**ReadMe.txt Naga Sai Ravi Teja Majeti - 00927935**

To compile .cpp files, Use command g++ filename.cpp, an output file a.out will be generated. Now just run the command ./a.out and enter required input, then the code gets executed and output will be displayed.  
  
Python files can be just run by using command python filename.py and enter required input, then the code gets executed and output will be displayed.  
  
1. hca.cpp - This is for question 1. It displays the minimum found in each run as output  
  
2. ade.cpp - This is for question 2. It displays input that is generated randomly and converged points as output and also the minimum value found

Minimum point of Ackley function is at (0,0)   
  
3. ghc.cpp - This is for question 3. It displays queens initial and final positions for first 10 runs along with any queen positions found where zero queens attack each other. Later for another 90 runs, it displays queen positions found that do not attack each other along with run number.  At last it displays total number of positions (Optimizations) found where none of the queens attack each other and also total minimum fitness found in 100 runs (Fitness value is equal to number of pairs of queens that attach each other.)  
  
4. gan.py - This is for question 4. Input should be give for populations, number of queens and runs. It just displays queen positions found where none of them attack each other. And also total number of positions (Optimizations) found in total runs. — For 16 queens populations of 1000 or above and also more runs than 8 queens is recommended to get atleast one optimization. Also program is expected to run for more than 5min for higher runs and populations. (Children with same zero fitness positions may repeat in algorithm)

1. **Output and Plot for Ackleys Hill Climbing algorithm for 100 runs**

Below is output for minimum found in first 10 runs of hill climbing algorithm

0 -15.6644

1 -12.5307

2 -10.6031

3 -10.6934

4 -12.2777

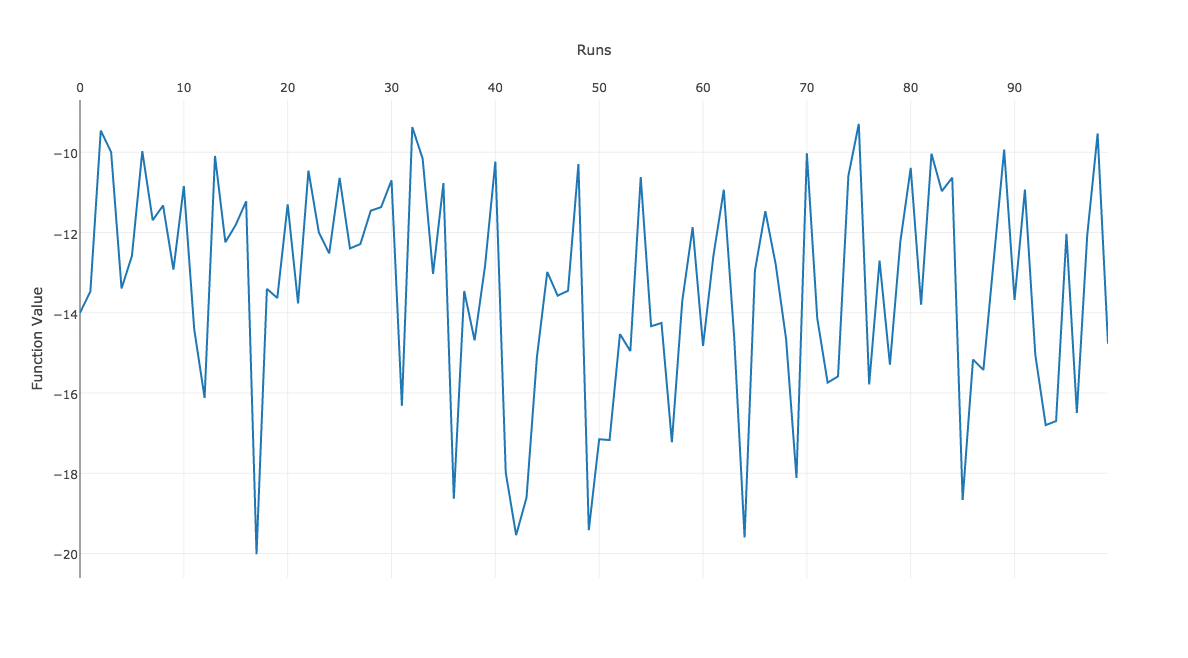
5 -11.1087

6 -14.3519

7 -9.97399

8 -10.872

9 -10.74



1. **Output and Plot for Ackleys Differential Evolution algorithm for 100 runs**

Output displays intial x,y and final x,y values. We can see that final x,y converge near 0,0 which is minimum for ackleys function. Below is output for 1 run

