**PROGRAM**

**BoothsMultip.java:**

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

@WebServlet(urlPatterns = {"/BoothMultiplication"})

public class BoothMultiplication extends HttpServlet {

protected void processRequest(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

int i;

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

try {

out.println("<html>");

out.println("<head>");

out.println("<title>Servlet Calculator</title>");

out.println("</head>");

out.println("<body>");

String num1 = request.getParameter("no1");

String num2 = request.getParameter("no2");

int a = Integer.parseInt(num1);

int b = Integer.parseInt(num2);

if (a > 15 || b > 15) {

out.println("Please enter numbers less than 15");

}

else

{

int tempnum1 = a;

int tempnum2 = b;

int num1arr[] = binaryConversion(a); //Converting Decimal to Binary

int temp1arr[] = new int[5];

int temp2arr[] = new int[5];

for (i = 0; i <= 4; i++) {

temp1arr[i] = num1arr[i];

}

int num2arr[] = binaryConversion(b); //Converting Decimal to Binary

for (i = 0; i <= 4; i++) {

temp2arr[i] = num2arr[i];

}

int tocomp[] = {0, 0, 0, 0, 1};

int comp1arr[] = binaryComplement(num1arr);

int twos1arr[] = binaryAddition(comp1arr, tocomp); //taking 2's Complement

int comp2arr[] = binaryComplement(num2arr);

int twos2arr[] = binaryAddition(comp2arr, tocomp); //taking 2's Complement

int binaryResult[] = boothsAlgo(temp1arr, temp2arr, twos2arr);

if ((tempnum1 < 0) && (tempnum2 > 0) || (tempnum1 > 0) && (tempnum2 < 0)) {

int compResult[] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 1};

int tempResult[] = binarytenComplement(binaryResult);

int twosResult[] = binarytenAddition(tempResult, compResult);

for (i = 0; i <= 9; i++) {

binaryResult[i] = twosResult[i];

}

int deciResult = convDecimal(binaryResult);

out.println("Result in Decimal form is -" + deciResult);

} else {

int deciResult = convDecimal(binaryResult);

out.println("<h1>Result in Decimal form is " + deciResult + "</h1>");

}

out.println("</body>");

out.println("</html>");

}

} finally {

out.close();

}

}

public static int convDecimal(int n[]) //Method for converting Binary to decimal

{

int deci = 0;

int j, k;

for (int i = 9; i >= 0; i--) {

k = 1;

for (j = i; j <= 9; j++) {

k = k \* 2;

}

deci = deci + n[i] \* (k / 2);

}

return deci;

}

public static int[] binaryConversion(int num) //Method for converting Decimal to Binary

{

int i, temp, store;

store = num;

int binaryNum[] = {0, 0, 0, 0, 0};

if (num < 0) {

num = num \* (-1);

}

for (i = 4; i >= 1; i--) {

temp = num % 2;

num = num / 2;

binaryNum[i] = temp;

}

if (store < 0) {

int storecomp[] = binaryComplement(binaryNum);

int addone[] = {0, 0, 0, 0, 1};

int storeadd[] = binaryAddition(storecomp, addone);

for (i = 0; i < 5; i++) {

binaryNum[i] = storeadd[i];

}

}

return binaryNum;

}

public static int[] binaryComplement(int comp[]) //Method for taking complement 5 Bits

{

for (int i = 4; i >= 0; i--) {

if (comp[i] == 0) {

comp[i] = 1;

} else {

comp[i] = 0;

}

}

return comp;

}

public static int[] binarytenComplement(int comp[]) //Method for taking complement 10 Bits

{

for (int i = 9; i >= 0; i--) {

if (comp[i] == 0) {

comp[i] = 1;

} else {

comp[i] = 0;

}

}

return comp;

}

public static int[] binaryAddition(int add1[], int add2[]) //Method for addition of array 5 Bits

{

int i, carry = 0;

int add3[] = {0, 0, 0, 0, 0};

for (i = 4; i >= 0; i--) {

if (add1[i] == 1 && add2[i] == 1 && carry == 0) {

add3[i] = 0;

carry = 1;

} else if (add1[i] == 1 && add2[i] == 1 && carry == 1) {

add3[i] = 1;

carry = 1;

} else if (add1[i] == 1 && add2[i] == 0 && carry == 0) {

add3[i] = 1;

carry = 0;

