

Index.html

```
<!DOCTYPE html>
```

```
<!--
```

To change this license header, choose License Headers in Project Properties.

To change this template file, choose Tools | Templates

and open the template in the editor.

```
-->
```

```
<html>
```

```
  <head>
```

```
    <title> Booths Algorithm </title>
```

```
    <meta charset="UTF-8">
```

```
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  </head>
```

```
  <body>
```

```
    <h1>Booths Algorithm</h1>
```

```
    <center><form name="Input" action="Booths" method="post" >
```

```
    Multiplicand: <input type="text" name="multiplicand" size="30" />
```

```
    <br><br>
```

```
    Multiplier: <input type="text" name="multiplier" size="30" />
```

```
    <br><br>
```

```
    <input type="submit" value="Submit" /><br><br>
```

```
  </form>
```

```
  </center>
```

```
  </body>
```

```
</html>
```

Booths.java

```
/*
```

```
 * To change this license header, choose License Headers in Project Properties.
```

```
 * To change this template file, choose Tools | Templates
```

```
 * and open the template in the editor.
```

```
*/
```

```
import java.io.IOException;
```

```
import java.io.PrintWriter;
```

```
import javax.servlet.ServletConfig;
```

```
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 = {"/Booths"})
```

```
public class Booths extends HttpServlet
```

```
{
```

```
  public int multiply(int n1, int n2,PrintWriter writer)
```

```
  {
```

```
    int q=0;
```

```
int i,j,a,b,temp;
int[] A={0,0,0,0,0,0,0,0},C={0,0,0,0,0,0,0,1},C1={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1};
int s=0,z=0;
int [] Q=new int[8];
int [] M=new int[8];
int [] temp1=new int[8];
int [] ans=new int[16];
int y,x=0,c=0;
a=n1;
b=n2;
binary(a,M,writer);
binary(b,Q,writer);
writer.println("<br><br>-----<br>");
writer.println("Operation &nbsp&nbsp&nbsp A"
+"&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp Q"
+"&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp Q"
+"&nbsp&nbsp&nbsp&nbsp&nbsp&nbsp M");
writer.println("<br><br> INITIAL &nbsp&nbsp&nbsp");
for(i=0;i<8;i++)
{
writer.println(A[i]);
}
writer.println("&nbsp&nbsp");
for(i=0;i<8;i++){
writer.println(Q[i]);
}
writer.println("&nbsp&nbsp");
writer.println(q + "&nbsp&nbsp");
for(i=0;i<8;i++)
{
writer.println(M[i]);
}
for(j=0;j<8;j++)
{
if((Q[7]==0)&&(q==1))
{
writer.println("<br> A=A+M &nbsp&nbsp&nbsp");
add(A,M);
for(i=0;i<8;i++)
{
writer.println(A[i]);
}
writer.println("&nbsp&nbsp");
for(i=0;i<8;i++)
{
writer.println(Q[i]);
}
writer.println("&nbsp&nbsp"+q + "&nbsp&nbsp");
for(i=0;i<8;i++)
```

[illegible]

```

for(i=0;i<8;i++)
{
ans[i]=A[i];
}
for(i=0;i<8;i++)
{
ans[i+8]=Q[i];
}
for(i=0;i<16;i++)
{
writer.println(ans[i]);
}
if(((a < 0)&&(b > 0))||((a > 0)&&(b < 0)))
{
    for(i=0;i<16;i++)
    {
        ans[i]=1-ans[i];
    }
    for(i=15;i>=0;i--)
    {
        x=ans[i];
        ans[i]=c^x^C1[i];
        if(((c==1)&&(x==1))||((x==1)&&(C1[i]==1))||((C1[i]==1)&&(c==1)))
        {
            c=1;
        }
        else
        {
            c=0;
        }
    }
} //for end
} // end if

for(i=15;i>0;i--)
{
s=s+((int)Math.pow(2,z)*ans[i]);
z=z+1;
if(((a < 0)&&(b > 0))||((a > 0)&&(b < 0)))
{
    s=s*-1;
}
}

return s;
//}
}
public static void rshift(int x,int y[])
{
    int i;for(i=7;i>0;i--)

```

```

{
y[i]=y[i-1];
}
y[0]=x;
}
public static void add(int a[], int b[])
{
int x,i,c=0;
for(i=7;i>=0;i--)
{
x=a[i];
a[i]=c^x^b[i];
if(((c==1)&&(x==1)) || ((x==1)&&(b[i]==1)) || ((b[i]==1)&&(c==1)))
{
c=1;
}
else
{
c=0;
}
}
}
public static void binary(int x, int arr[],PrintWriter writer)
{
int i,p=x;
int[]c = {0,0,0,0,0,0,0,1};
for(i=0;i<8;i++)
{
arr[i]=0;
}
if(x<0)
{
x=x*-1;
}
i=7;
do
{
arr[i] = x%2;x = x/2;
i--;
}while(x!=0);
if(p<0)
{
for(i=0;i<8;i++)
{
arr[i]=1-arr[i];
}
add(arr,c);
}
writer.println("<br><br>THE BINARY EQUIVALENT OF "+p+" IS : ");

