In [1]:
```
using Graphs
using GraphPlot
using DataFrames
using CSV
using LinearAlgebra
```

Lucas Schmidt Ferreira de Araujo

Report 01

Exercice I

- Alice - 1, Bob - 2 , Gail - 3 , Irene - 4 , Carl - 5
- Harry - 6 , Jen - 7 , David - 8 , Ernest - 9 , Frank - 10

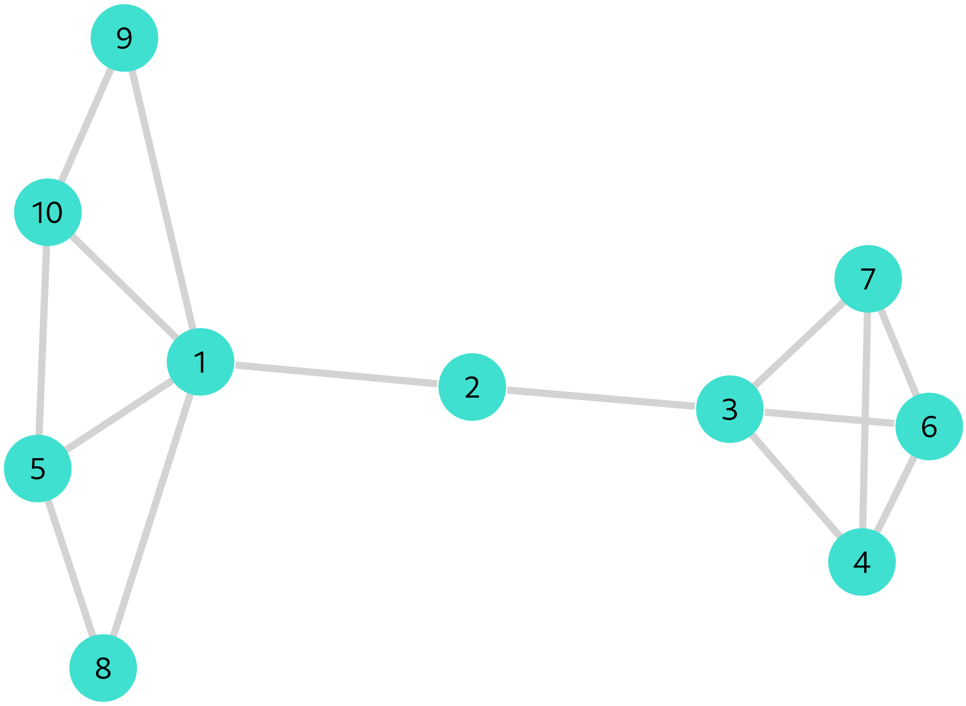
This is a graph with:

- Number of nodes: $n = 10$
- Number of edges: $m = 15$

In [2]:
```
n = 10
m = 15

G = Graph(n) # graph with 10 vertices
add_edge!(G, 1, 2)
add_edge!(G,2,3)
add_edge!(G,4,3)
add_edge!(G,5,1)
add_edge!(G,3,6)
add_edge!(G,4,7)
add_edge!(G,1,8)
add_edge!(G,6,7)
add_edge!(G,9,10)
add_edge!(G,1,9)
add_edge!(G,7,3)
add_edge!(G,8,5)
add_edge!(G,1,10)
add_edge!(G,6,4)
add_edge!(G,5,10)

gplot(G, nodelabel=1:10)
```



c) The Density of the network is given by the formula

$$\rho = \frac{m}{\binom{n}{2}} = \frac{2m}{n(n-1)}$$

In [3]:
```
ρ = 2*m/(n*(n-1))
println("The density ρ = $(ρ)")
```

```
The density ρ = 0.3333333333333333
```

d) The degree k_i of the node i is given by the number of edges conected to it

In [4]:
```
degrees = degree(G)
println("Node i | Degree ki")
for v in 1:n
    println("i = $v, ki = $(degrees[v])")
end
```

```
Node i | Degree ki
i = 1, ki = 5
i = 2, ki = 2
i = 3, ki = 4
i = 4, ki = 3
i = 5, ki = 3
i = 6, ki = 3
i = 7, ki = 3
i = 8, ki = 2
i = 9, ki = 2
i = 10, ki = 3
```

e) The clustering coefficiente C_i for a node i is given by

$$C_i = \frac{\text{Number of links between neighbors of } i}{\text{Number of all possible conection between them}}$$

In [5]:
```
clust_coeff = local_clustering_coefficient(G)
println("Node i | Coefficient Ci")
for v in 1:n
    println("i = $v, Ci = $(clust_coeff[v])")
end
```

```
Node i | Coefficient Ci
i = 1, Ci = 0.3
i = 2, Ci = 0.0
i = 3, Ci = 0.5
i = 4, Ci = 1.0
i = 5, Ci = 0.6666666666666666
i = 6, Ci = 1.0
i = 7, Ci = 1.0
i = 8, Ci = 1.0
i = 9, Ci = 1.0
i = 10, Ci = 0.6666666666666666
```

f) The Closeness Centrality coefficient for a vertix i is given by the relation

$$C_B(v_i) = \frac{n-1}{\sum_k d(v_k, v_i)}$$

In [6]:
```
cb = closeness_centrality(G)
println("Node i | Coefficient Cbi")
for v in 1:n
    println("i = $v, Ci = $(cb[v])")
end
```

```
Node i | Coefficient Cbi
i = 1, Ci = 0.5625
i = 2, Ci = 0.5625
i = 3, Ci = 0.5
i = 4, Ci = 0.375
i = 5, Ci = 0.4090909090909091
i = 6, Ci = 0.375
i = 7, Ci = 0.375
i = 8, Ci = 0.391304347826087
i = 9, Ci = 0.391304347826087
i = 10, Ci = 0.4090909090909091
```

f) The Degree of Centrality is devinde by

$$C_D = \frac{d_v(s,t)}{d(s,t)}$$

Where

- $d_v(s,t)-$ Is the number of shortest path that passe through the vertix v
- $d(s,t)-$ Is the total amount of shortest paths

