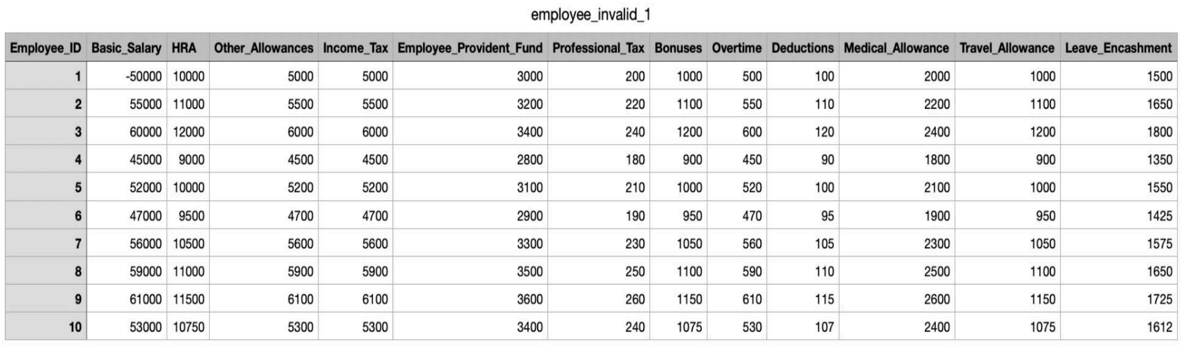
Mihir Katakdhond I 231070025 I SY BTECH COMPUTER ENGINEERING

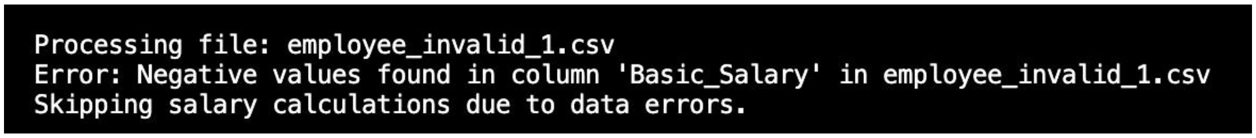
TEST CASES:

Test case 1 negative:



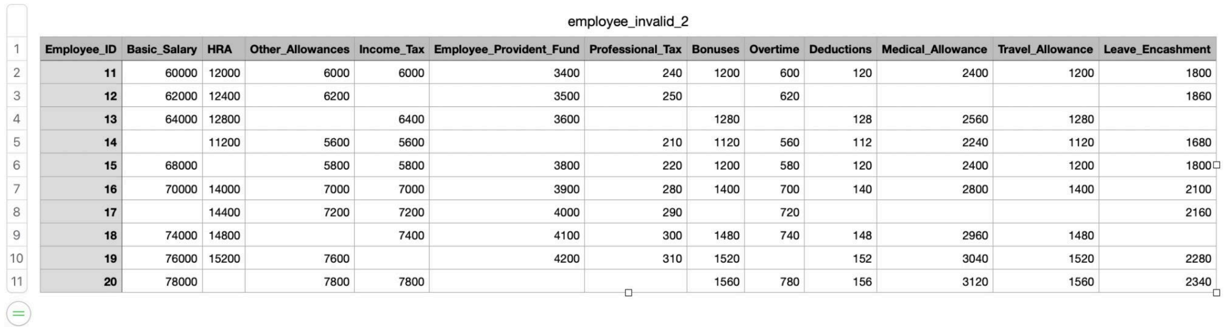
employee_invalid_1

Employee_ID	Basic_Salary	HRA	Other_Allowances	Income_Tax	Employee_Provident_Fund	Professional_Tax	Bonuses	Overtime	Deductions	Medical_Allowance	Travel_Allowance	Leave_Encashment
1	-50000	10000	5000	5000	3000	200	1000	500	100	2000	1000	1500
2	55000	11000	5500	5500	3200	220	1100	550	110	2200	1100	1650
3	60000	12000	6000	6000	3400	240	1200	600	120	2400	1200	1800
4	45000	9000	4500	4500	2800	180	900	450	90	1800	900	1350
5	52000	10000	5200	5200	3100	210	1000	520	100	2100	1000	1550
6	47000	9500	4700	4700	2900	190	950	470	95	1900	950	1425
7	56000	10500	5600	5600	3300	230	1050	560	105	2300	1050	1575
8	59000	11000	5900	5900	3500	250	1100	590	110	2500	1100	1650
9	61000	11500	6100	6100	3600	260	1150	610	115	2600	1150	1725
10	53000	10750	5300	5300	3400	240	1075	530	107	2400	1075	1612



