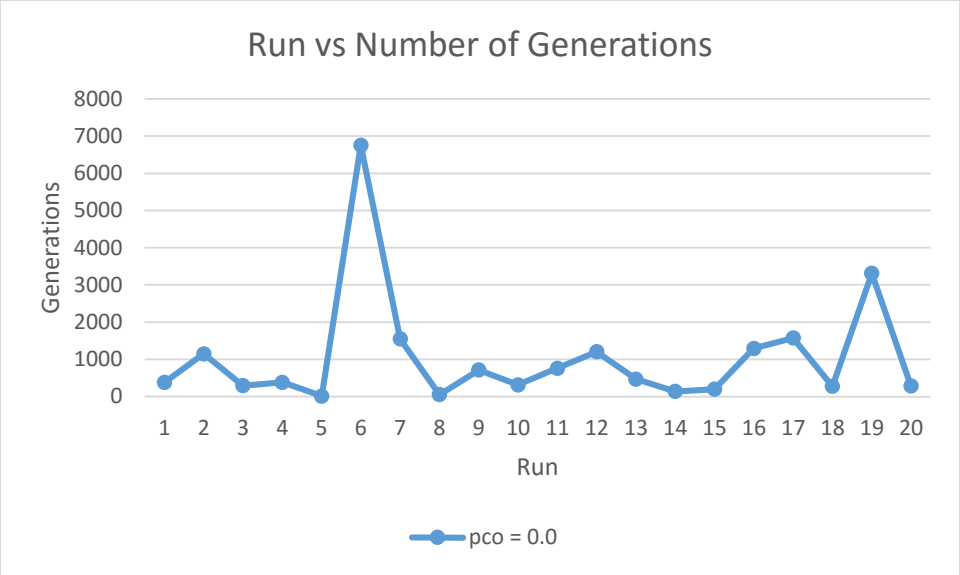
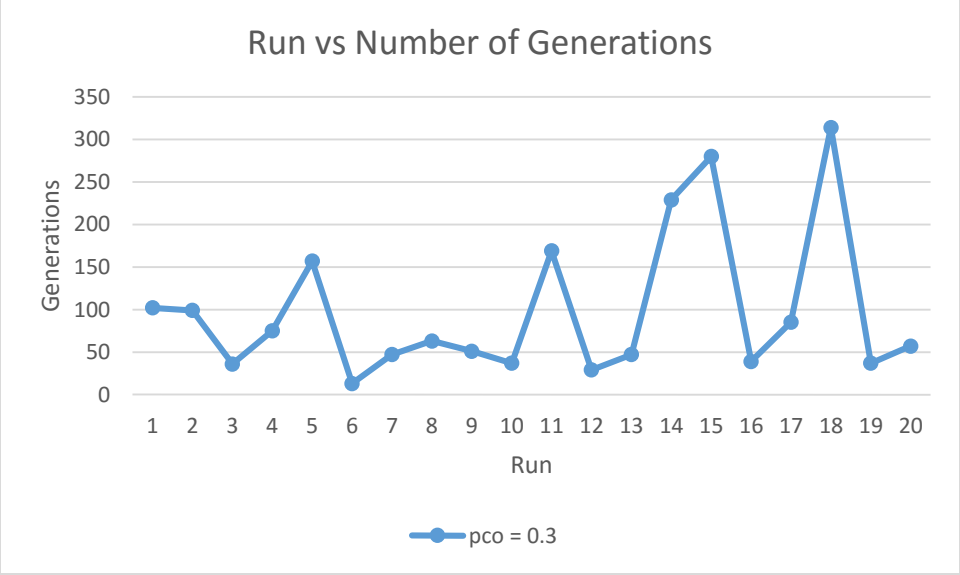
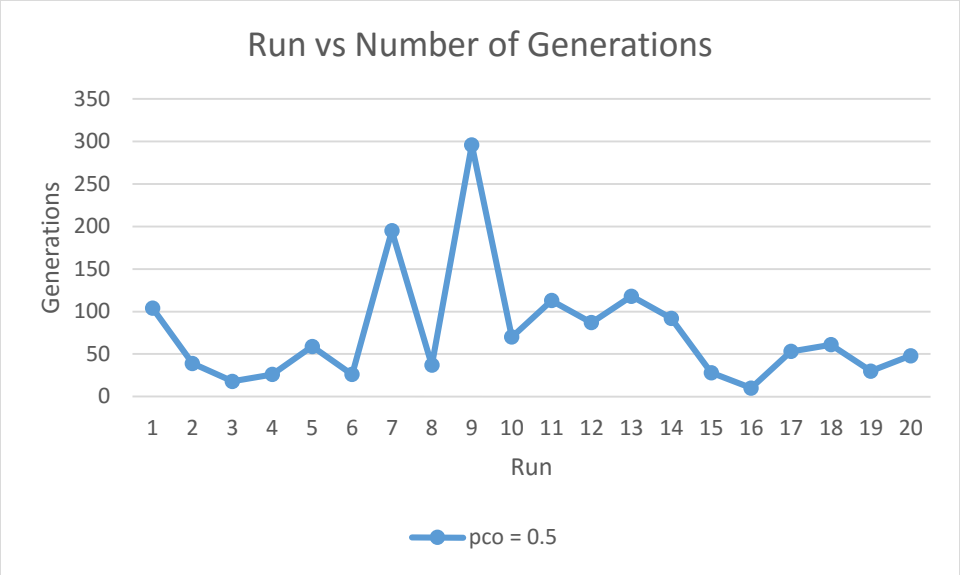
Homework 1

