|  |  |  |
| --- | --- | --- |
|  | **Low level** | **High Level** |
| 1 | Define Board | Get the initial board from player and set the maximum number of geerations to attempt |
| 2 | Get user input for all squares |
| 3 | Define dataframe |
| 4 | distinct\_letters = number of letters player has entered |
| 5 | Max\_gen |
| 6 | empty\_squares = 16 – number of non empty inputs |
| 7 | Initial\_population | Generate an initial population of Chromosome s |
| 8 | fitness function | Measure fitness of population |
| 9 | Define population size N |
| 10 | Define chromosome as binary number dimension distinct\_letters + 1 |
| 11 | Termination\_condition = 1 |
| 12 | Fitness = 1 |
| 13 | For all chromosomes do |
| 14 | If row contains duplicate letters the fitness = fitness – 1 |
| 15 | If column contains duplicate letters the fitness = fitness – 1 |
| 16 | If Quadrant 1 contains duplicate letters the fitness = fitness – 1 |
| 17 | If Quadrant 2 contains duplicate letters the fitness = fitness – 1 |
| 18 | If Quadrant 3 contains duplicate letters the fitness = fitness – 1 |
| 19 | If Quadrant 4 contains duplicate letters the fitness = fitness – 1 |
| 20 | Number\_empty = 16 – number populated |
| 21 | Fitness = fitness – ((1/16) \* number\_empty) |
| 22 | next |
| 23 | If fitness = 1 or max\_gen for any chomesome then | If termination condition satisified then end |
| 24 | return solution |
| 25 | For all chromosomes do | Run crossover function |
| 26 | Crossover |
| 27 | Mutation | Run mutation function and repeat until new population of same size generated |
| 28 | Next |
| 29 | Go to 9 | Repeat process until termination condition is met |