Part A Code

import java.io.\*;  
public class Proj2A  
{  
public static void main(String args[])  
{  
char F = 'F';  
char T = 'T';  
System.out.println("(~q ^ r ^ ~p) v (~(r v ~p))");  
System.out.println();  
System.out.println("x y z f");  
System.out.println("-------------");  
LProp(F, F, F);  
LProp(F, F, T);  
LProp(F, T, F);  
LProp(F, T, T);  
LProp(T, F, F);  
LProp(T, F, T);  
LProp(T, T, F);  
LProp(T, T, T);  
}  
  
public static char LProp(char p, char q, char r)  
{  
  
 char f = ORlogic(ANDlogic(ANDlogic(NOTlogic(p), NOTlogic(q)), r),   
 NOTlogic(ORlogic(NOTlogic(p), r)));  
   
 System.out.println(p + " " + q + " " + r + " " + f);  
   
 return 0;  
}  
  
public static char NOTlogic(char x)  
{  
  
 if(x == 'F')  
 {  
 x = 'T';  
 }  
 else  
 {  
 x = 'F';  
 }   
   
 return x;  
}  
  
  
public static char ORlogic(char a, char b)  
{  
char c;  
  
 if(a == 'F' && b == 'F')  
 {  
 c = 'F';  
 }  
 else  
 {  
 c = 'T';  
 }  
   
 return c;  
  
}  
  
  
public static char ANDlogic(char d, char e)  
{  
char f;  
  
 if(d == 'F' || e == 'F')  
 {  
 f = 'F';  
 }  
 else  
 {  
 f = 'T';  
 }  
   
 return f;  
  
}  
  
}

Part A Output

(~q ^ r ^ ~p) v (~(r v ~p))  
  
x y z f  
-------------  
F F F F  
F F T T  
F T F F  
F T T F  
T F F T  
T F T F  
T T F T  
T T T F

Part B

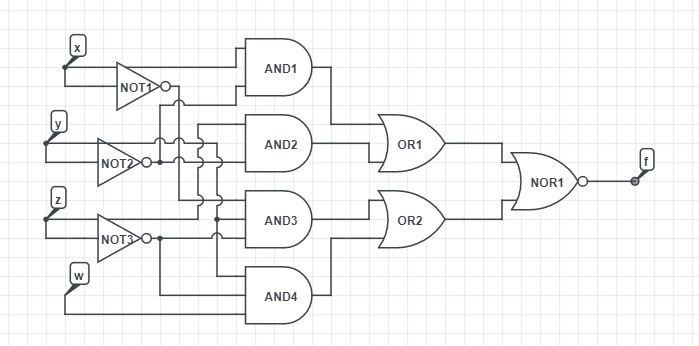
1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 00 | 01 | 11 | 10 |
| 00 | 0 | 0 | 1 | 1 |
| 01 | 1 | 1 | 0 | 0 |
| 11 | 0 | 1 | 0 | 0 |
| 10 | 1 | 1 | 1 | 1 |

2.

Y = zy’ + xy’ + x’yz’ + yz’w

3.



4.

import java.io.\*;  
public class Proj2A2  
{  
public static void main(String args[])  
{  
int F = 0;  
int T = 1;  
System.out.println("x y z w f");  
System.out.println("-------------");  
  
LCircuit(0,0,0,0);  
LCircuit(0,0,0,1);  
LCircuit(0,0,1,0);  
LCircuit(0,0,1,1);  
LCircuit(0,1,0,0);  
LCircuit(0,1,0,1);  
LCircuit(0,1,1,0);  
LCircuit(0,1,1,1);  
LCircuit(1,0,0,0);  
LCircuit(1,0,0,1);  
LCircuit(1,0,1,0);  
LCircuit(1,0,1,1);  
LCircuit(1,1,0,0);  
LCircuit(1,1,0,1);  
LCircuit(1,1,1,0);  
LCircuit(1,1,1,1);  
}  
  
public static char LCircuit(int x, int y, int z, int w)  
{  
  
 int f = ORgate(ORgate(ORgate(ANDgate(z,NOTgate(y)),  
 ANDgate(x,NOTgate(y))),  
 ANDgate(ANDgate(NOTgate(x),y),NOTgate(z))),  
 ANDgate(ANDgate(y,NOTgate(z)),w));  
  