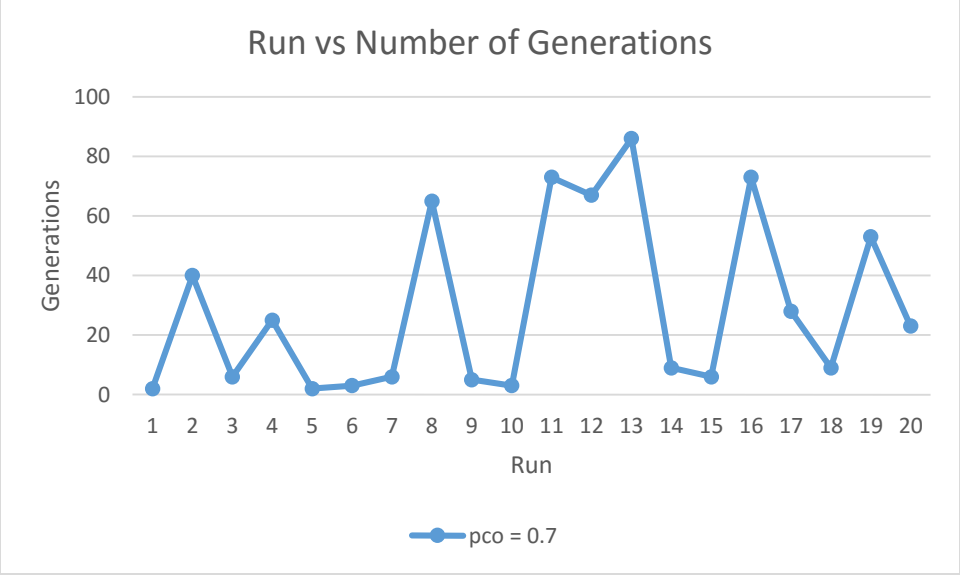
The genetic algorithm will use three classes to perform it. The Chromosome class is the class containing constructors and functions to manipulate the contents of each Chromosome. The Population class is the class containing the constructors and functions to manipulate the population of 20 chromosomes called chromosomes which are chromosome objects of size 10. The GA class will perform the actual algorithm by using the function GAEvaluate which takes in the initial population as input. The function does a roulette selection using the rouletteSelection function and proceeds to do replication, crossover and then mutation. This is done until the solution with a fitness value of ten is found in the population.

