```
                map.put(split[i], 1);
```

```
            }
```

```
        }
```

```
        System.out.println(map);
```

```
    }
```

```
}
```

Output:


```
{Saket=2, by=1, this=1, This=1, is=2, done=1}
```

Q #6) Write a Java Program to iterate HashMap using While and advance for loop.

Answer: Here we have inserted three elements in HashMap using put() function.

The size of the map can get using the size() method. Thereafter, we have used a While loop for iterating through the map which contains one key-value pair for each element. Keys and Values can be retrieved through getKey() and getValue().

Likewise, we have used advanced for loop where we have a "me2" object for the HashMap.

```
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;

public class HashMapIteration {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        HashMap<Integer,String> map = new HashMap<Integer,String>();
        map.put(2, "Saket");
        map.put(25, "Saurav");
        map.put(12, "HashMap");
        System.out.println(map.size());
        System.out.println("While Loop:");
        Iterator itr = map.entrySet().iterator();
        while(itr.hasNext()) {
            Map.Entry me = (Map.Entry) itr.next();
            System.out.println("Key is " + me.getKey() + " Value is " + me.getValue());
        }
    }
}
```

```

        System.out.println("For Loop:");
        for(Map.Entry me2: map.entrySet()) {
            System.out.println("Key is: " + me2.getKey() + " Value is: " + me2.getValue());
        }
    }
}

```

Output:

3

While Loop:

Key is 2 Value is Saket

Key is 25 Value is Saurav

Key is 12 Value is HashMap

For Loop:

Key is: 2 Value is: Saket

Key is: 25 Value is: Saurav

Key is: 12 Value is: HashMap

Q #7) Write a Java Program to find whether a number is prime or not.

Answer: Here, we have declared two integers temp and num and used Scanner class with nextInt(as we have integer only).

One boolean variable isPrime is set to true. Thereafter, we have used for loop starting from 2, less than half of the number is entered and incremented by 1 for each iteration. Temp will have the remainder for every iteration. If the remainder is 0, then isPrime will be set to False.

Based on isPrime value, we are coming to the conclusion that whether our number is prime or not.

```
import java.util.Scanner;
```

```

public class Prime {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        int temp, num;
        boolean isPrime = true;
        Scanner in = new Scanner(System.in);
        num = in.nextInt();
        in.close();
        for (int i = 2; i<= num/2; i++) {
            temp = num%i;
            if (temp == 0) {
                isPrime = false;
                break;
            }
        }
        if(isPrime)
            System.out.println(num + "number is prime");
        else
            System.out.println(num + "number is not a prime");

    }

}

```

Output:

445

445number is not a prime

Q #8) Write a Java Program to find whether a string or number is palindrome or not.

Answer: You can use any of the reverse string program explained above to check whether the number or string is palindrome or not.

What you need to do is to include one if-else statement. If the original string is equal to a reversed string then the number is a palindrome, otherwise not.

```
import java.util.Scanner;

public class Palindrome {
    public static void main (String[] args) {
        String original, reverse = "";
        Scanner in = new Scanner(System.in);
        int length;
        System.out.println("Enter the number or String");
        original = in.nextLine();
        length = original.length();
        for (int i =length -1; i>=0; i--) {
            reverse = reverse + original.charAt(i);
        }
        System.out.println("reverse is:" +reverse);

        if(original.equals(reverse))
            System.out.println("The number is palindrome");
        else
            System.out.println("The number is not a palindrome");

    }
}
```

Output:

For String-

Enter the number or String

vijay

reverse is:yajiv

The number is not a palindrome

For Number-

Enter the number or String

99

reverse is:99

The number is palindrome

Q #9) Write a Java Program for the Fibonacci series.

Answer: Fibonacci series is a series of numbers where after the initial two numbers, every occurring number is the sum of two preceding numbers.

For Example 0,1,1,2,3,5,8,13,21.....

In this program, we have used Scanner class again with nextInt (discussed above). Initially, we are entering (through command line) the number of times the Fibonacci has to iterate. We have declared integer num and initialized a,b with zero and c with one. Then, we have used for loop to iterate.

