```
================================================================
== Communities_Detection.exe                                  ==
================================================================
== Community detection in complex networks by                 ==
== optimization of modularity:                                ==
==   - unweighted modularity (UN)                             ==
==   - weighted modularity (WN)                               ==
==   - weighted modularity with positive and negative links (WS) ==
==   - other measures                                         ==
== Implements several algorithms, which can be combined:      ==
==   - exhaustive search (h)                                  ==
==   - tabu search (t)                                        ==
==   - extremal optimization (e)                              ==
==   - spectral optimization (s)                              ==
==   - fast algorithm (f)                                     ==
==   - fine-tuning by reposition (r) or bootstrapping (b)     ==
================================================================


Usage:  Communities_Detection.exe  log_level  modularity_type  heuristics  repetitions  [
  resistance  [ penalty_coeff ] ]  net_name  lol_best_name

   Logging Levels      :  N | S | P | V
                            also lowercase symbols
                            also case-insensitive full names (None, ...)
                            N = None
                            S = Summary
                            P = Progress
                            V = Verbose

   Modularity Types    :  UN | UUN | WN | WS | WUN | WLA | WULA | WLUN | WNN | WLR
                            also lowercase symbols
                            also case-insensitive full names (Unweighted_Newman, ...)
                            UN   = Unweighted_Newman
                            UUN  = Unweighted_Uniform_Nullcase
                            WN   = Weighted_Newman
                            WS   = Weighted_Signed
                            WUN  = Weighted_Uniform_Nullcase
                            WLA  = Weighted_Local_Average
                            WULA = Weighted_Uniform_Local_Average
                            WLUN = Weighted_Links_Unweighted_Nullcase
                            WNN  = Weighted_No_Nullcase
                            WLR  = Weighted_Link_Rank

   Heuristics String   :  [htsefrb]+
                            also uppercase symbols
                            also single case-insensitive full names (Exhaustive, ...)
                            h = Exhaustive
                            t = Tabu
                            s = Spectral
                            e = Extremal
                            f = Fast
                            r = Reposition
                            b = Bootstrapping

   Repetitions         :  positive integer
                            does not apply to [hfr] algorithms

   Resistance          :  resistance of nodes to join communities in the form of a common
     self-loop
                            positive or negative real number
                            0 | 0.0 | default => no resistance, i.e. do not add self-
                              loops

   Penalty Coefficient :  relative importance of null-case term
                            non-negative real number
                            default => 1.0

   Network name        :  name of the input network file in Pajek format (*.net)
```

```
   Lol Best Filename   :  file for the best partition found in Lol format
                            if file exists before running, the partition in the file
                              becomes the initial partition


================================================================
== Communities_Network.exe                                    ==
================================================================
== Find the Network of Communities of a given Network         ==
================================================================


Usage:  Communities_Network.exe  net_name  clu_or_lol_name  [ number_of_lines_to_skip ]
  comms_net_name  weights_type  [ decimal_digits ]

   net_name                :  name of the input network file in Pajek format (*.net)

   clu_or_lol_name         :  name of the file with the partition in Pajek or Lol format

   number_of_lines_to_skip :  non-negative integer
                                default => 0
                                ignored for partitions in Pajek format

   comms_net_name          :  name of the output network of communities file in Pajek
     format

   weights_type            :  I | F | D
                                also lowercase symbols
                                also case-insensitive full names
                                I = Integer = Int
                                F = Float
                                D = Double

   decimal_digits          :  number of decimal digits for Float or Double output weights
                                ignored for Integer weights
                                default => 5


================================================================
== Compare_Partitions.exe                                     ==
================================================================
== Compare partitions in Lol or Pajek format                  ==
== Many indices and metrics are calculated:                   ==
==   - number of pairs, agreements and disagreements         ==
==   - Jaccard, Rand, adjusted Rand, Fowlkes Mallows          ==
==   - normalized mutual information, asymmetric Wallace      ==
==   - Mirkin, van Dongen, variation of information           ==
================================================================


Usage:  Compare_Partitions.exe  clu_or_lol(s)_1_name  clu_or_lol(s)_2_name  [ out_name [
  out_format ] ]  [ number_of_lines_to_skip ]

   clu_or_lol(s)_1_name    :  name of the file with the first partition(s) in Pajek or
     Lol(s) format
                                only one partition per file if in Pajek format

   clu_or_lol(s)_2_name    :  name of the file with the second partition(s) in Pajek or
     Lol(s) format
                                only one partition per file if in Pajek format

   out_name                :  name of the output file
                                contingency table not shown in verbose format if output
                                  is not file and size > 30x10

   out_format              :  V | T
                                also lowercase symbols
                                also case-insensitive full names (Verbose, ...)