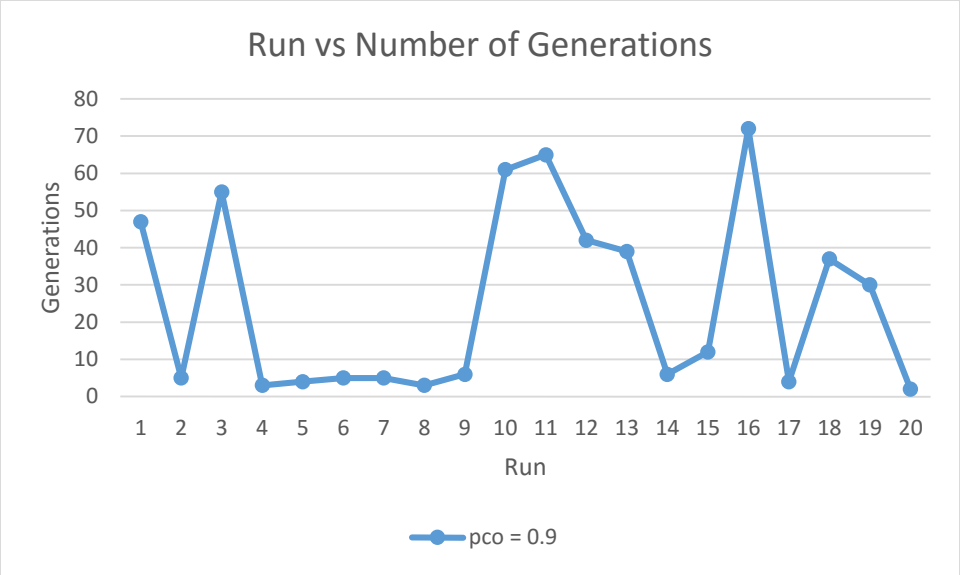
```
GAEvalute
Input: Population initialPopulation
Output: Number of generations
        Set Population object pop to values of the initialPopulation
        Set numbOfGenerations to one
        if solution is in pop then set variable found true else false
        while found is not true
                set Population object newGen with the values in pop
                do rouleteSelection on pop
                do replication on pop
                if Pco value is not zero then do crossOver
                do mutate on pop
                put all the new values of newGen in pop
                calculate the fitness values of all the chromosomes in pop
                if solution is in pop then set variable found true else false
                increase numbOfGenerations by one
        end while
        return numbOfGenerations
```


Run vs Number of Generations
Generations
8000
7000
6000
5000
4000
3000
2000
1000
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Run
pco = 0.0


Run vs Number of Generations
Generations
350
300
250
200
150
100
50
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Run
pco = 0.3


Run vs Number of Generations
Generations
350
300
250
200
150
100
50
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Run
pco = 0.5


Run vs Number of Generations
Generations
100
80
60
40
20
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Run
pco = 0.7



Pco Value	0	0.3	0.5	0.7	0.9
Average Number of Generations	1056.5	98.3	75.5	29.2	25.15

Pco 0 Output:
Initial Population:
0110010000 0111101011 1100010011 1101111100 0101110110 0010000100 1000111110
0100010111 0100011111 0000000000 1011011111 1001101110 1101011001 0001101100
0011011100 0000000011 0000100000 1001001001 0101001000 1001110000
Population After CrossOver:
0110010000 0111101011 1100010011 1101111100 0101110110 0010000100 1000111110
0100010111 0100011111 0000000000 1011011111 1001101110 1101011001 0001101100
0011011100 0000000011 0000100000 1001001001 0101001000 1001110000
Population After Mutation:
0110010000 0111101011 1100010011 1101111100 0101110110 0010000100 1000111110
0100010111 0100011111 0000000000 1011011111 1001101110 1101011001 0001001100
0011011100 0000000011 0000100000 1001001001 0101001000 1001110000
Population After CrossOver:
0110010000 0111101011 1100010011 1101111100 0101110110 0010000100 1000111110
0100010111 0100011111 0000000000 1011011111 1001101110 1101011001 0001001100
0011011100 0000000011 0000100000 1001001001 0101001000 1001110000
Population After Mutation:
0110010010 0111101011 1100010011 1101111100 0101110110 0010000100 1000111110
0100010111 0100011111 0000000000 1011011111 1001101110 1101011001 0001001100
0011011100 0000000011 0000100000 1001001001 0101001000 1001110000

