```
SystemError: opening file "C:\\Users\\victo\\OneDrive - KU
Leuven\\Canada\\MTH6412B\\mth6412b-starter-code\\projet\\phase1\\node.jl": No such
file or directory

  1. var"#systemerror#83"(::Nothing, ::typeof(systemerror), ::String,
     ::Int32) @ error.jl:176
  2. #systemerror#82 @ error.jl:175 [inlined]
  3. systemerror @ error.jl:175 [inlined]
  4. var"#open#705"(::Bool, ::Nothing, ::Nothing, ::Nothing, ::Nothing, ::Nothing,
     ::typeof(open), ::String) @ iostream.jl:293
  5. open @ iostream.jl:275 [inlined]
  6. var"#open#409"(::Base.Pairs{Symbol, Union{}, Tuple{}, NamedTuple{(), Tuple{}}},
     ::typeof(open), ::Base.var"#418#419"{String}, ::String) @ io.jl:393
  7. open @ io.jl:392 [inlined]
  8. read @ io.jl:473 [inlined]
  9. _include(::Function, ::Module, ::String) @ loading.jl:1959
 10. include(::Module, ::String) @ Base.jl:457
 11. include(::String) @ PlutoRunner.jl:93
 12. top-level scope @ Local: 5
```

```julia
#the repository can be found on the github link:
https://github.com/houskkam/mth6412b-starter-code
begin
import Pkg
Pkg.add("Plots")
include("projet\\phase1\\node.jl")
include("projet\\phase1\\edge.jl")
include("projet\\phase1\\graph.jl")
include("projet\\phase1\\read_stsp.jl")
end
```

```
  Resolving package versions...
 No Changes to `C:\Users\victo\.julia\environments\v1.9\Project.toml`
 No Changes to `C:\Users\victo\.julia\environments\v1.9\Manifest.toml`
```

```
UndefVarError: `read_header` not defined

  1. top-level scope @ Local: 5
```

```julia
#A file name (fn) is defined by obtaining the current working directory with pwd() and combining it with a relative path to a file named "bayg29.tsp."
begin
fn = pwd() * "\\instances\\stsp\\bayg29.tsp"
#reads the data from the specified file (fn) and stores it in various variables such as header, almost_edges, and almost_nodes.
header = read_header(fn)
almost_edges = read_edges(header, fn)
almost_nodes = read_nodes(header, fn)
end

```

```
UndefVarError: `Node` not defined

  1. top-level scope @ Local: 4
```

```julia
# Constructing my_nodes of type Node from the given file
#starting with an empty array for nodes and filling it up with a for loop
begin
my_nodes = Vector{Node{Float64}}()
for almost_node in almost_nodes
    new_node = Node(string(almost_node[2][1]), almost_node[2][2])
    #adding the new nodes
    push!(my_nodes, new_node)
end
end
```

```
UndefVarError: `Edge` not defined

  1. top-level scope @ Local: 4
```

```julia
# Constructing my_edges of type Edge from the given file
#starting with an empty array for edges and filling it up with a for loop
begin
my_edges = Vector{Edge{Float64, Node{Float64}}}()
for almost_edge in almost_edges
    new_edge = Edge(my_nodes[almost_edge[1]], my_nodes[almost_edge[2]], almost_edge[3])
    #adding the new edges
    push!(my_edges, new_edge)
end
end
```

```
UndefVarError: `Graph` not defined

  1. top-level scope @ Local: 1
```

```julia
G = Graph("Ick", my_nodes, my_edges)
```