

Question 1

a)

$1 < 3$

```
ans = logical
      1
```

b)

$3 < 2$

```
ans = logical
      0
```

c)

$3 < 3$

```
ans = logical
      0
```

d)

$1 <= 3$

```
ans = logical
      1
```

e)

$3 <= 3$

```
ans = logical
      1
```

f)

$1 > 2$

```
ans = logical
      0
```

g)

```
2>=2
```

```
ans = logical  
1
```

h)

```
14~=15
```

```
ans = logical  
1
```

i)

```
14~=14
```

```
ans = logical  
0
```

question 2

```
x = 12 % in the command window
```

```
x =  
12
```

```
x == 13
```

```
ans = logical  
0
```

```
x==12
```

```
ans = logical  
1
```

```
x~= 13
```

```
ans = logical  
1
```

```
(x==12)|(x>3)
```

```
ans = logical  
1
```

```
(x==12)|(x<3)
```

```
ans = logical  
1
```

```
(x==12)&(x>3)
```

```
ans = logical  
1
```

```
~(x==12)|(x>3)
```

```
ans = logical
      1
```

```
~((x==12)|(x>3))
```

```
ans = logical
      0
```

question 3

```
a=1
```

```
a =
    1
```

```
b=true
```

```
b = logical
      1
```

```
c=0
```

```
c =
    0
```

```
f=a&b % 1 / true
```

```
f = logical
      1
```

```
g=a|b&c % 1 true
```

```
g = logical
      1
```

```
h=xor(~a,c)
```

```
h = logical
      0
```

```
v=[f,g,h]
```

```
v = 1x3 logical array
     1     1     0
```

```
% there will be no change if b is 1 because true and 1 are the same
```

Question 4 (bit strings)

```
a = [0 1 0 0 0 1 0 1]
```

```
a = 1x8
     0     1     0     0     0     1     0     1
```

```
b = [0 0 0 1 1 1 0 0]
```

```
b = 1x8
    0    0    0    1    1    1    0    0
```

```
c = a & b
```

```
c = 1x8 logical array
    0    0    0    0    0    1    0    0
```

```
d = a | b
```

```
d = 1x8 logical array
    0    1    0    1    1    1    0    1
```

```
r = xor(a,b)
```

```
r = 1x8 logical array
    0    1    0    1    1    0    0    1
```

```
f = ~a
```

```
f = 1x8 logical array
    1    0    1    1    1    0    1    0
```

question 5 (XOR Truth Table)

```
n = 2;
A = dec2bin((2^n)-1:-1:0) - '0';
fprintf("      A      B      xor(A,B)")
```

```
      A      B      xor(A,B)
```

```
for i = 1 : (2^n)
    A(i,3)=xor(A(i,1),A(i,2));
end
A
```

```
A = 4x3
    1    1    0
    1    0    1
    0    1    1
    0    0    0
```

```
dec2bin([76 79 72 73 84])
```

```
ans = 5x7 char array
'1001100'
'1001111'
'1001000'
'1001001'
'1010100'
```

Question 6

```
n = 2;
A = dec2bin((2^n)-1:-1:0) - '0';
fprintf("      A      B      A->B")
```

```
      A      B      A->B
```

```

for i = 1 : (2^n)
    X = A(i,1);
    Y = A(i,2);
    Z = ~X | Y;
    A(i,3) = Z;
end
A

```

```

A = 4x3
     1     1     1
     1     0     0
     0     1     1
     0     0     1

```

Question 7

```

n = 2;
A = dec2bin((2^n)-1:-1:0) - '0';
fprintf("      A      B      A<->B")

```

```

      A      B      A<->B

```

```

for i = 1 : (2^n)
    X = A(i,1);
    Y = A(i,2);
    Z = ~xor(X,Y);
    A(i,3) = Z;
end
A

```

```

A = 4x3
     1     1     1
     1     0     0
     0     1     0
     0     0     1

```

Question 8

```

% In tautology.m
n = input('Enter the number of propositions: ');
A= dec2bin(2^n-1:-1:0) - '0';
for i = 1 : (2^n)
    if ( A(i, 1)==1 &A(i, 2)==0 );
        A(i, 3)= 0;
    else
        A(i,3)=1;
    end
    A(i, 4)=~A(i, 1)|A(i, 2);
    if (A(i,3)==A(i, 4))
        A(i, 5)=1;
    else
        A(i,5)=0;
    end
end
ans=[A]

```

```

ans = 8x5

```

1	1	1	1	1
1	1	1	1	1
1	0	0	0	1
1	0	0	0	1
0	1	1	1	1
0	1	1	1	1
0	0	1	1	1
0	0	1	1	1

question 9

```
n = 3;
A= dec2bin(2^n-1:-1:0)-'0';
for i = 1 : (2^n)
    if ( A(i, 1)==1 &A(i, 2)==0 );
        A(i, 3)= 0;
    else
        A(i,3)=1;
    end
    A(i, 4)=~A(i, 1)|A(i, 2);
    if (A(i,3)==A(i, 4))
        A(i, 5)=1;
    else
        A(i,5)=0;
    end
end
ans=[A]
```

```
ans = 8x5
    1     1     1     1     1
    1     1     1     1     1
    1     0     0     0     1
    1     0     0     0     1
    0     1     1     1     1
    0     1     1     1     1
    0     0     1     1     1
    0     0     1     1     1
```