Initial x,y values are -- x is : -3.94289 -- y is 1.91314 -- Its Ackley is : -13.2098

Initial x,y values are -- x is : 4.06345 -- y is 4.4241 -- Its Ackley is : -9.5691

Initial x,y values are -- x is : -4.08249 -- y is -4.37845 -- Its Ackley is : -9.65328

Initial x,y values are -- x is : 1.47349 -- y is 4.9507 -- Its Ackley is : -10.6167

Initial x,y values are -- x is : -3.53373 -- y is -1.38835 -- Its Ackley is : -12.1094

Initial x,y values are -- x is : -3.95888 -- y is 3.14701 -- Its Ackley is : -11.9739

Initial x,y values are -- x is : 1.72705 -- y is -3.50593 -- Its Ackley is : -12.0725

Initial x,y values are -- x is : -4.12788 -- y is 2.64486 -- Its Ackley is : -11.0394

Initial x,y values are -- x is : 2.15988 -- y is 1.08082 -- Its Ackley is : -16.2373

Initial x,y values are -- x is : -4.66218 -- y is 2.80918 -- Its Ackley is : -10.1851

Final x,y values are -- x is : 0.173384 -- y is -0.00199919 -- Its Ackley is : -21.5937

Final x,y values are -- x is : -0.0328449 -- y is -0.0136245 -- Its Ackley is : -22.5844

Final x,y values are -- x is : 0.0511929 -- y is 0.00592935 -- Its Ackley is : -22.5033

Final x,y values are -- x is : 0.0963608 -- y is 0.00168356 -- Its Ackley is : -22.2163

Final x,y values are -- x is : 0.0074695 -- y is 0.162591 -- Its Ackley is : -21.6841

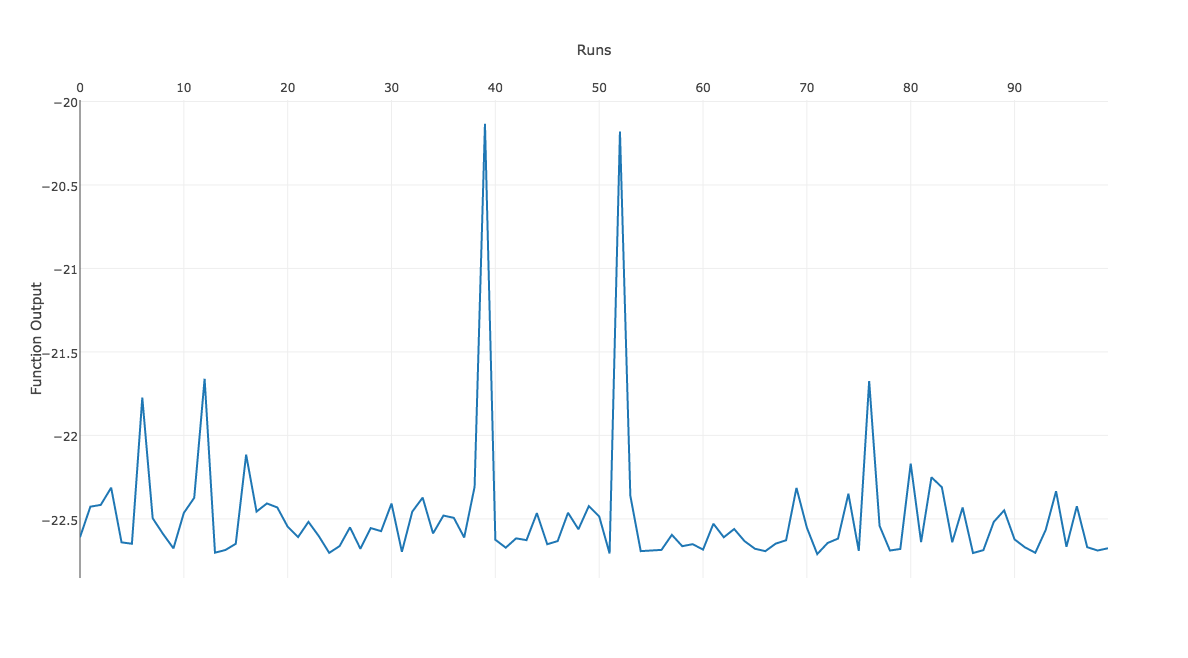
Final x,y values are -- x is : -0.0450856 -- y is -0.00371386 -- Its Ackley is : -22.5367