                                V = Verbose
                                T = Table
                                default => Verbose
```

```
   number_of_lines_to_skip :  number of lines to skip at the beginning of the Lol files
                                ignored for partitions in Pajek format
                                non-negative integer
                                default => 0


================================================================
== Connected_Subgraphs.exe                                    ==
================================================================
== Split a network into its weak or strong connected components  ==
================================================================


Usage:  Connected_Subgraphs.exe  net_name_without_ext  [ components_type ]   [ skip_size
  ]

   net_name_without_ext :  name of the network file in Pajek format without the .net
     extension

   components_type      :  W | S
                           also lowercase symbols
                           also case-insensitive full names (Weak, Strong)
                           W = Weak
                           S = Strong
                           default => Weak

   skip_size            :  components smaller or equal to this size are skipped
                           non-negative integer
                           default => 0


================================================================
== Convert_Clu_To_Lol.exe                                     ==
================================================================
== Convert a file with a partition in Pajek format (*.clu)    ==
== into a file with a partition in Lol format                 ==
================================================================


Usage:  Convert_Clu_To_Lol.exe  clu_file_name  lol_file_name  [ sorted ]

   clu_file_name :  name of the input partition file in Pajek format (*.clu)

   lol_file_name :  name of the output partition file in Lol format

   sorted        :  any string as 3rd parameter produces a sorted List of Lists
                      communities sorted by decreasing size
                      elements of each community sorted by index


================================================================
== Convert_Lol_To_Clu.exe                                     ==
================================================================
== Convert a file with a partition in Lol format into         ==
== a file with a partition in Pajek format (*.clu)            ==
================================================================


Usage:  Convert_Lol_To_Clu.exe  lol_file_name  clu_file_name  [ number_of_lines_to_skip ]

   lol_file_name           :  name of the input partition file in Lol format

   clu_file_name           :  name of the output partition file in Pajek format (*.clu)

   number_of_lines_to_skip :  number of lines to skip at the beginning of the Lol file
                                non-negative integer
                                default => 0


================================================================
== Data_Statistics.exe                                        ==
================================================================
```

```
== Find statistic indicators of a data set, in rows or columns:  ==
== - minimum, maximum, percentiles                               ==
== - means: arithmetic, geometric, harmonic                      ==
== - variance, standard deviation, skewness, kurtosis            ==
== - covariance, central moments                                 ==
== - Pearson and Spearman correlation, and correlation errors    ==
== - linear regression                                           ==
====================================================================


Usage:  Data_Statistics.exe  data_name  [ statistics_name ]  [ index1 [ index2 ] ]
  rows_or_columns  [ decimal_digits ]

   data_name       :  name of the data file

   statistics_name :  name of the file with the output proximities matrix

   index1 index2   :  indices of the row(s) or column(s) to obtain the statistics
                        if no indices indicated, all individual and pair statistics
                          calculated
                        if first index indicated, the statistics of that row or column
                          calculated
                        if both indices indicated, the pair statistics are calculated
                        0 < index1 < index2

   rows_or_columns :  R | C
                        also lowercase symbols
                        also case-insensitive full names (Rows, ...)
                        R = Rows
                        C = Cols = Columns

   decimal_digits  :  number of decimal digits for float values
                        default => 14


====================================================================
== Data_To_Correlations.exe                                       ==
====================================================================
== Find the correlations network of a data set                    ==
====================================================================


Usage:  Data_To_Correlations.exe  data_file  rows_or_columns  scaling_type
  correlations_file  [ decimal_digits ]

   rows_or_columns :  R | C
                        also lowercase symbols
                        also case-insensitive full names (Rows, ...)