} else if (add1[i] == 1 && add2[i] == 0 && carry == 1) {

add3[i] = 0;

carry = 1;

} else if (add1[i] == 0 && add2[i] == 1 && carry == 0) {

add3[i] = 1;

carry = 0;

} else if (add1[i] == 0 && add2[i] == 1 && carry == 1) {

add3[i] = 0;

carry = 1;

} else if (add1[i] == 0 && add2[i] == 0 && carry == 0) {

add3[i] = 0;

carry = 0;

} else if (add1[i] == 0 && add2[i] == 0 && carry == 1) {

add3[i] = 1;

carry = 0;

}

}

return add3;

}

public static int[] binarytenAddition(int add1[], int add2[]) //Method for addition of array 10 Bits

{

int i, carry = 0;

int add3[] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

for (i = 9; i >= 0; i--) {

if (add1[i] == 1 && add2[i] == 1 && carry == 0) {

add3[i] = 0;

carry = 1;

} else if (add1[i] == 1 && add2[i] == 1 && carry == 1) {

add3[i] = 1;

carry = 1;

} else if (add1[i] == 1 && add2[i] == 0 && carry == 0) {

add3[i] = 1;

carry = 0;

} else if (add1[i] == 1 && add2[i] == 0 && carry == 1) {

add3[i] = 0;

carry = 1;

} else if (add1[i] == 0 && add2[i] == 1 && carry == 0) {

add3[i] = 1;

carry = 0;

} else if (add1[i] == 0 && add2[i] == 1 && carry == 1) {

add3[i] = 0;

carry = 1;

} else if (add1[i] == 0 && add2[i] == 0 && carry == 0) {

add3[i] = 0;

carry = 0;

} else if (add1[i] == 0 && add2[i] == 0 && carry == 1) {

add3[i] = 1;

carry = 0;

}

}

return add3;

}

public static int[] boothsAlgo(int X[], int addY[], int subY[]) //Method for Booths Algorithm

{

int i, x1 = 0;

int U[] = {0, 0, 0, 0, 0};

System.out.println();

System.out.println(" U\t X\tX-1");

for (i = 0; i <= 4; i++) {

System.out.print(U[i]);

}

System.out.print("\t");

for (i = 0; i <= 4; i++) {

System.out.print(X[i]);

}

System.out.print("\t ");

System.out.print(x1);

for (int j = 0; j <= 4; j++) {

if ((X[4] == 0) && (x1 == 1)) {

int tempU[] = binaryAddition(U, addY);

for (i = 0; i <= 4; i++) {

U[i] = tempU[i];

}

x1 = X[4];

X[4] = X[3];

X[3] = X[2];

X[2] = X[1];

X[1] = X[0];

X[0] = U[4];

U[4] = U[3];

U[3] = U[2];

U[2] = U[1];

U[1] = U[0];

System.out.println();

for (i = 0; i <= 4; i++) {

System.out.print(U[i]);

}

System.out.print("\t");

for (i = 0; i <= 4; i++) {

System.out.print(X[i]);

}

System.out.print("\t ");

System.out.print(x1);

} else if ((X[4] == 1) && (x1 == 0)) {

int tempU[] = binaryAddition(U, subY);

for (i = 0; i <= 4; i++) {

U[i] = tempU[i];

}

x1 = X[4];

X[4] = X[3];

X[3] = X[2];

X[2] = X[1];

X[1] = X[0];

X[0] = U[4];

U[4] = U[3];

U[3] = U[2];

U[2] = U[1];

U[1] = U[0];

System.out.println();

for (i = 0; i <= 4; i++) {

System.out.print(U[i]);

}

System.out.print("\t");

for (i = 0; i <= 4; i++) {

System.out.print(X[i]);

}

System.out.print("\t ");

System.out.print(x1);

} else {

x1 = X[4];

X[4] = X[3];

X[3] = X[2];

X[2] = X[1];

X[1] = X[0];

X[0] = U[4];

U[4] = U[3];

U[3] = U[2];

U[2] = U[1];

U[1] = U[0];

System.out.println();

for (i = 0; i <= 4; i++) {

System.out.print(U[i]);

}

System.out.print("\t");

for (i = 0; i <= 4; i++) {

System.out.print(X[i]);

}

System.out.print("\t ");

System.out.print(x1);