Final x,y values are -- x is : 0.0753578 -- y is -0.000208557 -- Its Ackley is : -22.3608

Final x,y values are -- x is : 0.0417065 -- y is 0.164024 -- Its Ackley is : -21.6231

Final x,y values are -- x is : 0.0570074 -- y is 0.131464 -- Its Ackley is : -21.8404

Final x,y values are -- x is : -0.0526748 -- y is 0.00439487 -- Its Ackley is : -22.4961



**Analysis**

By referring the above plots we can see that differential evolution is efficient and best method for finding the minimum of ackleys function. Hill Climbing algorithm almost gets stuck at the local minimum’s, where as differential evolution almost converges to minimum in each run. If we increase number of cycles in the differential evolution, then it will converge to minimum for sure. But on the other side there is no gurantee that hill climbing algorithm will produce minimum function value

1. **Output of n-queens hill climbing algorithm**

Below is the output for n-queens hill climbing algorithm. Initial and final positions of queens are displayed for each run. If zero queens attacking each other positions are found in same run, those will also be displayed and they will be same as final positions. After 10 runs only zero queens attacking positions will be displayed along with run number. Also at last total number of optimizations(Zero queen attacking positions) are displayed and also minimum fitness found in all runs will be displayed. Same applies for 16 queens and also any number of queens. Sample output for 16 queens zero positions are displayed after 8 queens output

Input should be given by user for number of queens

MacBook-Pro-Ravi:Assignment\_2 Sai$ ./a.out

Enter n for number of queens :

8

Run Number is - 0

Initial Positions are

. . . . . . . .

. . . . . . . .

. . . \* . . \* .

. . . . . . . .

\* . . . . . . .

. . . . . \* . .

. \* . . . . . .

. . \* . \* . . \*

4 6 7 2 7 5 2 7

Zero positions are found as below in run : 0

. . \* . . . . .

. . . . . . . \*

. . . \* . . . .

. . . . . . \* .

\* . . . . . . .

. . . . . \* . .

. \* . . . . . .

. . . . \* . . .

4 6 0 2 7 5 3 1

Final Positions are

. . \* . . . . .

. . . . . . . \*

. . . \* . . . .

. . . . . . \* .

\* . . . . . . .

. . . . . \* . .

. \* . . . . . .

. . . . \* . . .

4 6 0 2 7 5 3 1

Run Number is - 1

Initial Positions are

. . . . . . . .

. . . . . . . .

. \* . \* . \* \* .

. . . . . . . \*

. . . . . . . .

. . . . \* . . .

. . \* . . . . .

\* . . . . . . .

7 2 6 2 5 2 2 3

Final Positions are

. . . . . . \* .

. \* . . . . . .

. . . \* . . . .

. . . . . . . \*

. . . . . . . .

. . . . \* . . .

. . \* . . . . .

\* . . . . \* . .

7 1 6 2 5 7 0 3

Run Number is - 2

Initial Positions are

\* \* . . . . \* .

. . . . . . . .

. . . . . \* . .

. . . . \* . . .

. . . . . . . .

. . . \* . . . .

. . . . . . . \*

. . \* . . . . .

0 0 7 5 3 2 0 6

Final Positions are

. . . . \* . \* .

\* . . . . . . .

. . . . . \* . .

. . . . . . . .

. \* . . . . . .

. . . \* . . . .

. . . . . . . \*

. . \* . . . . .

1 4 7 5 0 2 0 6

Run Number is - 3

Initial Positions are

. . . . . . . .