                        R = Rows
                        C = Cols = Columns

   scaling_type    :  NS | S01 | ZS
                        also lowercase symbols
                        also case-insensitive full names (No_Scale, ...)
                        NS  = No_Scale
                        S01 = Scale_01
                        ZS  = Z_Score

   decimal_digits  :  number of decimal digits for float values
                        default => 14


====================================================================
== Data_To_Proximities.exe                                        ==
====================================================================
== Calculate many types of proximities (distances or              ==
== similarities) between rows or columns in a data set:           ==
==   - Eucldean, Manhattan, Chebyshev, Minkowski, Canberra        ==
==   - Bray Curtis, correlation, cosine                           ==
==   - several scalings and transformations available             ==
====================================================================
```

```
Usage:  Data_To_Proximities.exe  data_name  proximities_name  rows_or_columns
  scaling_type  dissimilarity_type  [ dissimilarity_param ]  transform_type  [
  decimal_digits ]

   data_name           :  name of the data file

   proximities_name    :  name of the file with the output proximities matrix
                            if name has .net extension, the output is a network file in
                              Pajek format

   rows_or_columns     :  R | C
                            also lowercase symbols
                            also case-insensitive full names (Rows, ...)
                            R = Rows
                            C = Cols = Columns

   scaling_type        :  NS | S01 | SZS
                            also lowercase symbols
                            also case-insensitive full names (No_Scaling, ...)
                            NS  = No_Scaling
                                    x
                            S01 = Scaling_01
                                    (x - x_{min}) / (x_{max} - x_{min})
                            SZS = Scaling_Z_Score
                                    (x - <x>) / sigma_x

   dissimilarity_type  :  EUCL | MANH | CHEB | MINK | CANB | BRAY | CORD | CODI | CABS |
     CSQR | COSI
                            also lowercase symbols
                            also case-insensitive full names (Euclidean_Distance, ...)
                            EUCL = Euclidean_Distance
                                     \sum_k (x_k - y_k)^2
                            MANH = Manhattan_Distance
                                     \sum_k |x_k - y_k|
                            CHEB = Chebyshev_Distance
                                     \max_k |x_k - y_k|
                            MINK = Minkowski_Distance
                                     [\sum_k (x_k - y_k)^p]^(1/p)
                            CANB = Canberra_Distance:
                                     \sum_k \frac{ |x_k - y_k| }{ |x_k| + |y_k| }
                            BRAY = Bray_Curtis_Dissimilarity:
                                     \frac{2 \sum_k |x_k - y_k| }{ \sum_k (x_k + y_k) }
                            CORD = Correlation_Distance:
                                     \sqrt{2 (1 - \rho)}
                            CODI = Correlation_Dissimilarity:
                                     \frac{1}{2} (1 - \rho)
                            CABS = Correlation_Abs_Dissimilarity:
                                     1 - |\rho|
                            CSQR = Correlation_Sqr_Dissimilarity:
                                     \sqrt(1 - \rho^2)
                            COSI = Cosine_Dissimilarity
                                     \frac{1}{2} (1 - \frac{x y}{|x| |y|})

   dissimilarity_param :  parameter for some dissimilarity types, otherwise ignored
                            for Minkowski Distance: parameter p of the p-norm
                              integer or float number
                              default => 14
                            for Correlation Distances: correlation type
                              P | S
                              also lowercase symbols
                              also case-insensitive full names (Pearson, ...)
                              P = Pearson
                              S = Spearman
                              default => Pearson

   transform_type      :  NT | OMD | OM2D | IOD | EOMD
                            also lowercase symbols
                            also case-insensitive full names (No_Transform, ...)