}

}

int finalArr[] = new int[10]; //Shifting Result in finalArr

finalArr[9] = X[4];

finalArr[8] = X[3];

finalArr[7] = X[2];

finalArr[6] = X[1];

finalArr[5] = X[0];

finalArr[4] = U[4];

finalArr[3] = U[3];

finalArr[2] = U[2];

finalArr[1] = U[1];

finalArr[0] = U[0];

return finalArr;

}

@Override

protected void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

protected void doPost(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

processRequest(request, response);

}

@Override

public String getServletInfo() {

return "Short description";

}// </editor-fold>

}

**Mytest1.java:-**

package mytest1;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.firefox.FirefoxDriver;

public class Mytest1 {

public static void main(String[] args) {

WebDriver driver = new FirefoxDriver();

String baseUrl = "http://localhost:8080/BoothsMultiplier/";

driver.get(baseUrl);

String expected = "JSP Page";

String actual = "";

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

actual = driver.getTitle();

if (actual.equals(expected)) {

System.out.println("Title test passed");

} else {

System.out.println("Title test failed");}

WebElement text=driver.findElement(By.name("no1"));

text.sendKeys("5");

WebElement text1=driver.findElement(By.name("no2"));

text1.sendKeys("2");

WebElement btn=driver.findElement(By.name("btn"));

btn.click();

System.out.println(" test script sucessful");

driver.close();

} }

**.jsp:-**

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>JSP Page</title>

</head>

<body>

<h1>Booths Multiplier Application</h1>

<form method="Post" action="BoothMultiplication">

Enter number 1

<input type="text" name="no1" value=""/><br><br>

Enter number 2

<input type="text" name="no2" value="" /><br><br>

<input type="submit" value="Display Answer"/>

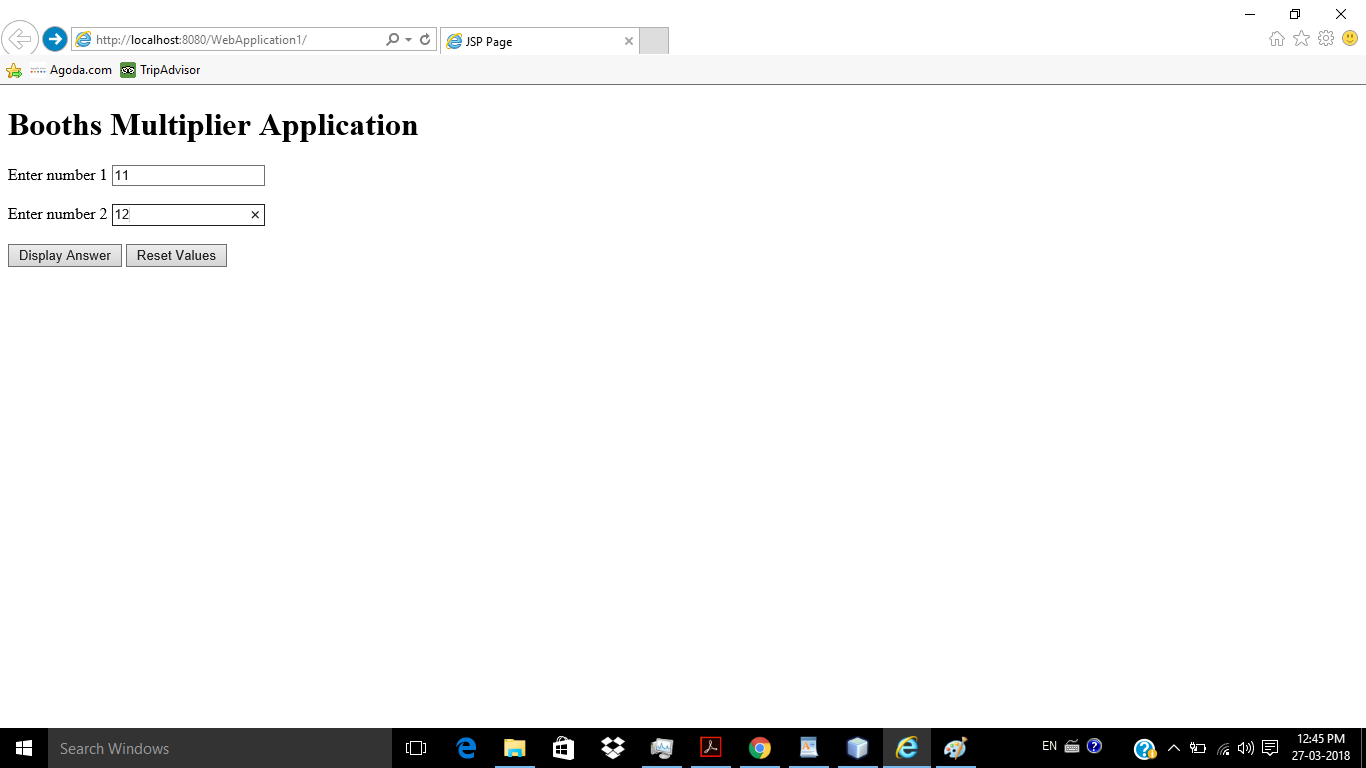
<input type="reset" value="Reset Values"/>