Population After CrossOver:
1000010110 1010110000 1010010001 0111110101 0001010110 1111011001 0100100111
0001100101 0000110000 0111010111 1010010111 1100111101 0111010011 0011010110
1010010101 1000101111 0101011001 1101101010 1111101111 0011100010
Population After Mutation:
1000010110 1010110000 1010010001 0111110101 0001010110 1111011001 0100100110
0001100101 0000110000 0111010111 1010010111 1100111101 0111010011 0011010110
1010010101 1000101111 0101011001 1101101010 1111101111 0011100010
Population After CrossOver:
1000010110 1010110000 1010010001 0111110101 0001010110 1111011001 0100100110
0001100101 0000110000 0111010111 1010010111 1100111101 0111010011 0011010110
1010010101 1000101111 0101011001 1101101010 1111101111 0011100010
Population After Mutation:
1000010110 1010110000 1010010001 0111110101 0001010110 1111011001 0100100110
0001100101 0000110000 0111010111 1010010111 1100111101 0111010011 0011010110
1010010101 1000101111 0101011001 1101101010 1111111111 0011100010

Pco 0.7 Output:
Initial Population:
1011001001 1111111100 1010100101 1011000010 0011111110 0110001001 1000010100
0000010100 1000110010 1010000101 1011001010 0011011110 1111100001 0000100100
0000111111 1110010000 1101001010 1110010011 0001000100 1011110001
Population After CrossOver:
1101001010 1010100101 1111111100 1011110001 0000111111 0001000100 0110000101
1010001001 1010000001 1111100101 0011010000 1110011110 1011011110 0011000010
1011011110 0011100010 1011010100 0000001010 0000010000 1110010100
Population After Mutation:
1101001010 1010100101 1111111100 1011110001 0000111111 0001000100 0110000101
1010001001 0010000001 1111100101 0011010000 1110011110 1011011110 0011000010
1011011110 0011100010 1011010100 0000001010 0000010000 1110010100
Population After CrossOver:
1110011110 1111100101 1010001001 1011110001 0011100010 0000010000 0011010100
1011000010 0010010000 0011000001 1011000001 0010011110 0000111110 1011011111
1010100100 0001000101 1011001010 1101011110 1011011100 1111111110
Population After Mutation:
1110011110 1111100101 1010001001 1011110001 0011100010 0000010000 0011010100
1011000010 0010010000 0011000001 1011000001 0010011110 0000111110 1011011111
1010100100 0001000101 1011001010 1101011110 1111011100 1111111110

Population After CrossOver:
1111011100 0111011110 1111011110 1111111110 1101111100 1101011110 0111011110
1111011110 0111011110 1111111110 1111111100 1111111110 1111111110 1111011100
1101011110 1111011110 1111011110 1101011110 1101011110 1111111110
Population After Mutation:
1111011100 0111011110 1111011110 1111111110 1101111100 1101011110 0111011110
1111011111 0111011110 1111111110 1111111100 1111111110 1111111110 1111011100
1101011110 1111011110 1111011110 1101011110 1101011110 1111111110
Population After CrossOver:
1101011110 0111011110 1111011110 1111011100 1101111100 1111111110 1101011110
1111111110 1111111110 1111011110 1111111110 1111111110 1111111111 1111011110
1111011110 1111111111 1111111110 1111111100 1111011111 1111011100
Population After Mutation:
1101011110 0111011110 1111011110 1111011100 1101111100 1111111110 1101011110
1111111110 1111111110 1111011110 1111111110 1111111110 1111111111 1111011110
1111011110 1111111111 1101111110 1111111100 1111011111 1111011100

Part 3:

(i) The pco value of 0.9 produced the best results with a average value of 25.15 average number of generations.

(ii) Mutation is better with crossover as with crossover. The reason being that crossover pulls the whole population towards the solution rather than just one individual chromosome like mutation does. This allows the population to converge faster towards the solution than just using mutation.