   
   
 System.out.println(x + " " + y + " " + z + " " + w + " " + f);  
   
 return 0;  
}  
  
public static int NOTgate(int x)  
{  
  
 if(x == 0)  
 {  
 x = 1;  
 }  
 else  
 {  
 x = 0;  
 }   
   
 return x;  
}  
  
  
public static int ORgate(int a, int b)  
{  
  
  
 if(a == 0 && b == 0)  
 {  
 a = 0;  
 }  
 else  
 {  
 a = 1;  
 }  
   
 return a;  
   
}  
  
  
public static int ANDgate(int d, int e)  
{  
  
  
 if(d == 0 || e == 0)  
 {  
 d = 0;  
 }  
 else  
 {  
 d = 1;  
 }  
   
 return d;  
  
}  
  
}

Output

x y z w f  
-------------  
0 0 0 0 0  
0 0 0 1 0  
0 0 1 0 1  
0 0 1 1 1  
0 1 0 0 1  
0 1 0 1 1  
0 1 1 0 0  
0 1 1 1 0  
1 0 0 0 1  
1 0 0 1 1  
1 0 1 0 1  
1 0 1 1 1  
1 1 0 0 0  
1 1 0 1 1  
1 1 1 0 0  
1 1 1 1 0  


Part C

import java.io.\*;  
import java.util.Scanner;  
public class Proj2C  
{  
  
 public static void main(String args[])  
 {  
 Scanner in = new Scanner(System.in);  
   
 System.out.println("Enter N1: ");  
 int N1 = in.nextInt();  
   
 System.out.println("Enter N2: ");  
 int N2 = in.nextInt();  
   
   
   
 for(int i = N1; i <= N2;i++)  
 {  
 OddInt(i);  
 SquareInt(i);  
 SymmetricInt(i);  
 KNumber(i);  
   
 if(KNumber(i) == true)  
 {  
 System.out.println(i + " is a Knumber");  
 }  
 }  
 }  
  
  
 public static boolean OddInt(long x)  
 {  
 boolean logical;  
   
 if(x % 2 == 0)  
 {  
 logical = false;  
 }  
 else  
 {  
 logical = true;  
 }  
   
 return logical;  
   
 }  
   
   
 public static boolean SquareInt(long x)  
 {  
 boolean logical;  
   
 int sqrt = (int) Math.sqrt(x);  
 if(sqrt\*sqrt == x)   
 {  
 logical = true;  
 }  
 else  
 {  
 logical = false;  
 }  
   
 return logical;  
 }  
   
   
   
 public static boolean SymmetricInt(long x)  
 {  
 boolean logical;  
   
   
 long y = 0;  
 long z = x;  
   
 while( z != 0 )  
 {  
 y = y \* 10;  
 y = y + z%10;  
 z = z/10;  
 }  
   
 if(y == x)  
 {  
 logical = true;  
 }  
 else  
 {  
 logical= false;  
 }  
  
 return logical;  
   
 }  
   
   
 public static boolean KNumber(long x)  
 {  
 boolean logical;  
   
 if(OddInt(x) == true && SquareInt(x) == true && SymmetricInt(x) == true)  
 {  
 logical = true;  
 }  
 else  
 {  
 logical = false;  
 }  
   
 return logical;  
   
 }  
   
   
   
}

Output

1.

Enter N1:   
1  
Enter N2:   
1000  
1 is a Knumber  
9 is a Knumber  
121 is a Knumber

2.

Enter N1:   
1  
Enter N2:   
1000000  
1 is a Knumber  
9 is a Knumber  
121 is a Knumber  
10201 is a Knumber  
12321 is a Knumber  
14641 is a Knumber  
94249 is a Knumber

3.

1 is a Knumber  
9 is a Knumber  
121 is a Knumber  
10201 is a Knumber  
12321 is a Knumber  
14641 is a Knumber  
94249 is a Knumber  
1002001 is a Knumber  
1234321 is a Knumber  
5221225 is a Knumber