In [7]:
```
cb = degree_centrality(G)
println("Node i | Coefficient Cbi")
for v in 1:n
    println("i = $v, Ci = $(cb[v])")
end
```

```
Node i | Coefficient Cbi
i = 1, Ci = 0.5555555555555556
i = 2, Ci = 0.2222222222222222
i = 3, Ci = 0.4444444444444444
i = 4, Ci = 0.3333333333333333
i = 5, Ci = 0.3333333333333333
i = 6, Ci = 0.3333333333333333
i = 7, Ci = 0.3333333333333333
i = 8, Ci = 0.2222222222222222
i = 9, Ci = 0.2222222222222222
i = 10, Ci = 0.3333333333333333
```

Exercice II

a) Prepare a CSV with the edge list

In [8]:
```
df = DataFrame( From = [1,2,4,5,3,4,1,6,9,1,7,8,1,6,5], To = [2,3,3,1,6,7
CSV.write("data.csv",df)
df
```

15×2 DataFrame

Row	From	To
	Int64	Int64
1	1	2
2	2	3
3	4	3
4	5	1
5	3	6
6	4	7
7	1	8
8	6	7
9	9	10
10	1	9
11	7	3
12	8	5
13	1	10
14	6	4
15	5	10

Exercice III

a) The vector k of nodes degrees $k_i, \quad i \in \{1, 2, \ldots, N\}$

$$k = Ae$$

In [9]:
```
A = adjacency_matrix(G)
e = ones(n)

k = A * e
```

```
10-element Vector{Float64}:
 5.0
 2.0
 4.0
 3.0
 3.0
 3.0
 3.0
 2.0
 2.0
 3.0
```

b) The ammount of links L in the network

$$L = \frac{1}{2} k \cdot e$$

In [10]:
```
L = .5 * sum(k.*e)
```

```
15.0
```

c) The matrix N whose elements n_{ij} is equal to the number of common neighbors of nodes i and j

$$N = AA^T - D$$

such that D is the diagonal matrix representing the degrees of the nodes

In [11]:
```
N = A * transpose(A) - Diagonal(k)
```

```
10×10 SparseArrays.SparseMatrixCSC{Float64, Int64} with 48 stored entries:
  ⋅    ⋅   1.0   ⋅   2.0   ⋅    ⋅   1.0  1.0  2.0
  ⋅    ⋅    ⋅   1.0  1.0  1.0  1.0  1.0  1.0  1.0
 1.0   ⋅    ⋅   2.0   ⋅   2.0  2.0   ⋅    ⋅    ⋅
  ⋅   1.0  2.0   ⋅    ⋅   2.0  2.0   ⋅    ⋅    ⋅
 2.0  1.0   ⋅    ⋅    ⋅    ⋅    ⋅   1.0  2.0  1.0
  ⋅   1.0  2.0  2.0   ⋅    ⋅   2.0   ⋅    ⋅    ⋅
  ⋅   1.0  2.0  2.0   ⋅   2.0   ⋅    ⋅    ⋅    ⋅
 1.0  1.0   ⋅    ⋅   1.0   ⋅    ⋅    ⋅   1.0  2.0
 1.0  1.0   ⋅    ⋅   2.0   ⋅    ⋅   1.0   ⋅   1.0
 2.0  1.0   ⋅    ⋅   1.0   ⋅    ⋅   2.0  1.0   ⋅
```

d) The number T of triangles present in the network

$$T = \frac{1}{6} trA^3$$

In [12]:
```
T = tr(A^3) / 6
```

```
7.0
```

e) How would you determine whether the network is connected only by looking at the adjacency matrix?

A Network is connected if all the nodes are connected to at least with 1 other node. Since the element a_{ij} of the matrix A^k represents the number of paths of size k from the node i to j, we might say that if there is any 0 entry in the A^n matrix then there is a point not connected with anybody and therefore the network is disconected

In [13]:
```
A^n
```

```
10×10 SparseArrays.SparseMatrixCSC{Int64, Int64} with 100 stored entries:
 25032  10328   9291   6376  18504   6376   6376  13858  13858  18504
 10328   6849   9870   8577   7762   8577   8577   5975   5975   7762
  9291   9870  23289  20648   5561  20648  20648   4232   4232   5561
  6376   8577  20648  18670   3774  18669  18669   2959   2959   3774
 18504   7762   5561   3774  14179   3774   3774  10654  10709  14090
  6376   8577  20648  18669   3774  18670  18669   2959   2959   3774
  6376   8577  20648  18669   3774  18669  18670   2959   2959   3774
 13858   5975   4232   2959  10654   2959   2959   8116   8082  10709
 13858   5975   4232   2959  10709   2959   2959   8082   8116  10654
 18504   7762   5561   3774  14090   3774   3774  10709  10654  14179
```