\* \* . . \* . . \*

. . \* . . \* \* .

. . . . . . . .

. . . . . . . .

. . . . . . . .

. . . . . . . .

. . . \* . . . .

1 1 2 7 1 2 2 1

Zero positions are found as below in run : 3

. \* . . . . . .

. . . . \* . . .

. . . . . . \* .

\* . . . . . . .

. . \* . . . . .

. . . . . . . \*

. . . . . \* . .

. . . \* . . . .

3 0 4 7 1 6 2 5

Final Positions are

. \* . . . . . .

. . . . \* . . .

. . . . . . \* .

\* . . . . . . .

. . \* . . . . .

. . . . . . . \*

. . . . . \* . .

. . . \* . . . .

3 0 4 7 1 6 2 5

Run Number is - 4

Initial Positions are

. . . . . . . .

. . . . . \* \* .

. . \* \* . . . .

. . . . . . . .

. . . . . . . .

. \* . . \* . . .

. . . . . . . \*

\* . . . . . . .

7 5 2 2 5 1 1 6

Final Positions are

. . . \* . . . .

. . . . . . \* .

. . \* . . . . .

. . . . . \* . .

. \* . . . . . .

. . . . \* . . .

. . . . . . . \*

\* . . . . . . .

7 4 2 0 5 3 1 6

Run Number is - 5

Initial Positions are

. . . . . . . .

. . \* . . . \* .

. . . \* . . . .

. \* . . . \* . \*

\* . . . . . . .

. . . . . . . .

. . . . . . . .

. . . . \* . . .

4 3 1 2 7 3 1 3

Final Positions are

\* . . . . . . .

. . . . . . \* .

. . . \* . . . .

. . . . . \* . \*

. . \* . . . . .

. \* . . . . . .

. . . . . . . .

. . . . \* . . .

0 5 4 2 7 3 1 3

Run Number is - 6

Initial Positions are

. . . . \* . . .

. . . . . . . .

\* . . . . . . .

. . . . . . . .

. \* . . . . . .

. . . . . . \* \*

. . . . . . . .

. . \* \* . \* . .

2 4 7 7 0 7 5 5

Final Positions are

. . . . \* . . .

. . . . . \* . .

\* . . . . . . .

. . . . . . . .

. \* . . . . . .

. . . . . . \* \*

. . . . . . . .

. . \* \* . . . .

2 4 7 7 0 1 5 5

Run Number is - 7

Initial Positions are

. . . . . . . .

. . . . . . . .

. . \* . . . . .

. . . . \* . . .

\* . . \* . . . .

. . . . . \* \* \*

. \* . . . . . .

. . . . . . . .

4 6 2 4 3 5 5 5

Final Positions are

. . . \* . . . .

. . . . . \* . .

. . \* . . . . .

. . . . \* . . .

\* . . . . . . .

. . . . . . . \*

. \* . . . . . .

. . . . . . \* .

4 6 2 0 3 1 7 5

Run Number is - 8

Initial Positions are

. \* . . \* . . .

. . . . . \* \* \*

. . . . . . . .

. . \* . . . . .

. . . . . . . .

\* . . . . . . .

. . . . . . . .

. . . \* . . . .

5 0 3 7 0 1 1 1

Final Positions are

. \* . . \* . . .

. . . . . . . \*

. . . . . \* . .

. . . . . . . .

. . \* . . . . .

\* . . . . . . .

. . . . . . \* .

. . . \* . . . .

5 0 4 7 0 2 6 1

Run Number is - 9

Initial Positions are

. . . . . . . .

. \* . . . . . .

\* . . . . . . .

. . . . \* . . .

. . \* . . . . .

. . . \* . . . \*

. . . . . . \* .

. . . . . \* . .

2 1 4 5 3 7 6 5

Final Positions are

. . . . . \* . .

\* \* . . . . . .

. . . . . . \* .

. . . . \* . . .

. . \* . . . . .

. . . . . . . \*

. . . . . . . .

. . . \* . . . .

1 1 4 7 3 0 2 5

Zero positions are found as below in run : 11

. . \* . . . . .

. . . . \* . . .

. . . . . . \* .

\* . . . . . . .