The logic goes like a is set with the value of b which is 0, then b is set with the value of c which is 1. Then, c is set with the sum of both a and b.

```
import java.util.Scanner;
```

```
public class Fibonacci {  
    public static void main(String[] args) {  
        int num, a = 0, b = 0, c = 1;
```

```

Scanner in = new Scanner(System.in);

System.out.println("Enter the number of times");

num = in.nextInt();

System.out.println("Fibonacci Series of the number is:");

for (int i=0; i<num; i++) {

    a = b;

    b = c;

    c = a+b;

    System.out.println(a + ""); //if you want to print on the same line, use print()

}

}

}

```

Output:

Enter the number of times

10

Fibonacci Series of the number is:

0

1

1

2

3

5

8

13

21

34

Q #10) Write a Java Program to iterate ArrayList using for-loop, while-loop, and advance for-loop.

Answer: In this program, we have inserted three elements and printed the size of the ArrayList.

Then, we have used While Loop with an iterator. Whenever the iterator has (next) element, it will display that element until we reach the end of the list. So it will iterate three times.

Likewise, we have done for Advanced For Loop where we have created an object called obj for the ArrayList called list. Then printed the object.

Thereafter, we have put the condition of For Loop where the iterator i is set to 0 index, then it is incremented by 1 until the ArrayList limit or size is reached. Finally, we have printed each element using a get(index) method for each iteration of For Loop.

```
import java.util.*;
```

```
public class arrayList {  
    public static void main(String[] args) {  
        ArrayList list = new ArrayList();  
        list.add("20");  
        list.add("30");  
        list.add("40");  
        System.out.println(list.size());  
        System.out.println("While Loop:");  
        Iterator itr = list.iterator();  
        while(itr.hasNext()) {  
            System.out.println(itr.next());  
        }  
        System.out.println("Advanced For Loop:");  
        for(Object obj : list) {  
            System.out.println(obj);  
        }  
        System.out.println("For Loop:");  
        for(int i=0; i<list.size(); i++) {  
            System.out.println(list.get(i));  
        }  
    }  
}
```

```
    }  
}  
}
```

Output:

3

While Loop:

20

30

40

Advanced For Loop:

20

30

40

For Loop:

20

30

40

Q #11) Write a Java Program to demonstrate an explicit wait condition check.

Answer: There are two main types of wait – implicit and explicit. (We are not considering Fluent wait in this program)

The implicit wait is those waits that are executed irrespective of any condition. In the below program, you can see that it is for Google Chrome and we have used some inbuilt methods to set the property, maximizing window, URL navigation, and web element locating.

```
WebDriverWait wait = new WebDriverWait(driver, 20);
```

```
WebElement element2 =  
wait.until(ExpectedConditions.visibilityOfElementLocated(By.partialLinkText("Software testing -  
Wikipedia")));
```



```
element2.click();
```

In the above piece of code, you can see that we have created an object wait for WebDriverWait and then we have searched for WebElement called element2.

The condition is set in such a way that the webdriver will have to wait until we see the link “Software testing – Wikipedia” on a web page. It won’t execute if it does not find this link. If it does, then it will do a mouse click on that link.

```
package Codes;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class explicitWaitConditionCheck {

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        System.setProperty("webdriver.chrome.driver", "C:\\\\webdriver\\\\chromedriver.exe");

        ChromeOptions options = new ChromeOptions();

        options.addArguments("--disable-arguments");

        WebDriver driver = new ChromeDriver();

        driver.manage().window().maximize();

        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);

        driver.navigate().to("https://www.google.com");

        WebElement element = driver.findElement(By.name("q"));

        element.sendKeys("Testing");

        element.submit();
```

```

        WebDriverWait wait = new WebDriverWait(driver, 20);

        WebElement element2 =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.partialLinkText("Software testing -
Wikipedia"))));
element2.click();

}}

```

Q #12) Write a Java Program to demonstrate Scroll up/ Scroll down.

