

* Algorithm

1) Creating CSV file

// Aim: Generating & adding random name & salary

```
String generateRandomName ()
```

```
{
```

```
    vector <String> a = { // first names }
```

```
    vector <String> b = { // last names }
```

```
}
```

```
    String a = a [rand % a.size ()]
```

```
    // a is name of string of  
    first name.
```

```
    String b = b [rand % b.size ()]
```

```
    // b is name of string of last names
```

```
int generateRandomSalary
```

```
{ generate random salary }
```

// input : input is given as void .

// Output : output returns first & last
name .

2) Calculating Salary:

// Aim: To calculate taxes, HR, bonus and store it in csv file.

// input: input file, output file & vector

```
if (!inputfile.is_open() || !outputfile.is_open())  
{ // error }
```

```
outputfile << "Name, salary, Tax,  
HR, bonus";
```

```
getline(inputfile, line);
```

```
while (getline(inputfile, line))  
{
```

```
    stringstream ss(line);
```

```
    getline(ss, name, ',');
```

```
    getline(ss, salary, ','); // similar for salary as name
```

```
    tax = 0.15 * salary;
```

```
    HR = 0.18 * salary;
```

```
    bonus = 0.11 * salary;
```

```
    outputfile << name, salary, tax, HR, bonus;
```

```
Close input & output file;
```


// function to find maximum no.
int findmax (int a[], int lo, int hi)

{

// If lo become greater than hi, return
minimum

if (lo > hi)

return int_Min;

// if subarray has only one element,
return element.

if (lo == hi)

return a[lo];

int mid = (lo + hi) / 2;

int leftMax = findMax (a, lo, mid);

int rightMax = findMax (a, mid+1, hi);

return max (leftMax, rightMax).

// function to find minimum no
int findmin (int a[], int lo, int hi)
{

// If hi becomes greater than lo, return maximum.

if (hi > lo)

return int - Max;

if (lo == hi)

return a[hi];

int mid = (lo + hi) / 2;

int leftMin = findmin (a, ~~lo~~^{hi}, mid);

int rightMin = findmin (a, mid+1, lo)

return Min (leftMin, rightMin)

★ Test Cases:

1) Input : 1.csv
Output : min salary → 10002
max salary → 42744

2) Input : 2.csv
Output : min salary → 20020
max salary → 52748

3) Input : 3.csv
min salary → 1008
max salary → 33726

4) Input : 4.csv
Output : min salary \rightarrow 1001
max salary \rightarrow 20981

5) Input : 5.csv
Output : min salary \rightarrow 10012
max salary \rightarrow 42740

6) Input : 6.csv
Output : salary cannot be negative

7) Input : 7.csv
Output : Salary cannot be negative

8) Input : 8.csv
Output : Salary cannot be negative

* Time Complexity

By substitution :-

$$T(n) = 2^k T(N/2) + M(M)$$

where : $T(K)$ is time taken to sort K element

$$2^k T(M/2) + \text{const}^k M \quad (1)$$

$$T(M/2) = 2^k T(M/4) + \text{const} M/2 \quad (2)$$

Putting eq (2) in (1)

$$T(N) = 2^k T(N/2^k) + \text{const} \cdot \left(N + \frac{N}{2}\right)$$

$$T(N) = 2^k T(N/2^k) + \text{const} \cdot N \left(1 + 1/2\right)$$

$$T(N) = 2^k T\left(\frac{N}{2^k}\right) + \text{const} \cdot N \left(1 + \frac{1}{2} + \frac{1}{2^{k-1}}\right)$$

$$k = \log_2 N$$

$$T(N) = N (T(1) + \log_2 N \cdot \text{const})$$

$$\begin{aligned} T(N) &= N + N \log_2 N \cdot \text{const} \\ &\approx N \log_2 N \cdot \text{const} \approx N \log N \end{aligned}$$