                            NT   = No_Transform
                                     D
                            OMD  = One_Minus_Dissim
```

```
                                1 - D
                          OM2D = One_Minus_Two_Dissim
                                1 - 2 D
                          IOD  = Inverse_Of_Dissim
                                \frac{1}{D}
                          EOMD = Exp_Of_Minus_Dissim
                                \exp(-D)
                          OIZ  = One_If_Zero
                                \delta(D,0)

   decimal_digits      :  number of decimal digits for float values
                          default => 14


================================================================
== Extract_Subgraphs.exe                                      ==
================================================================
== Extract subgraphs from a graph                             ==
================================================================


Usage:  Extract_Subgraphs.exe  net_name  clu_or_lol_name  out_name_prefix  [
  number_of_lines_to_skip ]

   net_name               :  name of the network file in Pajek format

   clu_or_lol_name        :  name of the file with the lists of nodes in Pajek or Lol
     format

   out_name_prefix        :  prefix of the name of the output subgraph files

   number_of_lines_to_skip :  number of lines to skip at the beginning of the Lol file
                              ignored for partitions in Pajek format
                              non-negative integer
                              default => 0


================================================================
== Hierarchical_Clustering.exe                                ==
================================================================
== Agglomerative Hierarchical Clustering with MultiDendrograms   ==
== and Binary Dendrograms, for distances and similarities       ==
== Algorithms implemented:                                      ==
==   - Single linkage                                           ==
==   - Complete linkage                                         ==
==   - Unweighted average                                       ==
==   - Weighted average                                         ==
==   - Unweighted centroid                                      ==
==   - Weighted centroid                                        ==
==   - Ward                                                     ==
== MultiDendrograms generates always a unique dendrogram        ==
== For Binary Dendrograms, in case of ties, many dendrograms    ==
== may exist, and this tool can enumerate or count all of them,  ==
== or choose the one with maximum cophenetic correlation        ==
== See http://deim.urv.cat/~sergio.gomez/multidendrograms.php   ==
================================================================


Usage:  Hierarchical_Clustering.exe  proximities_name  output_prefix  dendrogram_type
  proximity_type  clustering_type  [ precision ]  [ dendrogram_mode ]  [
  internal_nodes_prefix ]

   proximities_name     :  name of the proximities file, either in matrix or list form
                           in matrix form, the names may be in first column, first row
                             , or none
                           in list form, missing values are filled with:
                             Double'Last for Distances
                             0.0         for Similarities

   output_prefix        :  prefix of the output files

   dendrogram_type      :  MD | BD
```

```
                              also lowercase symbols
                              also case-insensitive short and full names (Distance, ...)
                              MD | Multidendrogram
                              BD | Binary_Dendrogram

   proximity_type        :  D | S
                              also lowercase symbols
                              also case-insensitive short and full names (Distance, ...)
                              D | DIST | Distance
                              S | SIM  | Similarity

   clustering_type       :  SL | CL | UA | WA | UC | WC | WD | UPGMA | WPGMA
                              also lowercase symbols
                              also case-insensitive short and full names (Single_Linkage,
                                ...)
                              SL = Single_Linkage
                              CL = Complete_Linkage
                              UA = UPGMA = Unweighted_Average
                              WA = WPGMA = Weighted_Average
                              UC = Unweighted_Centroid
                              WC = Weighted_Centroid
                              WD = Ward

   precision             :  Number of decimal significant digits of the data and for the
     calculations
                              if not specified, is that of the value with largest number
                                of decimal digits

   dendrogram_mode       :  Sorted | Unsorted | Best | Count
                              also case-insensitive full names)
                              default => Sorted
                              mode discarded for MultiDendrograms

   internal_nodes_prefix :  Prefix for the names of the internal nodes
                              if 'None' (case insensitive) no names are assigned to
                                internal nodes
                              default => Cluster_


================================================================
== Links_Info.exe                                             ==
================================================================
== Obtain degrees and strengths of nodes attached to each link   ==
================================================================


Usage:  Links_Info.exe  net_name  [ num_random_links ]  links_info_name  [ decimal_digits
  ]

   net_name         :  name of the network file in Pajek format

   num_random_links :  number of random links in output info file
                         0 => all links
                         num_random_links >= num_links => all links
                         num_random_links > 1000000 => all links
                         default => 0

   links_info_name  :  name of the file with the info of links

   decimal_digits   :  number of decimal digits for float values
                         default => 5


================================================================
== List_To_Net.exe                                            ==
================================================================
== Convert a file with the list of links of a graph into      ==
== a network file in Pajek format (*.net)                     ==
================================================================