```

```

for(i=0;i<8;i++)
{
writer.println(arr[i]);
}
}
public void display(int [] P, char ch, PrintWriter writer)
{
writer.println("<br>" + ch + ":");
for(int i=0;i<P.length;i++)
{
if(i==4)
{
writer.println(" &nbsp; &nbsp;");
}
if(i==8)
{
writer.println(" &nbsp; &nbsp;");
}
writer.println(P[i]);
}
}
/**
 *
 * @author ccoew
 */

/**
 * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
 * methods.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    try (PrintWriter out = response.getWriter()) {
        /* TODO output your page here. You may use following sample code. */
        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet BoothServlet</title>");
        out.println("</head>");
        out.println("<body>");
        out.println("<h1>Servlet BoothServlet at " + request.getContextPath() + "</h1>");
        out.println("</body>");
    }
}

```

```

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

}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to
edit the code.">
/**
 * Handles the HTTP <code>GET</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
public void init(ServletConfig config)
{
System.out.println("Servlet is being initialized");
}

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {

    processRequest(request, response);
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    String paramMp = request.getParameter("multiplicand");
int multiplicand = Integer.parseInt(paramMp);
String paramMr = request.getParameter("multiplier");int multiplier = Integer.parseInt(paramMr);
PrintWriter writer = response.getWriter();
writer.println("<html>");
Booths b = new Booths();
int result = b.multiply(multiplicand,multiplier,writer);
writer.println("<br><br> Product is:" +result);
writer.println("<html>");
writer.flush();
    // processRequest(request, response);
}

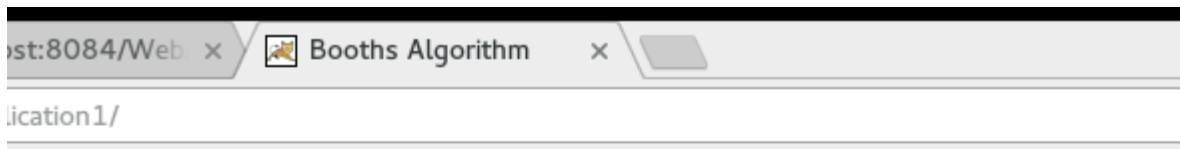
```

```
public void destroy()
{
    System.out.println("Servlet is being destroyed");
}

/**
 * Returns a short description of the servlet.
 *
 * @return a String containing servlet description
 */
@Override
public String getServletInfo() {
    return "Short description";
} // </editor-fold>

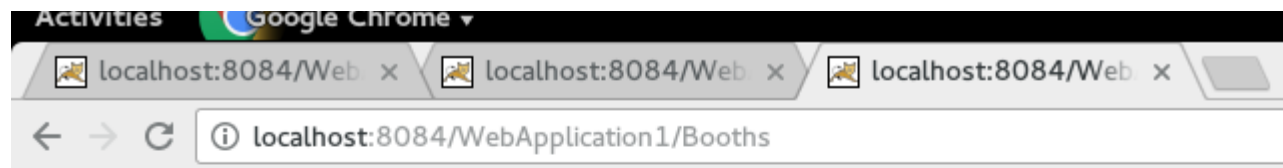
}
```

//OUTPUT



Multiplicand:

Multiplier:



THE BINARY EQUIVALENT OF -16 IS : 1 1 1 1 0 0 0 0

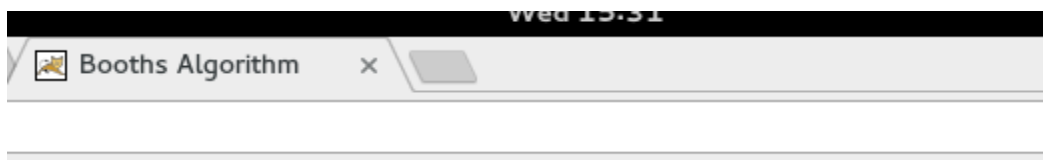
THE BINARY EQUIVALENT OF 5 IS : 0 0 0 0 0 1 0 1

```
-----
Operation  A    Q    Q'    M

INITIAL    00000000 00000101 0 11110000
A=A-M      00010000 00000101 0 11110000
SHIFT      00001000 00000010 1 11110000
A=A+M      11111000 00000010 1 11110000
SHIFT      11111100 00000001 0 11110000
A=A-M      00001100 00000001 0 11110000
SHIFT      00000110 00000000 1 11110000
A=A+M      11110110 00000000 1 11110000
SHIFT      11111011 00000000 0 11110000
SHIFT      11111101 10000000 0 11110000
SHIFT      11111110 11000000 0 11110000
SHIFT      11111111 01100000 0 11110000
SHIFT      11111111 10110000 0 11110000
```