Answer: All the lines of codes are easily relatable as we have discussed in our previous example.

However, in this program, we have included our JavascriptExecutor js which will do the scrolling. If you see the last line of the code, we have passed window.scrollTo(arg1,arg2).

If you want to scroll up then pass some value in arg1 if you want to scroll down then pass some value in arg2.

```

package Codes;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class ScrollDown {

    public static void main(String[] args) {

        // TODO Auto-generated method stub
    }
}

```

```

        System.setProperty("webdriver.chrome.driver", "C:\\webdriver\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        JavascriptExecutor js = (JavascriptExecutor) driver;
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
        driver.get("https://www.google.com");
        WebElement element = driver.findElement(By.name("q"));
        element.sendKeys("SoftwareTestingHelp");
        element.sendKeys(Keys.ENTER);
        js.executeScript("window.scrollTo(0,1000)");

    }

}

```

Q #13) Write a Java Program to open all links of gmail.com.

Answer: It is a typical example of advanced for loop which we have seen in our previous programs.

Once you have opened a website such as Gmail using get() or navigate().to(), you can use a tagName locator to find the tag name of a website that will return all the tags.

We have advanced for loop where we have created a new WebElement link2 for a link(which already has located all the tags), then we have got all the links through getAttribute("href") and got all the texts through getText().

```
package Codes;
```

```
import java.util.concurrent.TimeUnit;
```

```
import org.openqa.selenium.By;
```

```
import org.openqa.selenium.WebDriver;
```

```
import org.openqa.selenium.WebElement;
```

```

import org.openqa.selenium.chrome.ChromeDriver;

public class openAllLinks {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.setProperty("webdriver.chrome.driver", "C:\\webdriver\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
        driver.manage().window().maximize();
        driver.get("https://www.gmail.com/");
        java.util.List<WebElement> link = driver.findElements(By.tagName("a"));
        System.out.println(link.size());

        for (WebElement link2: link) {

            //print the links i.e. http://google.com or https://www.gmail.com
            System.out.println(link2.getAttribute("href"));

            //print the links text
            System.out.println(link2.getText());
        }
    }
}

```

Output:

Starting ChromeDriver 2.38.551601 (edb21f07fc70e9027c746edd3201443e011a61ed) on port 16163

Only local connections are allowed.

4

<https://support.google.com/chrome/answer/6130773?hl=en-GB>

Learn more

<https://support.google.com/accounts?hl=en-GB>

Help

<https://accounts.google.com/TOS?loc=IN&hl=en-GB&privacy=true>

Privacy

<https://accounts.google.com/TOS?loc=IN&hl=en-GB>

Terms

Q #14) Write a Selenium code to switch to the previous tab.

Answer: We have demonstrated the use of the Robot class. We see this as an important third party because we can achieve the different navigation within a browser and its tabs if you know the shortcut keys.

For example, if you have three tabs open in your chrome and you want to go to the middle tab, then you have to press control + 2 from your keyboard. The same thing can be achieved through the code as well.

Observe the following code (just after we see the instantiation of Robot class). we have used the Robot class object called a robot with two inbuilt methods `keyPress(KeyEvent.VK_*)` and `keyRelease(KeyEvent.VK_*)`.

```
package Codes;

import java.awt.AWTException;
import java.awt.Robot;
import java.awt.event.KeyEvent;
import java.util.concurrent.TimeUnit;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class PreviousTab {
```

```

public static void main(String[] args) throws AWTEException {
    // TODO Auto-generated method stub
    System.setProperty("webdriver.chrome.driver", "C:\\\\webdriver\\\\chromedriver.exe");
    WebDriver driver = new ChromeDriver();
    driver.manage().window().maximize();
    driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
    driver.get("https://www.google.com");
    WebElement element1 = driver.findElement(By.name("q"));
    element1.sendKeys("software testing help");
    element1.sendKeys(Keys.ENTER);
    String a = Keys.chord(Keys.CONTROL,Keys.RETURN);
    driver.findElement(By.partialLinkText("Software Testing Help - A Must Visit Software Testing
Portal")).sendKeys(a);
    Robot robot = new Robot(); // instantiated robot class
    robot.keyPress(KeyEvent.VK_CONTROL); // with robot class you can easily achieve anything if
you know the shortcut keys
    robot.keyPress(KeyEvent.VK_2); // here, we have just pressed ctrl+2
    robot.keyRelease(KeyEvent.VK_CONTROL); // once we press and release ctrl+2, it will go to
the second tab.
    robot.keyRelease(KeyEvent.VK_2); //if you again want to go back to first tab press and release
vk_1
}
}