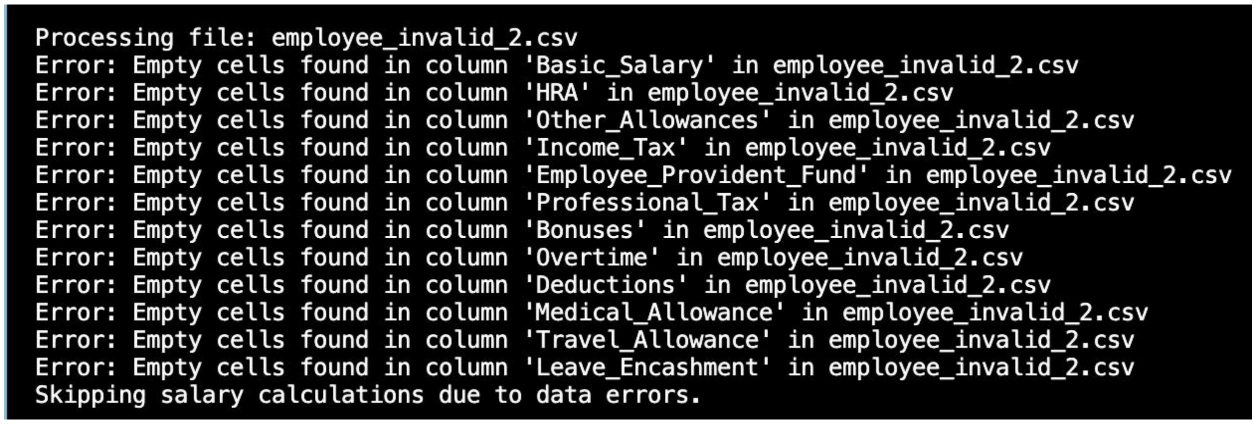
```
Processing file: employee_invalid_1.csv
Error: Negative values found in column 'Basic_Salary' in employee_invalid_1.csv
Skipping salary calculations due to data errors.
```

Test case 2 negative:



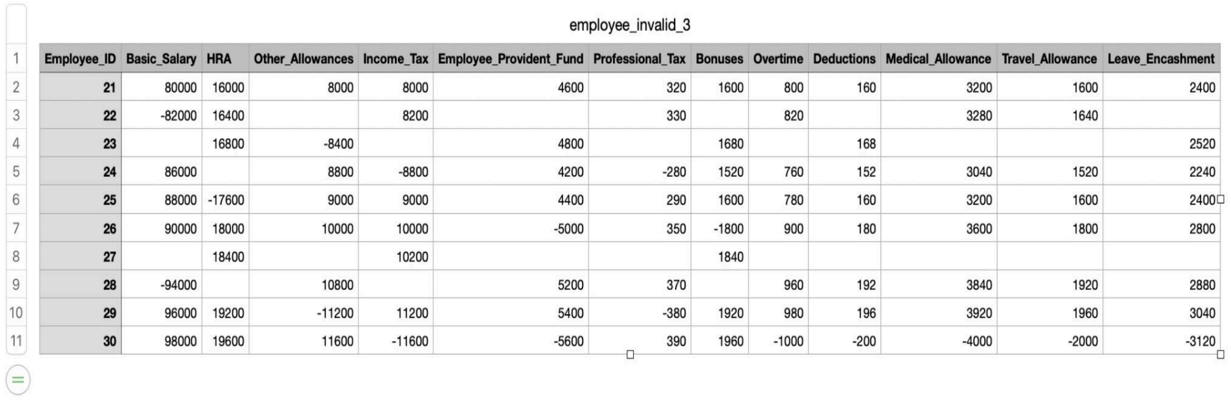
employee_invalid_2

Employee_ID	Basic_Salary	HRA	Other_Allowances	Income_Tax	Employee_Provident_Fund	Professional_Tax	Bonuses	Overtime	Deductions	Medical_Allowance	Travel_Allowance	Leave_Encashment
11	60000	12000	6000	6000	3400	240	1200	600	120	2400	1200	1800
12	62000	12400	6200		3500	250		620				1860
13	64000	12800		6400	3600		1280		128	2560	1280	
14		11200	5600	5600		210	1120	560	112	2240	1120	1680
15	68000		5800	5800	3800	220	1200	580	120	2400	1200	1800
16	70000	14000	7000	7000	3900	280	1400	700	140	2800	1400	2100
17		14400	7200	7200	4000	290		720				2160
18	74000	14800		7400	4100	300	1480	740	148	2960	1480	
19	76000	15200	7600		4200	310	1520		152	3040	1520	2280
20	78000		7800	7800			1560	780	156	3120	1560	2340



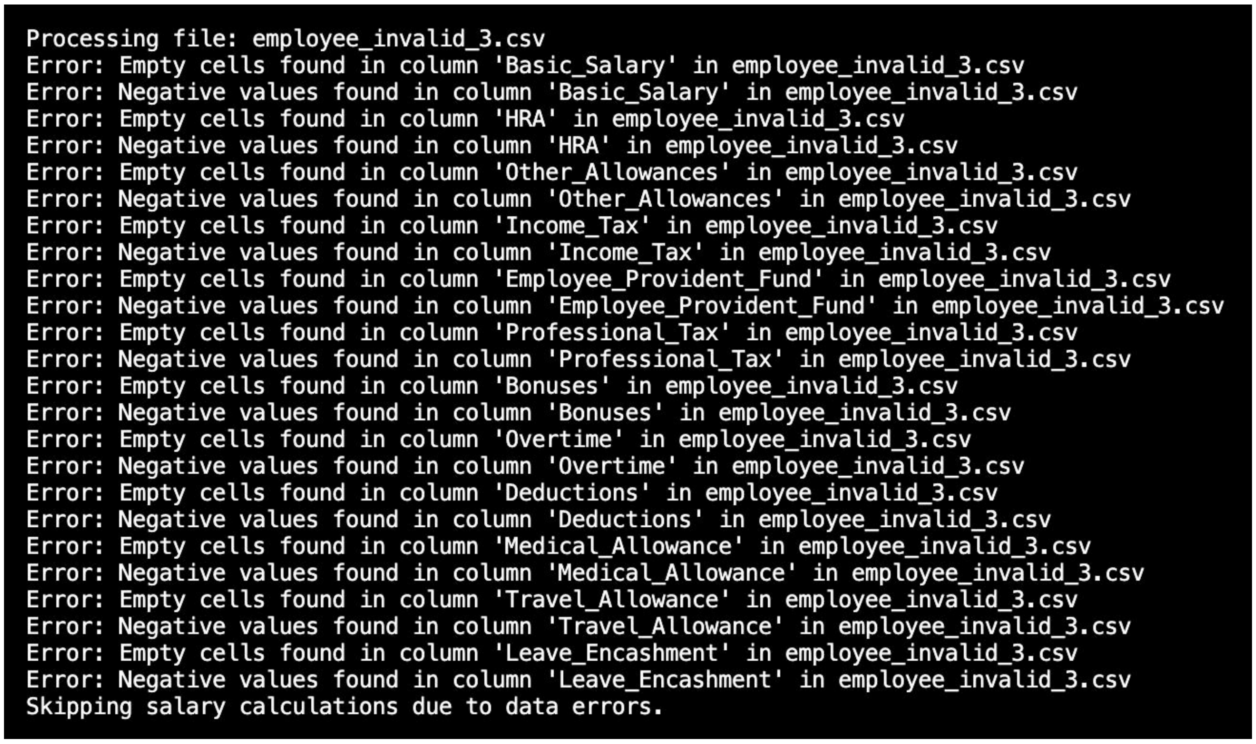
```
Processing file: employee_invalid_2.csv
Error: Empty cells found in column 'Basic_Salary' in employee_invalid_2.csv
Error: Empty cells found in column 'HRA' in employee_invalid_2.csv
Error: Empty cells found in column 'Other_Allowances' in employee_invalid_2.csv
Error: Empty cells found in column 'Income_Tax' in employee_invalid_2.csv
Error: Empty cells found in column 'Employee_Provident_Fund' in employee_invalid_2.csv
Error: Empty cells found in column 'Professional_Tax' in employee_invalid_2.csv
Error: Empty cells found in column 'Bonuses' in employee_invalid_2.csv
Error: Empty cells found in column 'Overtime' in employee_invalid_2.csv
Error: Empty cells found in column 'Deductions' in employee_invalid_2.csv
Error: Empty cells found in column 'Medical_Allowance' in employee_invalid_2.csv
Error: Empty cells found in column 'Travel_Allowance' in employee_invalid_2.csv
Error: Empty cells found in column 'Leave_Encashment' in employee_invalid_2.csv
Skipping salary calculations due to data errors.
```