question 10

```
clear all;
n = input('Enter the number of propositions: ');
A= dec2bin(2^n-1:-1:0)-'0';
for i =1 : 2^n
    if A(i, 1)==1&A(i,2)==0
        A(i, 4)=0;
    else
        A(i, 4)=1;
    end
    if A(i, 1)==1&A(i,3)==0
        A(i,5)=0;
    else
        A(i,5)=1;
    end
end
```

```

end
if A(i, 4)==1& A(i,5)==1
    A(i,6)=1;
else
    A(i,6)=0;
end
if A(i, 2)==1&A(i, 3)==1
    A(i,7)=1;
else
    A(i,7)=0;
end
if A(i, 1)==1&A(i,7)==0
    A(i,8)=0;
    else
        A(i,8)=1;
end
end
end
ans=[A]

```

```

ans = 64×8
    1     1     1     1     1     1     1     1
    1     1     1     1     1     1     1     1
    1     1     1     1     1     1     1     1
    1     1     1     1     1     1     1     1
    1     1     1     1     1     1     1     1
    1     1     1     1     1     1     1     1
    1     1     1     1     1     1     1     1
    1     1     1     1     1     1     1     1
    1     1     0     1     0     0     0     0
    1     1     0     1     0     0     0     0
    ⋮
    ⋮

```

```

if A(1:2^n, 6)== A(1:2^n, 8)
    fprintf('yes, , the propositions are equivalent')
else
    fprintf('No, , the propositions are not equivalent')
end

```

yes, , the propositions are equivalent

q11

```

clear all;
n = input('Enter the number of propositions : ');
A= dec2bin(2^n-1:-1:0) - '0';
for i=1 : 2^n
    % 4th column is P->Q
    A(i,4)= (~A(i, 1))|A(i,2)

    % 5th col is P->R
    A(i,5)= (~A(i, 1))|A(i,3)

    % 6th col represent P->Q & P->R
    A(i,6)= A(i, 4)&A(i,5)

```

```

% 7th col represent Q + R
A(i,7)= A(i, 2)|A(i,3)

% 8th col represent P->(Q + R)
A(i,8)= (~A(i, 1))|A(i,7)

end

ans=[A]

if A(1:2^n, 6)== A(1:2^n , 8)
    fprintf('yes, , the propositions are equivalent')
else
    fprintf('No, , the propositions are not equivalent')
end

```

q12

```

clear all;
n = input('Enter the number of propositions : ');
A= dec2bin(2^n-1:-1:0) - '0';
for i=1 : 2^n
    A(i,4)= (A(i, 1))|A(i,2);
    A(i,5)= (~A(i, 1))|A(i,3);
    A(i,6)= (~A(i, 2))|A(i,3);
    A(i,7)= A(i, 4)&A(i,5)&A(i,6);
    A(i,8)= (~A(i, 7))|A(i,3);

end

ans=[A]

if A (1:2^n,8)==ones(2 ^n, 1)
    fprintf('Tautology')
else
    if A (1:2^n,8)==zeros(2^n, 1)
        fprintf('contradiction')
    else
        fprintf('contingency')
    end
end
end

```

Q13

1)

```
clear all;
```



```

n = input('Enter the number of propositions : ');
A= dec2bin(2^n-1:-1:0)-'0';
for i=1 : 2^n
A(i,4)= (~A(i, 1))|A(i,2);
A(i,5)= (~A(i, 4))|A(i,3);
A(i,6)= (~A(i, 2))|A(i,3);
A(i,7)= (~A(i, 1))|A(i,6);
end

ans=[A]

if A(1:2^n, 5)== A(1:2^n , 7)
    fprintf('yes, , the propositions are equivalent')
else
    fprintf('No, , the propositions are not equivalent')
end

```

2)

```

clear all;
n = input('Enter the number of propositions : ');
A= dec2bin(2^n-1:-1:0)-'0';
for i=1 : 2^n
A(i,4)= (A(i, 1))&A(i,2);
A(i,5)= (~A(i, 4))|A(i,3);
A(i,6)= (~A(i, 1))|A(i,3);
A(i,7)= (~A(i, 2))|A(i,3);
A(i,8)= A(i,6)&A(i,7);
end

ans=[A]

if A(1:2^n, 5)== A(1:2^n , 8)
    fprintf('yes, , the propositions are equivalent')
else
    fprintf('No, , the propositions are not equivalent')
end

```

3)

```

clear all;
n = input('Enter the number of propositions : ');
A= dec2bin(2^n-1:-1:0)-'0';
for i=1 : 2^n
A(i,5)= (~A(i, 1))|A(i,2);
A(i,6)= (~A(i, 3))|A(i,4);
A(i,7)= (~A(i, 5))|A(i,6);
A(i,8)= (~A(i, 1))|A(i,3);
A(i,9)= (~A(i, 2))|A(i,4);
A(i,10)= (~A(i, 8))|A(i,9);
end

ans=[A]

```

```
if A(1:2^n, 7)== A(1:2^n , 10)
    fprintf('yes, , the propositions are equivalent')
else
    fprintf('No, , the propositions are not equivalent')
end
```