```

Q #15) Write a Java Program to find the duplicate characters in a string.

Answer: In this program, we have created a string variable str and initialized an integer count with zero.

Then, we have created a character array to convert our string variable to the character. With the help of for loop, we are performing a comparison between different characters at different indexes.

If two characters of consecutive index match, then it will print that character and the counter will be incremented by 1 after each iteration.

```

public class DuplicateCharacters {

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        String str = new String("Sakkett");

        int count = 0;

        char[] chars = str.toCharArray();

        System.out.println("Duplicate characters are:");

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

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

                if (chars[i] == chars[j]) {

                    System.out.println(chars[j]);

                    count++;

                    break;

                }

            }

        }

    }

}

```

Output:

Duplicate characters are:

k

t

Q #16) Write a Java Program to find the second-highest number in an array.

Answer: In this program, we have initialized an array with 10 random elements out of which we are going to find the second-highest number. Here, we have two integers- the largest and second-largest. Both set to the first index of the element. Then, we have printed all the elements using for loop.

Now, the logic is when the element at the 0th index is greater than the largest, then assign arr[0] to largest and secondLargest to largest. Again, if the element at the 0th index is greater than the secondLargest, then assign secondLargest to arr[0].

This will be repeated for each iteration and ultimately after comparing or completing iterations up to array length will give you the secondLargest element.

```
package codes;

public class SecondHighestNumberInArray {

public static void main(String[] args)

{

    int arr[] = { 100,14, 46, 47, 94, 94, 52, 86, 36, 94, 89 };

    int largest = 0;

    int secondLargest = 0;

    System.out.println("The given array is:");

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

    {

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

    }

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

    {

        if (arr[i] > largest)

        {

            secondLargest = largest;

            largest = arr[i];

        }

        else if (arr[i] > secondLargest)

        {

            secondLargest = arr[i];

        }

    }

}
```



```

        System.out.println("\nSecond largest number is:" + secondLargest);
        System.out.println("Largest Number is: " + largest);
    }
}

```

Output:

The given array is:

100 14 46 47 94 94 52 86 36 94 89

Second largest number is:94

Largest Number is: 100

Q #17) Write a Java Program to check Armstrong number.

Answer: First of all we need to understand what Armstrong Number is. Armstrong number is the number which is the sum of the cubes of all its unit, tens and hundred digits for three-digit numbers.

$$153 = 1*1*1 + 5*5*5 + 3*3*3 = 1 + 125 + 27 = 153$$

If you have a four-digit number lets say

$$1634 = 1*1*1*1 + 6*6*6*6 + 3*3*3*3 + 4*4*4*4 = 1 + 1296 + 81 + 256 = 1634$$

Now, in this program, we have a temp and integers declared. We have initialized c with value 0. Then, we need to assign the integer value which we are going to check for Armstrong (in our case, let us say 153). Then we have assigned our temp variable with that number which we are going to check.

Thereafter, we have used while conditional check where the remainder is assigned to a and the number is divided by 10 and assigned to n. Now, our c variable which was set to zero initially is assigned with $c+(a*a*a)$. Suppose we have to evaluate a four-digit number then c should be assigned with $c + (a*a*a*a)$.