Test case 3 negative:



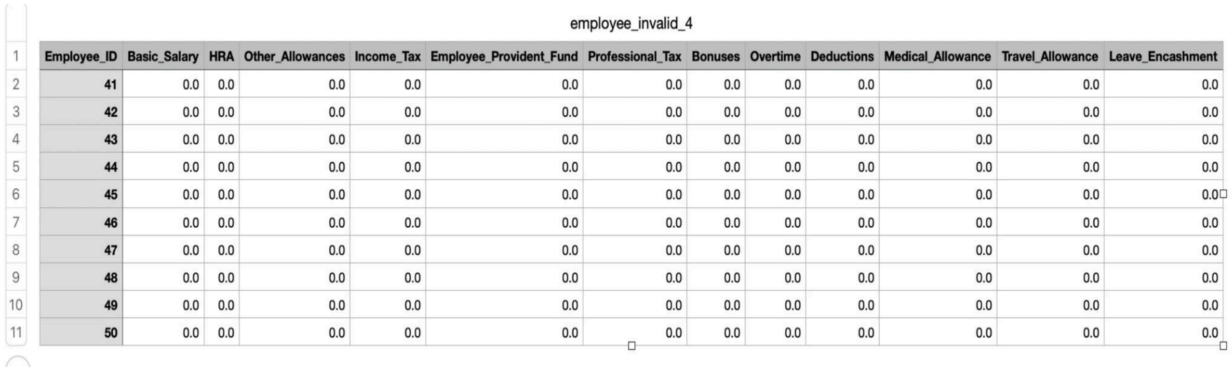
employee_invalid_3

Employee_ID	Basic_Salary	HRA	Other_Allowances	Income_Tax	Employee_Provident_Fund	Professional_Tax	Bonuses	Overtime	Deductions	Medical_Allowance	Travel_Allowance	Leave_Encashment
21	80000	16000	8000	8000	4600	320	1600	800	160	3200	1600	2400
22	-82000	16400		8200		330		820		3280	1640	
23		16800	-8400		4800		1680		168			2520
24	86000		8800	-8800	4200	-280	1520	760	152	3040	1520	2240
25	88000	-17600	9000	9000	4400	290	1600	780	160	3200	1600	2400
26	90000	18000	10000	10000	-5000	350	-1800	900	180	3600	1800	2800
27		18400		10200			1840					
28	-94000		10800		5200	370		960	192	3840	1920	2880
29	96000	19200	-11200	11200	5400	-380	1920	980	196	3920	1960	3040
30	98000	19600	11600	-11600	-5600	390	1960	-1000	-200	-4000	-2000	-3120



