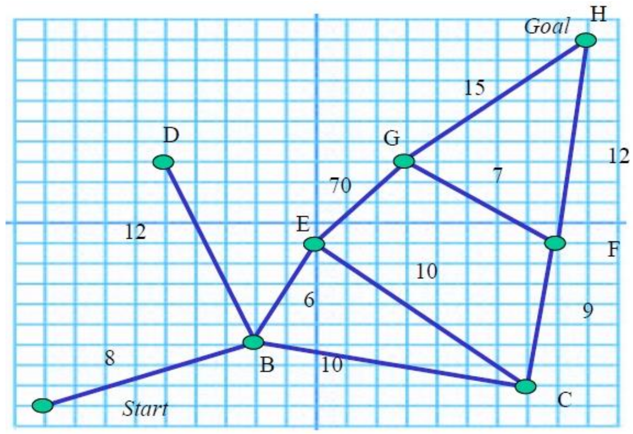
ASSIGNMENT – 3

1. Implement ***A**** search algorithms and test them on following graph and print the Search sequence (***Heuristic: Euclidian Distance***). Report Order of nodes visited and Solution Path for each of the search techniques.



2. Implement GA to solve N Queens problem.
   Following guidelines should be used to develop the solution.
   a. Use proper encoding approach to design the chromosome.
   b. Use proper objective function.
   c. Population size is 100 and it is fixed.
   d. Initial population should be constructed randomly.
   e. Proper parent selection mechanism should be incorporated.
   f. Single point crossover operator should be used. Initially, set the Probability of the crossover operator to 0.7 then decrease it over the iterations (i.e. 100) by 0.005. Mutation probability (i.e. 0.01) is fixed.
   g. Use proper metric to assess the quality of the population in each iteration. Plot these values.

3. Implement single solution based search algorithm (e.g., Hill Climbing Algorithm) for Function Optimization.
4. Implement Simulated Annealing algorithm to solve 15-puzzle problem.

PROLOG

1. Write a Prolog program that removes repetitions from a list; for example, given [1,5,6,6,4,4,4,5,2,2,3,2] it would produce [1,5,6,4,5,2,3,2].

2. Write a Prolog program that reverses a list.

3. Write a Prolog predicate distance/3 to calculate the distance between two points in the 2-dimensional plane. Points are given as pairs of coordinates. Examples:
?- distance((0,0), (3,4), X).
X = 5.0
Yes
?- distance((-2.5,1), (3.5,-4), X).
X = 7.810249675906654
Yes

4. Write a Prolog predicate fibonacci/2 to compute the nth Fibonacci number. The Fibonacci sequence is defined as follows:
F0 = 0
F1 = 1
Fn = $F_{n-1} + F_{n-2}$ for n >= 2
Examples:
?- fibonacci(1, X).
X = 1
Yes
?- fibonacci(3, X).
X = 2
Yes
?- fibonacci(7, X).
X = 13
Yes

5. Write a program that generates a(b*c + bd)*.

6. Write a Prolog program that (recognises and) generates the language defined by the grammar:

S → QNQ
QN → QR
RN → NNR
RQ → NNQ
N → a
Q → ε

7. Recall that the set of prime numbers is {2, 3, 5, 7, 11, 13, 17,…}, i.e., the set of numbers with exactly two divisiors each (namely 1 and the number itself). Write a Prolog predicate prime/1 to check whether given number is prime. Examples:
prime(17). prime(18).
Yes No

8. Implement Euclid's algorithm to compute the greatest common divisor (GCD) of two non-negative integers. This predicate should be called gcd/3 and, given two non-negative integers in the first two argument positions, should match the variable in the third position with the GCD of the two given numbers. Examples:

   ```
   ?- gcd(57, 27, X).
   X = 3
   Yes
   ?- gcd(1, 30, X).
   X = 1
   Yes
   ?- gcd(56, 28, X).
   X = 28
   Yes
   ```

   Make sure your program behaves correctly also when the semicolon key is pressed.

9. Write a Prolog predicate divisors/2 to compute the list of all divisors for a given natural number. Example:

   ```
   ?- divisors(30, X).
   X = [1, 2, 3, 5, 6, 10, 15, 30]
   Yes
   ```

   Make sure your program doesn't give any wrong alternative solutions and doesn't fall into an infinite loop when the user presses the semicolon key.

10. Translate the following Prolog program into a set of first-order logic formulas:

    ```
    parent(peter, sharon).
    parent(peter, lucy).

    male(peter).

    female(lucy).
    female(sharon).
    father(X, Y) :-
            parent(X, Y),
            male(X).

    sister(X, Y) :-
            parent(Z, X),
            parent(Z, Y),
            female(X).
    ```