. . . \* . . . .

. \* . . . . . .

. . . . . . . \*

. . . . . \* . .

3 5 0 4 1 7 2 6

Zero positions are found as below in run : 22

. . . . . . \* .

. . . \* . . . .

. \* . . . . . .

. . . . . . . \*

. . . . . \* . .

\* . . . . . . .

. . \* . . . . .

. . . . \* . . .

5 2 6 1 7 4 0 3

Zero positions are found as below in run : 34

. . \* . . . . .

. . . . . \* . .

. \* . . . . . .

. . . . . . \* .

. . . . \* . . .

\* . . . . . . .

. . . . . . . \*

. . . \* . . . .

5 2 0 7 4 1 3 6

Zero positions are found as below in run : 36

. . . . . \* . .

. . \* . . . . .

\* . . . . . . .

. . . . . . . \*

. . . . \* . . .

. \* . . . . . .

. . . \* . . . .

. . . . . . \* .

2 5 1 6 4 0 7 3

Zero positions are found as below in run : 39

. . . \* . . . .

. . . . . . . \*

\* . . . . . . .

. . . . \* . . .

. . . . . . \* .

. \* . . . . . .

. . . . . \* . .

. . \* . . . . .

2 5 7 0 3 6 4 1

Zero positions are found as below in run : 41

. . . \* . . . .

. . . . . \* . .

. . . . . . . \*

. \* . . . . . .

. . . . . . \* .

\* . . . . . . .

. . \* . . . . .

. . . . \* . . .

5 3 6 0 7 1 4 2

Zero positions are found as below in run : 55

. . . . \* . . .

. . \* . . . . .

\* . . . . . . .

. . . . . . \* .

. \* . . . . . .

. . . . . . . \*

. . . . . \* . .

. . . \* . . . .

2 4 1 7 0 6 3 5

Zero positions are found as below in run : 56

. . . . . \* . .

\* . . . . . . .

. . . . \* . . .

. \* . . . . . .

. . . . . . . \*

. . \* . . . . .

. . . . . . \* .

. . . \* . . . .

1 3 5 7 2 0 6 4

Zero positions are found as below in run : 58

. . . \* . . . .

. . . . . . \* .

\* . . . . . . .

. . . . . . . \*

. . . . \* . . .

. \* . . . . . .

. . . . . \* . .

. . \* . . . . .

2 5 7 0 4 6 1 3

Zero positions are found as below in run : 60

. . . \* . . . .

. . . . . . . \*

\* . . . . . . .

. . \* . . . . .

. . . . . \* . .

. \* . . . . . .

. . . . . . \* .

. . . . \* . . .

2 5 3 0 7 4 6 1

Zero positions are found as below in run : 74

. . . . \* . . .

. . . . . . \* .

\* . . . . . . .

. . \* . . . . .

. . . . . . . \*

. . . . . \* . .

. . . \* . . . .

. \* . . . . . .

2 7 3 6 0 5 1 4

Zero positions are found as below in run : 76

. . . \* . . . .

. . . . . . . \*

. . . . \* . . .

. . \* . . . . .

\* . . . . . . .

. . . . . . \* .

. \* . . . . . .

. . . . . \* . .

4 6 3 0 2 7 5 1

Zero positions are found as below in run : 77

\* . . . . . . .

. . . . . . \* .

. . . \* . . . .

. . . . . \* . .

. . . . . . . \*

. \* . . . . . .

. . . . \* . . .

. . \* . . . . .

0 5 7 2 6 3 1 4

Zero positions are found as below in run : 79

. . \* . . . . .

. . . . . . . \*

. . . \* . . . .

. . . . . . \* .

\* . . . . . . .

. . . . . \* . .

. \* . . . . . .

. . . . \* . . .

4 6 0 2 7 5 3 1

Zero positions are found as below in run : 84

. . . . . \* . .

. . . \* . . . .

. \* . . . . . .

. . . . . . . \*

. . . . \* . . .

. . . . . . \* .

\* . . . . . . .

. . \* . . . . .

6 2 7 1 4 0 5 3

Zero positions are found as below in run : 91

. . . . . \* . .

. . \* . . . . .

\* . . . . . . .