Usage:  List_To_Net.exe  list_input_file  net_output_file  [ network_type ]
```

```
   list_input_file : text file containing a list of links

   net_output_file : name of the output network file in Pajek format (*.net)

   network_type    :  A | D | U
                        also lowercase symbols
                        also case-insensitive full names (Auto, Directed, Undirected)
                        A = Auto
                        D = Directed
                        U = Undirected
                        default => Auto
                        in Auto, if the Graph is Symmetric, the output is Undirected
                        exception raised if inconsistent values exist


================================================================
== Matrix_To_List.exe                                         ==
================================================================
== Convert a file with a graph in matrix form into            ==
== a file with the list of links                              ==
================================================================


Usage:  Matrix_To_List.exe matrix_input_file  list_output_file  [ no_link_string ]

   matrix_input_file : text file containing a matrix

   list_output_file  : name of the output list file
                         if the matrix is symmetric, the lower triangular links are
                           discarded

   no_link_string    : string used to identify unexistent links within the matrix file
                         default => 0


================================================================
== Matrix_To_Net.exe                                          ==
================================================================
== Convert a file with a graph in matrix form into            ==
== a network file in Pajek format (*.net)                     ==
================================================================


Usage:  Matrix_To_Net.exe matrix_input_file  net_output_file  [ no_link_string ]

   matrix_input_file : text file containing an adjacency or weights matrix

   net_output_file   : name of the output network file in Pajek format (*.net)

   no_link_string    : string used to identify unexistent links within the matrix file
                         default => 0


================================================================
== Mesoscales_Detection.exe                                   ==
================================================================
== Mesoscales search in complex networks by optimization of   ==
== modularity using common self-loops                         ==
== Implements several algorithms for modularity optimization: ==
==   - exhaustive search (h)                                  ==
==   - tabu search (t)                                        ==
==   - extremal optimization (e)                              ==
==   - spectral optimization (s)                              ==
==   - fast algorithm (f)                                     ==
==   - fine-tuning by reposition (r) or bootstrapping (b)     ==
================================================================


Usage:  Mesoscales_Detection.exe  net_name  weighted_modularity_type  heuristics
  repetitions  [ num_steps  max_delta_loop_ratio ]  [ min_self_loop  max_self_loop ]
```

```
   Network Name              :  Name of the input network file in Pajek format (*.net)

   Weighted Modularity Types :  WN | WS | WUN | WLA | WULA | WLUN | WNN | WLR
                                  also lowercase symbols
                                  also case-insensitive full names (Weighted_Newman, ...)
                                  WN   = Weighted_Newman
                                  WS   = Weighted_Signed
                                  WUN  = Weighted_Uniform_Nullcase
                                  WLA  = Weighted_Local_Average
                                  WULA = Weighted_Uniform_Local_Average
                                  WLUN = Weighted_Links_Unweighted_Nullcase
                                  WNN  = Weighted_No_Nullcase
                                  WLR  = Weighted_Link_Rank

   Heuristics String         :  [htsefrb]+
                                  also uppercase symbols
                                  also single case-insensitive full names (Exhaustive, ..
                                    .)
                                  h = Exhaustive
                                  t = Tabu
                                  s = Spectral
                                  e = Extremal
                                  f = Fast
                                  r = Reposition
                                  b = Bootstrapping

   Repetitions               :  positive integer
                                  does not apply to [hfr] algorithms

   Number of Steps           :  default => 100

   Max Delta Loop Ratio      :  default => 1.0000
                                  ratio between the last and the first increments of the
                                    self-loop
                                  use 1 for a linear scale of the self-loop

   Min Self-loop             :  default => -1.0000
                                  for WN and WS the default is calculated from the
                                    network

   Max Self-loop             :  default => 1.0000
                                  for WN and WS the default is calculated from the
                                    network


================================================================
== Mesoscales_Fine_Tuning.exe                                 ==
================================================================
== Mesoscales fine-tuning after Mesoscales detection          ==
================================================================


Usage:  Mesoscales_Fine_Tuning.exe  net_name_without_ext  weighted_modularity_type

   net_name_without_ext      :  name of the network file in Pajek format without the .net
     extension
                                  it is supposed that files with this name and the
                                    following endings exist:
                                    *-table.txt: table with four columns: r, r-r_min, Q,
                                      num_comms
                                    *-lols.txt: the partitions found for the mesoscale in
                                      Lol format
                                    *-lols-extra.txt : optional file with extra
                                      partitions

   weighted_modularity_types :  WN | WS | WUN | WLA | WULA | WLUN | WNN | WLR
                                  also lowercase symbols
                                  also case-insensitive full names (Weighted_Newman, ...)