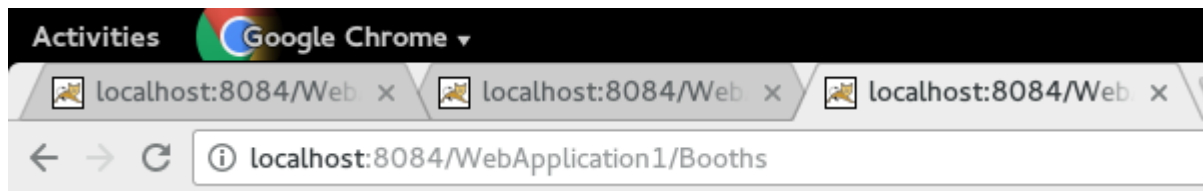
THE ANSWER IN BINARY IS: 1 1 1 1 1 1 1 1 0 1 1 0 0 0 0

Product is:-80



Multiplicand:

Multiplier:



THE BINARY EQUIVALENT OF -16 IS : 1 1 1 1 0 0 0 0

THE BINARY EQUIVALENT OF -5 IS : 1 1 1 1 0 1 1

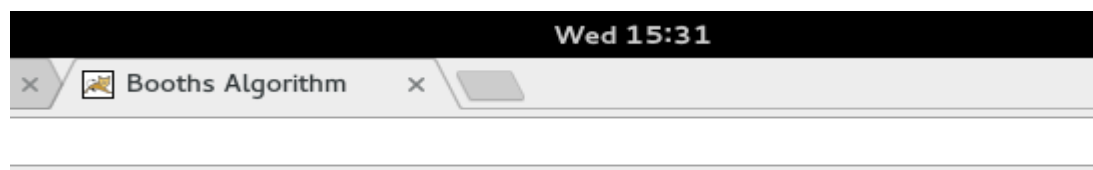
```

-----
Operation  A    Q    Q'    M

INITIAL    00000000  11111011  0 11110000
A=A-M      00010000  11111011  0 11110000
SHIFT      00001000  01111101  1 11110000
SHIFT      00000100  00111110  1 11110000
A=A+M      11110100  00111110  1 11110000
SHIFT      11111010  00011111  0 11110000
A=A-M      00001010  00011111  0 11110000
SHIFT      00000101  00001111  1 11110000
SHIFT      00000010  10000111  1 11110000
SHIFT      00000001  01000011  1 11110000
SHIFT      00000000  10100001  1 11110000
SHIFT      00000000  01010000  1 11110000
  
```

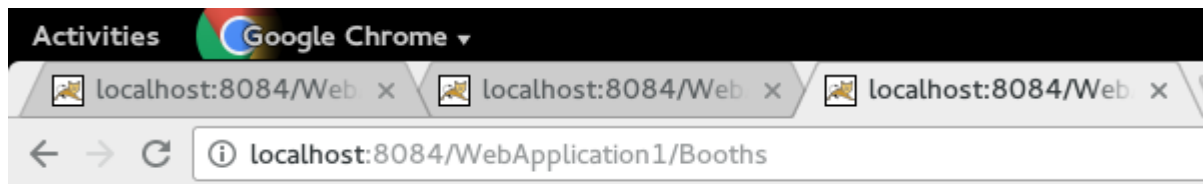
THE ANSWER IN BINARY IS: 0000000001010000

Product is:80



Multiplicand:

Multiplier:



THE BINARY EQUIVALENT OF 16 IS : 0 0 0 1 0 0 0 0

THE BINARY EQUIVALENT OF 5 IS : 0 0 0 0 0 1 0 1

```

-----
Operation  A    Q    Q'    M
INITIAL    00000000 00000101 0 00010000
A=A-M      11110000 00000101 0 00010000
SHIFT      11111000 00000010 1 00010000
A=A+M      00001000 00000010 1 00010000
SHIFT      00000100 00000001 0 00010000
A=A-M      11110100 00000001 0 00010000
SHIFT      11111010 00000000 1 00010000
A=A+M      00001010 00000000 1 00010000
SHIFT      00000101 00000000 0 00010000
SHIFT      00000010 10000000 0 00010000
SHIFT      00000001 01000000 0 00010000
SHIFT      00000000 10100000 0 00010000
SHIFT      00000000 01010000 0 00010000
  
```

THE ANSWER IN BINARY IS: 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0

Product is: 80