Page no.:____/
Date:__/__/____/

// Algorithm to create a csv file and randomly generate Name and Salary.

String { generateRandomName()

// Generate a Name :- To randomly generate a name and add it to csv. file

// input :- We give input as a void

// output :- We return the firstname and last name.

vector <string> firstname ← { John, Arthur ... }
vector< string> lastname ← { Bhoge, Wick, ... }

fn = firstname [rand() % fn.size];
ln = lastname [rand() % ln.size];

int getRandomNo()

// Generate a Number :- Randomly generate a salary and add it to csv file.
// input :- We dont take any input
// output :- We return a randomly generated number.

return { rand() % 90001 + 10000

Page no.: ____/
Date: ___/___/_____

// To calculate ~~Gross~~ taxes, Home Rent, Bonus.
// and store it in another CSV file.

//Calculate taxes, HR, Bonus :- Calculate the Taxes, HR, Bonus bonus of all employees

//Input :- We give inputfile, outputfile and a vector

// output :- We get a new CSV file with calculated taxes, HR, Bonus.

```
if ( ! inputfile.is_open() or !! outputfis.open)
    cout << " Error ";

line;
outputfile = " Name, Salary, Taxes, HR, Bonus"

getline ( inputfile, line );

while ( getline ( inputfile, line) ) do

    string-stream ss (line) ;
    getline ( ss, name, ',' )
    getline ( ss, salary, ',' )

    do tax = 0.10 * salary ;
       HR = 0.20 * Salary ;
       bonus = 0.15 * salary ;
outputfile << name << "," << Salary << "," << tax << HR
                                          << Bonus ;
inputfile.close ; outputfile.close
```

Page no.: ____/
Date: ____/____/____/

// Divide and conquer

int double findMin (vector a, int l, int h)

// Find the min salary :- To find the minimum
// salary for the company employee

// ~~Output~~ Input :- A vector, a starting point and a, ending point

// Output :- Return a double value minimum salary.

if (lo > h)
    return -1;
if (lo == h)
    return a[l]

mid ← (l + h) / 2

~~int Lmax =~~
Lmin = findMin (v, lo, mid)
R-~~min~~ = findMin (v, mid+1, h)

return min (Lmin, Rmin);

// Similarly for max

return max (Lmax, Rmin max);

Page no.:
Date: / /

Testcases :-

	input file :	output
1)	1 input	min → 10008 max → 42755
2)	2 input	10019 min → 10002 max → 42715
3)	3. input	error opening file
4)	4 - input	min → 10005 max → 42702
5)	5 - input	Salary can't be negative.

Time - Complexity :-

Time $O(n) = 2(n/2) + O(1)$

$= O(n) + 1$

Space Complexity $= O(\log n)$