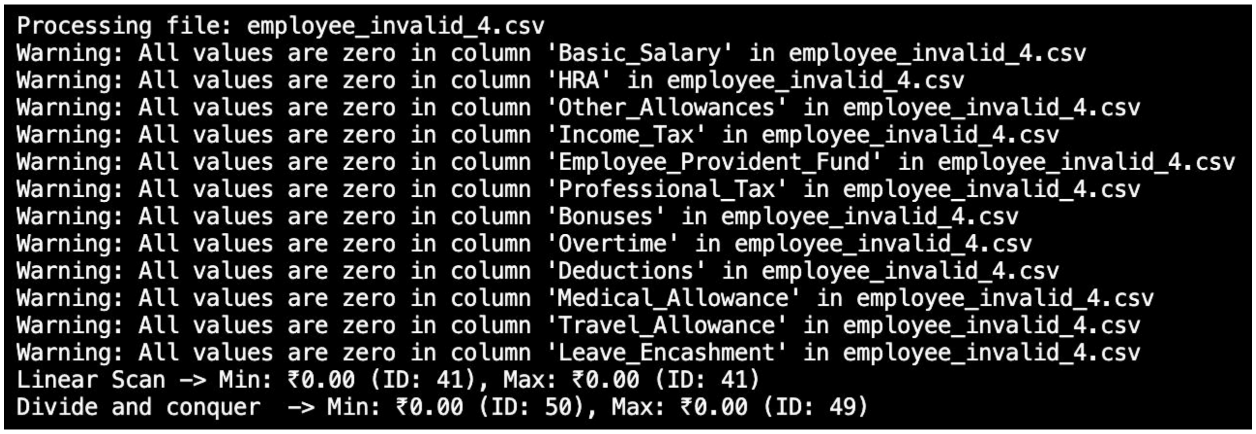
```
Processing file: employee_invalid_3.csv
Error: Empty cells found in column 'Basic_Salary' in employee_invalid_3.csv
Error: Negative values found in column 'Basic_Salary' in employee_invalid_3.csv
Error: Empty cells found in column 'HRA' in employee_invalid_3.csv
Error: Negative values found in column 'HRA' in employee_invalid_3.csv
Error: Empty cells found in column 'Other_Allowances' in employee_invalid_3.csv
Error: Negative values found in column 'Other_Allowances' in employee_invalid_3.csv
Error: Empty cells found in column 'Income_Tax' in employee_invalid_3.csv
Error: Negative values found in column 'Income_Tax' in employee_invalid_3.csv
Error: Empty cells found in column 'Employee_Provident_Fund' in employee_invalid_3.csv
Error: Negative values found in column 'Employee_Provident_Fund' in employee_invalid_3.csv
Error: Empty cells found in column 'Professional_Tax' in employee_invalid_3.csv
Error: Negative values found in column 'Professional_Tax' in employee_invalid_3.csv
Error: Empty cells found in column 'Bonuses' in employee_invalid_3.csv
Error: Negative values found in column 'Bonuses' in employee_invalid_3.csv
Error: Empty cells found in column 'Overtime' in employee_invalid_3.csv
Error: Negative values found in column 'Overtime' in employee_invalid_3.csv
Error: Empty cells found in column 'Deductions' in employee_invalid_3.csv
Error: Negative values found in column 'Deductions' in employee_invalid_3.csv
Error: Empty cells found in column 'Medical_Allowance' in employee_invalid_3.csv
Error: Negative values found in column 'Medical_Allowance' in employee_invalid_3.csv
Error: Empty cells found in column 'Travel_Allowance' in employee_invalid_3.csv
Error: Negative values found in column 'Travel_Allowance' in employee_invalid_3.csv
Error: Empty cells found in column 'Leave_Encashment' in employee_invalid_3.csv
Error: Negative values found in column 'Leave_Encashment' in employee_invalid_3.csv
Skipping salary calculations due to data errors.
```

Test case 4 negative:



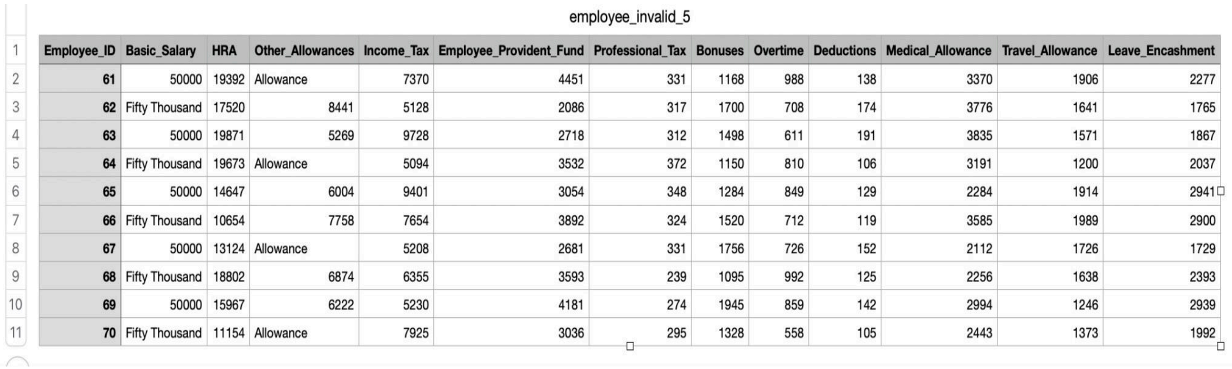
employee_invalid_4

Employee_ID	Basic_Salary	HRA	Other_Allowances	Income_Tax	Employee_Provident_Fund	Professional_Tax	Bonuses	Overtime	Deductions	Medical_Allowance	Travel_Allowance	Leave_Encashment
41	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
42	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
43	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
44	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
45	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
46	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
47	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
48	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
49	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0



