Class that offers a variety of matrices methods. including:

1. init a matrix - if not valid form of matrix, raise error.
2. validation of matrix - if the matrix hold non-int or non-float values, prints explnation of why the matrix is not valid. if there is a different dim of a specific vector, prints explnation of why the matrix is not valid.
3. multiplication - multiply the matrix with other matrix (both objects of the class). can multiply only if dimensions are compatible for multiplication, prints explnation if not compatible.
4. addition - adds the values of the matrix to another matrix (both objects of the class). also adds only if dimensions are compatible for multiplication, prints explnation if not compatible.
5. transpose - transpose the matrix (all of its relevant values are transposed).
6. find\_trace - returns the matrix trace, if not square, returns False.
7. find\_rank - returns matrix rank by using the "echelon" method (to be explained).
8. echelon - returns the echelon form of the row vectors or column vectors, depends on users decision. can also return the logs of elementary operations that were used, dependes on users decision.
9. canonical\_form - returns the canonical form of the row vectors or column vectors, depends on users decision. can also return the logs of elementary operations that were used, dependes on users decision.
10. basis - returns the basis of row vectors or column vectors, depends on users decision.
11. determinant - returns the determinant of the matrix.
12. create\_block\_matrix - returns the block matrix created by two matrices (both objects of the class), creates the block matrix only if dimensions compatible, depends on users decision (on how to connect the matrices).
13. invert - returns the inverse of the matrix, uses both "echelon" and "canonical\_form" methods, and the log of elementary operations used in both methods.
14. is symmetric - returns if the matrix is symmetric or not, adding a new object value "self.symmetric".
15. is diagonal - returns if the matrix is diagonal or not, adding a new object value "self.diagonal".