                                  WN   = Weighted_Newman
                                  WS   = Weighted_Signed
                                  WUN  = Weighted_Uniform_Nullcase
                                  WLA  = Weighted_Local_Average
```

```
                                WULA = Weighted_Uniform_Local_Average
                                WLUN = Weighted_Links_Unweighted_Nullcase
                                WNN  = Weighted_No_Nullcase
                                WLR  = Weighted_Link_Rank


================================================================
== Modularity_Calculation.exe                                 ==
================================================================
== Calculate the total modularity, decomposed in node         ==
== and community contributions                                ==
================================================================


Usage:  Modularity_Calculation.exe  net_name  clu_or_lol_name  [ resistance  [
  penalty_coeff ] ]  modularity_type  [ modularity_details ]  [ number_of_lines_to_skip ]

   net_name                :  name of the input network file in Pajek format (*.net)

   clu_or_lol_name         :  name of the file with the partition in Pajek or Lol format

   resistance              :  resistance of nodes to join communities in the form of a
     common self-loop
                                positive or negative real number
                                0 | 0.0 | default => no resistance, i.e. do not add self-
                                  loops

   penalty_coeff           :  relative importance of null-case term
                                non-negative real number
                                default => 1.0

   modularity_type         :  UN | UUN | WN | WS | WUN | WLA | WULA | WLUN | WNN | WLR
                                also lowercase symbols
                                also case-insensitive full names (Unweighted_Newman, ...)
                                UN   = Unweighted_Newman
                                UUN  = Unweighted_Uniform_Nullcase
                                WN   = Weighted_Newman
                                WS   = Weighted_Signed
                                WUN  = Weighted_Uniform_Nullcase
                                WLA  = Weighted_Local_Average
                                WULA = Weighted_Uniform_Local_Average
                                WLUN = Weighted_Links_Unweighted_Nullcase
                                WNN  = Weighted_No_Nullcase
                                WLR  = Weighted_Link_Rank

   modularity_details      :  T | TC | TN | TCN
                                also lowercase symbols
                                also case-insensitive full names (Total,
                                  Total_Communities, ...)
                                T   = Total
                                TC  = Total_Communities
                                TN  = Total_Nodes
                                TCN = Total_Communities_Nodes
                                default => Total_Communities_Nodes

   number_of_lines_to_skip :  number of lines to skip at the beginning of the Lol files
                                ignored for partitions in Pajek format
                                non-negative integer
                                default => 0


================================================================
== Multiplex_Aggregate.exe                                    ==
================================================================
== Aggregate the Layers of a Multiplex                        ==
================================================================


Usage:  Multiplex_Aggregate.exe  list_input_file  output_file  network_type
  aggregation_type  weights_type  [ decimal_digits ]

   list_input_file  :  text file containing the list of links of a multiplex
```

```
   output_file      :  prefix of output layer networks

   network_type     :  D | U
                         also lowercase symbols
                         also case-insensitive full names (Directed, Undirected)
                         D = Directed
                         U = Undirected
                         for repeated Edges, only the last one is stored

   aggregation_type :  W | U
                         also lowercase symbols
                         also case-insensitive full names (Weighted, Unweighted)
                         W = Weighted
                         U = Unweighted

   weights_type     :  I | F | D
                         also lowercase symbols
                         also case-insensitive full names (Integer, ...)