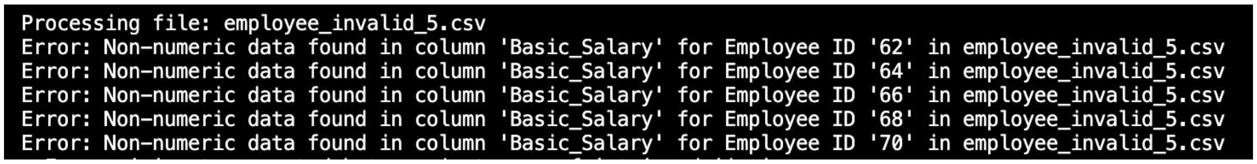
```
Processing file: employee_invalid_4.csv
Warning: All values are zero in column 'Basic_Salary' in employee_invalid_4.csv
Warning: All values are zero in column 'HRA' in employee_invalid_4.csv
Warning: All values are zero in column 'Other_Allowances' in employee_invalid_4.csv
Warning: All values are zero in column 'Income_Tax' in employee_invalid_4.csv
Warning: All values are zero in column 'Employee_Provident_Fund' in employee_invalid_4.csv
Warning: All values are zero in column 'Professional_Tax' in employee_invalid_4.csv
Warning: All values are zero in column 'Bonuses' in employee_invalid_4.csv
Warning: All values are zero in column 'Overtime' in employee_invalid_4.csv
Warning: All values are zero in column 'Deductions' in employee_invalid_4.csv
Warning: All values are zero in column 'Medical_Allowance' in employee_invalid_4.csv
Warning: All values are zero in column 'Travel_Allowance' in employee_invalid_4.csv
Warning: All values are zero in column 'Leave_Encashment' in employee_invalid_4.csv
Linear Scan -> Min: ₹0.00 (ID: 41), Max: ₹0.00 (ID: 41)
Divide and conquer  -> Min: ₹0.00 (ID: 50), Max: ₹0.00 (ID: 49)
```

Test case 5 negative:



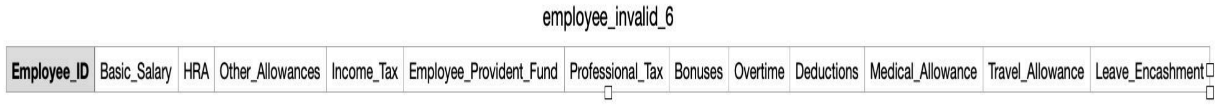
employee_invalid_5

Employee_ID	Basic_Salary	HRA	Other_Allowances	Income_Tax	Employee_Provident_Fund	Professional_Tax	Bonuses	Overtime	Deductions	Medical_Allowance	Travel_Allowance	Leave_Encashment
61	50000	19392	Allowance	7370	4451	331	1168	988	138	3370	1906	2277
62	Fifty Thousand	17520	8441	5128	2086	317	1700	708	174	3776	1641	1765
63	50000	19871	5269	9728	2718	312	1498	611	191	3835	1571	1867
64	Fifty Thousand	19673	Allowance	5094	3532	372	1150	810	106	3191	1200	2037
65	50000	14647	6004	9401	3054	348	1284	849	129	2284	1914	2941
66	Fifty Thousand	10654	7758	7654	3892	324	1520	712	119	3585	1989	2900
67	50000	13124	Allowance	5208	2681	331	1756	726	152	2112	1726	1729
68	Fifty Thousand	18802	6874	6355	3593	239	1095	992	125	2256	1638	2393
69	50000	15967	6222	5230	4181	274	1945	859	142	2994	1246	2939
70	Fifty Thousand	11154	Allowance	7925	3036	295	1328	558	105	2443	1373	1992

