

```
inder@inderpreet-singh MINGW64 /d/DAA
```

```
$ python 1.py  
Inderpreet Singh  
2104118
```

```
Enter the roll no.: 505
```

```
Roll numbers in database: [101, 202, 303, 404, 505, 606, 707, 808, 909]  
Roll number 505 found at index 4.
```

```
inder@inderpreet-singh MINGW64 /d/DAA
```

```
$ python 2.py  
Inderpreet Singh  
2104118
```

```
Sorted Roll Numbers: [1, 1, 4, 4, 7, 9, 9, 9, 12, 12, 13, 13, 14, 16, 16, 1  
8, 18, 18, 19, 19, 19, 20, 20, 21, 22, 22, 22, 24, 24, 28, 29, 30, 30, 33,  
34, 37, 37, 37, 37, 38, 40, 43, 43, 45, 46, 47, 50, 50, 50, 50]  
Time taken to sort: 0.0000000000 seconds
```

```
inder@inderpreet-singh MINGW64 /d/DAA
```

```
$ python 3.py  
Inderpreet Singh  
2104118
```

```
Sorted Roll Numbers: [3, 9, 10, 10, 12, 13, 14, 15, 15, 17, 19, 20, 21,  
21, 21, 21, 23, 24, 25, 25, 26, 26, 27, 27, 27, 28, 29, 30, 30, 31, 32,  
33, 34, 34, 35, 35, 38, 39, 42, 43, 43, 44, 46, 46, 46, 46, 48, 48, 49,  
49]  
Time taken to sort: 0.0000000000 seconds
```

```
inder@inderpreet-singh MINGW64 ~/OneDrive/Desktop/Python3
```

```
$ python 5.py  
Enter the number of cities: 4  
Enter the number of connections: 5  
Enter the connections in the format: 'city1 city2 cost'  
1 2 56  
1 3 45  
2 3 46  
2 1 64  
0 3 32  
The minimum cost to connect all cities: 123
```

```
inder@inderpreet-singh MINGW64 /d/DAA
```

```
$ python 10.py  
Inderpreet Singh  
2104118
```

```
Maximum values: 220  
Selected items: [1, 2]
```

```
inder@inderpreet-singh MINGW64 ~/OneDrive/Desktop/Python3
```

```
$ python Kruskal\s\ Algorithm.py  
Enter the number of engineering Colleges: 4  
Enter the number of connections: 5  
Enter connection in the format: 'college1 college2 cost'  
1 2 5  
4 5 45  
1 3 45  
1 4 44  
2 3 46  
Minimum cost to connect all engineering colleges: 5
```

```
inder@inderpreet-singh MINGW64 /d/DAA
```

```
$ python 8.py  
The shortest distance from Home to College is 5 units.
```

```
inder@inderpreet-singh MINGW64 /d/DAA
```

```
$ python 7.py  
Inderpreet Singh  
2104118
```

```
Minimum Route: ((0, 0), (1, 2), (2, 4), (3, 1))  
Minimum Distance: 7.63441361516796
```

```
inder@inderpreet-singh MINGW64 /d/DAA
```

```
$ python 4.py  
Inderpreet Singh  
2104118
```

```
220
```

```
inder@inderpreet-singh MINGW64 /d/DAA
$ python 12.py
Inderpreet Singh
2104118
```

```
Minimum distance between different cities:
Distance from City A to City B: 10
Distance from City A to City C: 15
Distance from City A to City D: 15
Distance from City A to City E: 20
Distance from City B to City A: 10
Distance from City B to City C: 14
Distance from City B to City D: 5
Distance from City B to City E: 17
Distance from City C to City A: 15
Distance from City C to City B: 14
Distance from City C to City D: 9
Distance from City C to City E: 10
Distance from City D to City A: 15
Distance from City D to City B: 5
Distance from City D to City C: 9
Distance from City D to City E: 12
Distance from City E to City A: 20
Distance from City E to City B: 17
Distance from City E to City C: 10
Distance from City E to City D: 12
```

```
inder@inderpreet-singh MINGW64 /d/DAA
$ python 13.py
Inderpreet Singh
2104118
```

```
[ 'Q', '.', '.', '.', '.', 'Q', '.', '.', '.' ]
[ '.', '.', '.', '.', 'Q', '.', '.', '.' ]
[ '.', '.', '.', '.', 'Q', '.', '.', 'Q' ]
[ '.', '.', 'Q', '.', '.', 'Q', '.', '.' ]
[ '.', 'Q', '.', '.', '.', 'Q', '.', '.' ]
[ '.', 'Q', '.', 'Q', '.', '.', 'Q', '.' ]
[ '.', 'Q', '.', 'Q', '.', '.', 'Q', '.' ]
```

```
inder@inderpreet-singh MINGW64 /d/DAA
$ python 14.py
Inderpreet Singh
2104118
```

```
Enter number of elements: 4
Enter elements:
4
5
6
5
Enter required Sum: 10
[4, 6]
[5, 5]
```

```
inder@inderpreet-singh MINGW64 /d/DAA
$ python 11.py
Inderpreet Singh
2104118
```

Shortest path length: 5

```
inder@inderpreet-singh MINGW64 /d/DAA
$ python 15.py
Inderpreet Singh
2104118
```

```
BFS starting from node 2
Visited: 2 -> Visited: 0 -> Visited: 3 -> Visited: 1 ->
```

```
inder@inderpreet-singh MINGW64 /d/DAA
$ python 15.py
Inderpreet Singh
2104118
```

```
Following is the Breadth-First Search
A B C D E F
```

```
inder@inderpreet-singh MINGW64 /d/DAA
$ python 16.py
Inderpreet Singh
2104118
```

```
Following is the Depth-First Search
A
B
D
E
F
C
```

```
inder@inderpreet-singh MINGW64 /d/DAA
$ python 9.py
Inderpreet Singh
2104118
```

The shortest distance from home to college is 3.

```
inder@inderpreet-singh MINGW64 /d/DAA
$ python 17.py
Inderpreet Singh
2104118
```

```
Brute Force: Pattern found at index 7
Rabin-Karp: Pattern found at index 7
KMP: Pattern found at index 7
Boyer-Moore: Pattern found at index 7
```