                         I = Integer = Int
                         F = Float
                         D = Double

   decimal_digits   :  number of decimal digits for float and double weights
                         ignored for Integer weights
                         default => 5


================================================================
== Multiplex_Extract_Layers.exe                               ==
================================================================
== Extract the Layers of a Multiplex as Networks in Pajek format ==
================================================================


Usage:  Multiplex_Extract_Layers.exe  list_input_file  net_output_prefix  network_type

   list_input_file   : text file containing the list of links of a multiplex

   net_output_prefix : prefix of output layer networks

   network_type      : D | U
                         also lowercase symbols
                         also case-insensitive full names (Directed, Undirected)
                         D = Directed
                         U = Undirected
                         for repeated Edges, only the last one is stored


================================================================
== Net_To_List.exe                                            ==
================================================================
== Convert a network file in Pajek format (*.net) into        ==
== a file with the list of links                              ==
================================================================


Usage:  Net_To_List.exe net_input_file  list_output_file

   net_input_file   : name of the input network file in Pajek format (*.net)

   list_output_file : name of the output network file in list format


================================================================
== Net_To_Matrix.exe                                          ==
================================================================
== Convert a network file in Pajek format (*.net) into        ==
== a file with a graph in matrix form                         ==
================================================================
```

```
Usage:  Net_To_Matrix.exe net_input_file  matrix_output_file  [ no_link_string ]

   net_input_file     : name of the input network file in Pajek format (*.net)

   matrix_output_file : output text file containing the weights matrix of the network
                          the first line contains the names of the nodes

   no_link_string     : string used to identify unexistent links within the matrix file
                          default => 0


================================================================
== Network_Properties.exe                                     ==
================================================================
== Find many global, node and edge properties of a network:   ==
==   - connectedness (weak or strong)                         ==
==   - degrees, strengths, clustering coefficients, entropies ==
==   - assortativities, path lengths, efficiencies, diameters ==
==   - betweenness (nodes and edges)                          ==
==   - degree distribution                                    ==
== Works with weighted and unweighted, directed and undirected,  ==
== positive and signed networks                               ==
================================================================


Usage:  Network_Properties.exe  net_name  [ properties ]  [ decimal_digits ]

   Network Name      :  Name of the input network file in Pajek format (*.net)

   Properties String :  [GNEDLUFA]+
                          also uppercase symbols
                          also single case-insensitive full names (All, Global, ...)
                            G = Global
                            N = Nodes
                            E = Edges
                            D = Degrees
                            L = Distances
                            U = Unweighted
                            F = Fast
                            A = All
                          default => All
                          properties available in each class
                            G: type and size of graph, connectedness, average and total
                              degree and strength,
                               minimum and maximum values, asymmetry, reciprocity,
                                 assortativity,
                               average clustering coefficient, average path length,
                                 diameter, efficiency,
                               average entropy
                            N: degrees, strengths, self-loop, minimum, maximum and
                              average values,
                               clustering coefficient, average and maximum path lengths,
                                 efficiency,
                               entropy, node betweenness
                            E: edge betweenness
                            D: degree distribution
                            L: distances between nodes
                            U: unweighted properties, excluding weighted ones
                            F: only fast calculation properties:
                               exclude average and maximum path length, diameter,
                                 efficiency,
                               betweenness and distances
                            A: all properties available; disables Unweighted and Fast
                          processed from left to right, thus AU is not equivalent to UA
                          weights should be distances to have meaningful shortest path
                            weighted properties

   Decimal Digits    :  number of decimal digits for float values
                          default => 14


================================================================
```

```
== Reformat_Partitions.exe                                          ==
======================================================================
== Reformat partitions in Lol or Pajek format changing              ==
== nodes' indices by nodes' names, and grouping in columns          ==
======================================================================


Usage:  Reformat_Partitions.exe net_name  clu_or_lol_name  lol_out_name  [ header_lines
  header_mode ]  [ group_by  justify_width  skip_size ]

   net_name        : name of the network file in Pajek format (*.net)

   clu_or_lol_name : name of the partitions file in Pajek format (*.clu) or Lol format
                       in Lol format, the file may contain many partitions, e.g. those
                         describing mesoscales

   lol_out_name    : name of the reformatted partition file

   header_lines    : number of lines of the header before a partition in Lol format
                       non-negative integer
                       default => 0
                       ignored for partitions in Pajek format (*.clu)

   header_mode     : CH | NH | SH
                       also lowercase symbols
                       also case-insensitive full names (Copy_Header, ...)
