User defined function

```
function verify_if_sqr_matrix(m)
    if(width(m) ~= height(m))
    error("please enter a square matrix")
    end
end
```

Get matrix

```
R = input("enter the relational matrix 'R':  ")
```

```
R = 3×3
     1     1     0
     0     1     1
     1     0     1
```

```
S = input("enter the relational matrix 'S':  ")
```

```
S = 3×3
     1     1     1
     1     1     1
     1     1     1
```

```

verify_if_sqr_matrix(R)
verify_if_sqr_matrix(S)

if(size(R) ~= size(S))
    error("Given Relational Matrices are of different sizes!")
end
```

Check reflexivity

```
function chk_reflexivity(M,name_of_matrix)
    is_reflexive = true;
    for i = 1:width(M)
        if(M(i,i) ~= 1)
            is_reflexive = false;
            break
        end
    end

    if(is_reflexive)
            fprintf("The given matrix %s is reflixive",name_of_matrix)
    else
            fprintf("The given matrix %s is not reflixive",name_of_matrix)
    end
end

chk_reflexivity(R,"R")
```

```
The given matrix R is reflixive
```

```
chk_reflexivity(S,"S")
```

```
The given matrix S is reflixive
```

Check symmetry

```
function chk_symmetry(M, name_of_matrix)
    is_symmetric = true;
    for i = 1:size(M, 1)
        for j = 1:size(M, 2)
            if M(i, j) ~= M(j, i)
                is_symmetric = false;
                break
            end
        end
    end

    if(is_symmetric)
        fprintf('The given matrix %s is symmetric.\n', name_of_matrix);
    else
        fprintf('The given matrix %s is asymmetric.\n', name_of_matrix);
    end
end

chk_symmetry(R, 'R');
```

```
The given matrix R is asymmetric.
```

```
chk_symmetry(S, 'S');
```

```
The given matrix S is symmetric.
```

Check transitivity

```
function chk_transitivity(M, name_of_matrix)
    is_transitive = true;
    for i = 1:size(M, 1)
        for j = 1:size(M, 2)
            for k = 1:size(M, 2)
                if (M(i, j) == 1 && M(j, k) == 1 && M(i, k) ~= 1)
                    is_transitive = false;
                    break;
                end
            end
        end
    end

    if(is_transitive)
```

```
        fprintf('The given matrix %s is transitive.\n', name_of_matrix);
    else
        fprintf('The given matrix %s is NOT transitive.\n', name_of_matrix);
    end
end

chk_transitivity(R,'R')
```

```
The given matrix R is NOT transitive.
```

```
chk_transitivity(S,'S')
```

```
The given matrix S is transitive.
```

R union S

```
R | S
```

```
ans = 3×3 logical array
   1   1   1
   1   1   1
   1   1   1
```

R intersection S

```
R & S
```

```
ans = 3×3 logical array
   1   1   0
   0   1   1
   1   0   1
```

R - S

```
R - (R&S)
```

```
ans = 3×3
     0     0     0
     0     0     0
     0     0     0
```

S - R

```
S - (S&R)
```

```
ans = 3×3
     0     0     1
     1     0     0
     0     1     0
```

R xor S

```
xor(R,S)
```

```
ans = 3×3 logical array
   0   0   1
   1   0   0
   0   1   0
```

R*R

```
R*R
```

```
ans = 3×3
     1     2     1
     1     1     2
     2     1     1
```

R*S

```
R*S
```

```
ans = 3×3
     2     2     2
     2     2     2
     2     2     2
```