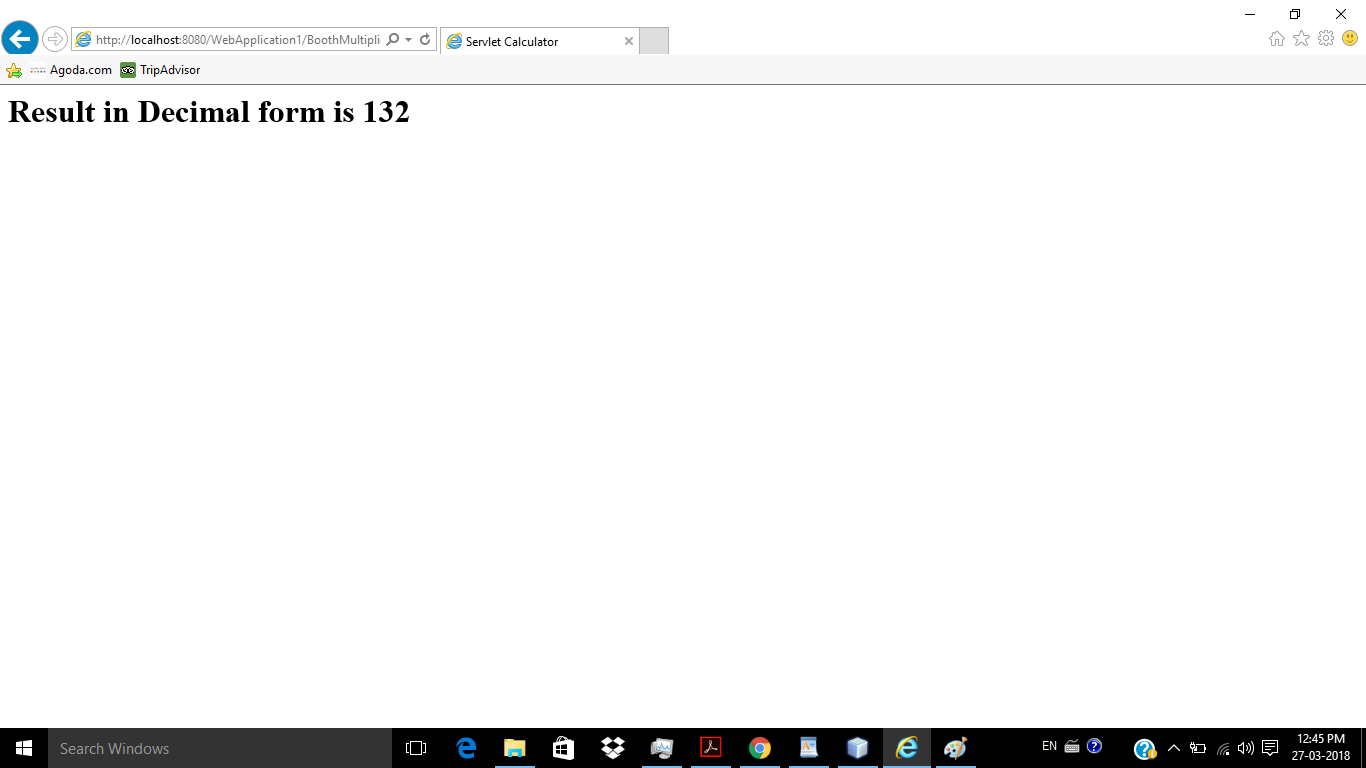
</form>

</body>

</html>

**OUTPUT**

****

****