1. EmployeeNode readEmployees(EmployeeNode emp, int id):
   1. This function reads the employee ids from input.txt and creates a balanced binary search tree.
   2. This function is called at the beginning of the main before options to run other operations are shown.
   3. For showing operations input.txt from the problem statement is used to create binary search tree
2. int getHeadcount(EmployeeNode emp):

Output - Head count: 4

1. boolean searchID(EmployeeNode emp, int id):
   1. Input - Enter employee id: 41

Output - Employee 41 is present in organization.

* 1. Input - Enter employee id: 125

Output - Employee 125 is not present in organization.

1. int howOften(EmployeeNode emp, int id):
   1. Input - Enter employee id: 23

Output - Number of times employee 23 entered the organization: 1

* 1. Input - Enter employee id: 41

Output - Number of times employee 41 entered the organization: 3

* 1. Input - Enter employee id: 100

Output - Number of times employee 100 entered the organization: 0

1. EmployeeNode frequentVisitor(EmployeeNode emp):

Output - Employee 41 is the most frequent visitor with 3 visits.

1. void printRangePresent(Employees emp, int id1, int id2):
   1. Input - Enter range of emp ids: 23 125

Output – Contents of output.txt are as below

23, 1

41, 3

121, 2

* 1. Input - Enter range of emp ids: 30 100

Output – Contents of output.txt are as below

41, 3