. . . . . . \* .

. . . . \* . . .

. . . . . . . \*

. \* . . . . . .

. . . \* . . . .

2 6 1 7 4 0 3 5

Zero positions are found as below in run : 95

. . . . . \* . .

. . . \* . . . .

. . . . . . \* .

\* . . . . . . .

. . \* . . . . .

. . . . \* . . .

. \* . . . . . .

. . . . . . . \*

3 6 4 1 5 0 2 7

Zero positions are found as below in run : 96

. . . . \* . . .

. . . . . . . \*

. . . \* . . . .

\* . . . . . . .

. . \* . . . . .

. . . . . \* . .

. \* . . . . . .

. . . . . . \* .

3 6 4 2 0 5 7 1

Zero positions are found as below in run : 97

\* . . . . . . .

. . . . . . \* .

. . . \* . . . .

. . . . . \* . .

. . . . . . . \*

. \* . . . . . .

. . . . \* . . .

. . \* . . . . .

0 5 7 2 6 3 1 4

Min fitness on the whole is : 0

Number of optimizations are : 21

**16-Queens Output**

Zero positions are found as below in run : 18

. . . . . . . \* . . . . . . . .

\* . . . . . . . . . . . . . . .

. . . . . . . . . . \* . . . . .

. . . . . . \* . . . . . . . . .

. . . . \* . . . . . . . . . . .

. . . . . . . . . . . \* . . . .

. . . . . . . . . . . . . . . \*

. . . . . . . . . . . . \* . . .

. . \* . . . . . . . . . . . . .

. . . . . . . . . . . . . \* . .

. . . . . . . . \* . . . . . . .

. . . \* . . . . . . . . . . . .

. \* . . . . . . . . . . . . . .

. . . . . . . . . . . . . . \* .

. . . . . . . . . \* . . . . . .

. . . . . \* . . . . . . . . . .

1 12 8 11 4 15 3 0 10 14 2 5 7 9 13 6

Zero positions are found as below in run : 36

. . . . . \* . . . . . . . . . .

. . . . . . . . . \* . . . . . .

. . . . \* . . . . . . . . . . .

. . . . . . . . . . . . . . . \*

. . . . . . . . . . \* . . . . .

. . . . . . . . . . . . \* . . .

. \* . . . . . . . . . . . . . .

. . . . . . \* . . . . . . . . .

. . . . . . . . \* . . . . . . .

. . . . . . . . . . . . . \* . .

. . \* . . . . . . . . . . . . .

\* . . . . . . . . . . . . . . .

. . . \* . . . . . . . . . . . .

. . . . . . . . . . . . . . \* .

. . . . . . . \* . . . . . . . .

. . . . . . . . . . . \* . . . .

11 6 10 12 2 0 7 14 8 1 4 15 5 9 13 3

Min fitness on the whole is : 0

Number of optimizations are : 3

1. **Output of n-queens Genetic Algorithm**

Here only zero queens attacking positions will be shown as output. User should enter required input such as population, number of queens and runs. Below is sample output for 16 queens. Number of optimizations are also shown at last.

MacBook-Pro-Ravi:Assignment\_2 Sai$ python gan.py

Enter population : 1000

Enter number of queens : 16

Enter number of runs : 100

Zero fitness found for child chromosome - [6, 11, 7, 2, 0, 12, 14, 8, 10, 13, 3, 5, 15, 1, 9, 4] Run number is - 2

Zero fitness found for child chromosome - [11, 8, 2, 15, 9, 7, 14, 0, 6, 1, 13, 5, 3, 12, 10, 4] Run number is - 21

Zero fitness found for child chromosome - [7, 12, 1, 13, 10, 5, 3, 8, 4, 11, 14, 0, 15, 6, 9, 2] Run number is - 47

Zero fitness found for child chromosome - [10, 8, 4, 14, 12, 2, 7, 13, 0, 5, 1, 11, 15, 6, 9, 3] Run number is - 54

Zero fitness found for child chromosome - [3, 5, 11, 2, 14, 12, 4, 1, 13, 6, 10, 0, 7, 9, 15, 8] Run number is - 99

Total Minimum fitness found is : 0

Total number of optimizations are : 5