                       CH = Copy_Header
                       NH = No_Header
                       SH = Separator_Header
                       default => No_Header

   group_by        : number of columns for the nodes' names in the reformatted partition
     file
                       positive integer
                       default => 1

   justify_width   : width of the columns for the nodes' names
                       positive integer
                       default => 1

   skip_size       : modules smaller or equal to this size are skipped
                       non-negative integer
                       default => 0


======================================================================
== Size_Reduction.exe                                               ==
======================================================================
== Reduction of the size of a network preserving modularity,        ==
== by elimination of simple and triangular 'hairs'                  ==
== Only for Weighted_Newman (WN) modularity type                    ==
======================================================================


Usage:  Size_Reduction.exe  net_name_without_ext  weights_type  [ decimal_digits ]

   net_name_without_ext :  name of the network file in Pajek format without the .net
     extension

   weights_type         :  I | F | D
                             also lowercase symbols
                             also case-insensitive full names
                             I = Integer = Int
                             F = Float
                             D = Double

   decimal_digits       :  number of decimal digits for Float or Double output weights
                             ignored for Integer weights
                             default => 5


======================================================================
```

```
== Size_Reduction_Lol_Expand.exe                                  ==
====================================================================
== Expansion of a partition of a size-reduced network into        ==
== a partition of the original network                            ==
====================================================================


Usage:  Size_Reduction_Lol_Expand.exe  reduced_lol_name  reducing_lol_name
  expanded_lol_name  [ header_lines  header_mode ]

   reduced_lol_name  : name of the input partition file in Lol format of a size-reduced
     network
                         the file may contain many partitions

   reducing_lol_name : name of the input partition file in Lol format which has reduced a
     network

   expanded_lol_name : name of the output partition file in Lol format
                         corresponds to the expansion of the partition of the size-
                           reduced network

   header_lines      : number of lines of the header before a partition in Lol format
                         non-negative integer
                         default => 0
                         ignored for partitions in Pajek format (*.clu)

   header_mode       : CH | NH | SH
                         also lowercase symbols
                         also case-insensitive full names (Copy_Header, ...)
                         CH = Copy_Header
                         NH = No_Header
                         SH = Separator_Header
                         default => No_Header


====================================================================
== Sort_Nodes.exe                                                 ==
====================================================================
== Sort nodes randomly or according to degree                     ==
====================================================================


Usage:  Sort_Nodes.exe  net_name  sorted_net_name  [ sort_direction ]

   sort_direction :  A | D | R
                       also lowercase symbols
                       also case-insensitive full names (Ascending, ...)
                       A = Asc  = Ascending
                       D = Desc = Descending
                       R = Rand = Random
                       default => Ascending


====================================================================
== Spanning_Tree.exe                                              ==
====================================================================
== Find the minimum or maximum spanning tree of a                 ==
== weighted network                                               ==
====================================================================


Usage:  Spanning_Tree.exe  net_name  mst_net_name  optimization_type  weights_type  [
  decimal_digits ]

   net_name          : name of the input network file in Pajek format (*.net)

   mst_net_name      : name of the output spanning tree file in Pajek format (*.net)

   optimization_type :  MIN | MAX
                          also lowercase symbols
                          also case-insensitive full names
                          MIN = Minimum
```

```
                          MAX = Maximum

   weights_type      :  I | F | D
                          also lowercase symbols
                          also case-insensitive full names
                          I = Integer = Int
                          F = Float
                          D = Double

   decimal_digits    :  number of decimal digits for Float or Double output weights
                          ignored for Integer weights
                          default => 5


=====================================================================
== Symmetrize_Network.exe                                          ==
=====================================================================
== Symmetrization of a directed graph                              ==
=====================================================================


Usage:  Symmetrize_Network.exe  net_name  sym_net_name  weights_type  [ decimal_digits ]

   net_name       : name of the input network file in Pajek format (*.net)

   sym_net_name   : name of the output symmetrized network file in Pajek format (*.net)

   weights_type   :  I | F | D
                       also lowercase symbols
                       also case-insensitive full names
                       I = Integer = Int
                       F = Float
                       D = Double

   decimal_digits :  number of decimal digits for Float or Double output weights
                       ignored for Integer weights
                       default => 5
```