Lastly, we have put an if-else statement for conditional checking where we have compared the value contained in c against temp(which has the actual number stored at this point). If it matches, then the number is Armstrong otherwise not.

```

class Armstrong{
public static void main(String[] args) {
    int c=0,a,temp;
    int n=153;//It is the number to check Armstrong
    temp=n;
    while(n>0)
    {
        a=n%10;
        n=n/10;
        c=c+(a*a*a);
    }
    if(temp==c)
        System.out.println("armstrong number");
    else
        System.out.println("Not armstrong number");
    }
}

```

Output:

armstrong number

Q #18) Write a Java Program to remove all white spaces from a string with using replace().

Answer: This is a simple program where we have our string variable str1.

Another string variable str2 is initialized with the replaceAll option which is an inbuilt method to remove n number of whitespaces. Ultimately, we have printed str2 which has no whitespaces.

```

class RemoveWhiteSpaces
{
    public static void main(String[] args)

```

```

{
    String str1 = "Saket Saurav    is a QualityAna list";

    //1. Using replaceAll() Method

    String str2 = str1.replaceAll("\\s", "");

    System.out.println(str2);

}
}
}

```

Output:

SaketSauravisaQualityAnalist

Q #19) Write a Java Program to remove all white spaces from a string without using replace().

Answer: This is another approach to removing all the white spaces. Again, we have one string variable str1 with some value. Then, we have converted that string into a character array using toCharArray().

Then, we have one StringBuffer object sb which will be used to append the value stored at chars[i] index after we have included for loop and one if condition.

If the condition is set such that then the element at i index of the character array should not be equal to space or tab. Finally, we have printed our StringBuffer object sb.

```

class RemoveWhiteSpaces
{
    public static void main(String[] args)
    {

```

```

String str1 = "Saket Saurav    is an Autom ation Engi ne    er";

char[] chars = str1.toCharArray();

StringBuffer sb = new StringBuffer();

for (int i = 0; i < chars.length; i++)
{
    if( (chars[i] != ' ') && (chars[i] != '\t') )
    {
        sb.append(chars[i]);
    }
}

System.out.println(sb);    //Output : CoreJavajspervletsjdbcstrutshibernatespring
}
}

```

Output:

SaketSauravisanAutomationEngineer

Q #20) Write a Java Program to read an excel.

Answer: These types of programs are generally used in Selenium framework. We have added detailed comments for every step to make the program more understandable.

The logic starts after we have loaded the sheet in which the data is stored. We are trying to import email and password. For this, we are retrieving the cell using `getRow()` and `getCell()` method. Let's say we have email and passwords at the 1st and 2nd cell.

Then we are setting the type of cell to string. Thereafter we are carrying out a normal web element locator operation (By.id) where we have passed unique locator values such as "email" and "password" which will identify these elements.

Finally, we are sending keys using `element.sendKeys` where `cell.getStringCellValue()` is the key. This will return you the value stored at cell number 1 and 2 respectively.

@Test

```
public void ReadData() throws IOException
{
    // Import excel sheet from a webdriver directory which is inside c drive.
    //DataSource is the name of the excel
    File src=new File("C:\\webdriver\\DataSource.xls");

    //This step is for loading the file. We have used FileInputStream as
    //we are reading the excel. In case you want to write into the file,
    //you need to use FileOutputStream. The path of the file is passed as an argument to
    FileInputStream

    FileInputStream finput = new FileInputStream(src);

    //This step is to load the workbook of the excel which is done by global HSSFWorkbook in which
    we have
    //passed finput as an argument.
    workbook = new HSSFWorkbook(finput);

    //This step is to load the sheet in which data is stored.
    sheet= workbook.getSheetAt(0);

    for(int i=1; i<=sheet.getLastRowNum(); i++)
    {
        // Import data for Email.
        cell = sheet.getRow(i).getCell(1);
        cell.setCellType(Cell.CELL_TYPE_STRING);
        driver.findElement(By.id("email")).sendKeys(cell.getStringCellValue());

        // Import data for the password.
```

```
cell = sheet.getRow(i).getCell(2);  
cell.setCellType(Cell.CELL_TYPE_STRING);  
driver.findElement(By.id("password")).sendKeys(cell.getStringCellValue());  
  
}  
}
```