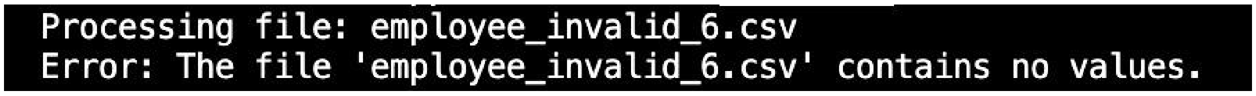


```
Processing file: employee_invalid_5.csv
Error: Non-numeric data found in column 'Basic_Salary' for Employee ID '62' in employee_invalid_5.csv
Error: Non-numeric data found in column 'Basic_Salary' for Employee ID '64' in employee_invalid_5.csv
Error: Non-numeric data found in column 'Basic_Salary' for Employee ID '66' in employee_invalid_5.csv
Error: Non-numeric data found in column 'Basic_Salary' for Employee ID '68' in employee_invalid_5.csv
Error: Non-numeric data found in column 'Basic_Salary' for Employee ID '70' in employee_invalid_5.csv
```

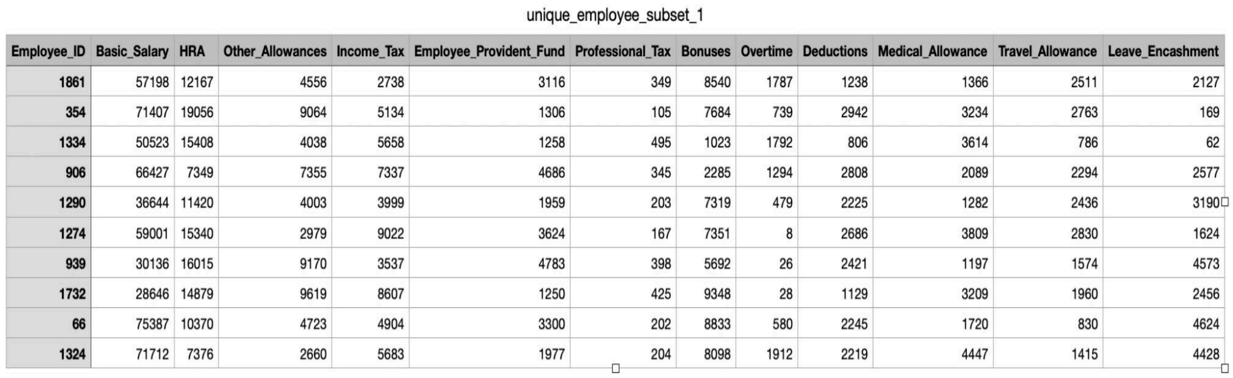
Test case 6 negative:


employee_invalid_6

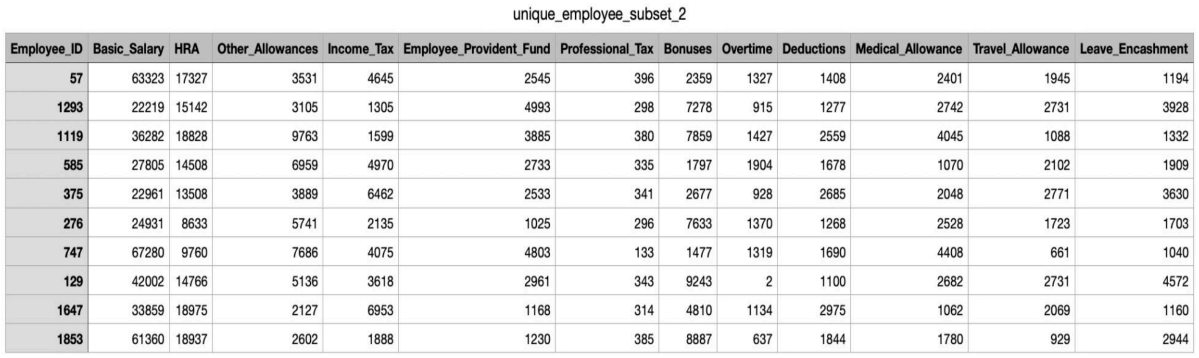
Employee_ID	Basic_Salary	HRA	Other_Allowances	Income_Tax	Employee_Provident_Fund	Professional_Tax	Bonuses	Overtime	Deductions	Medical_Allowance	Travel_Allowance	Leave_Encashment


```
Processing file: employee_invalid_6.csv
Error: The file 'employee_invalid_6.csv' contains no values.
```

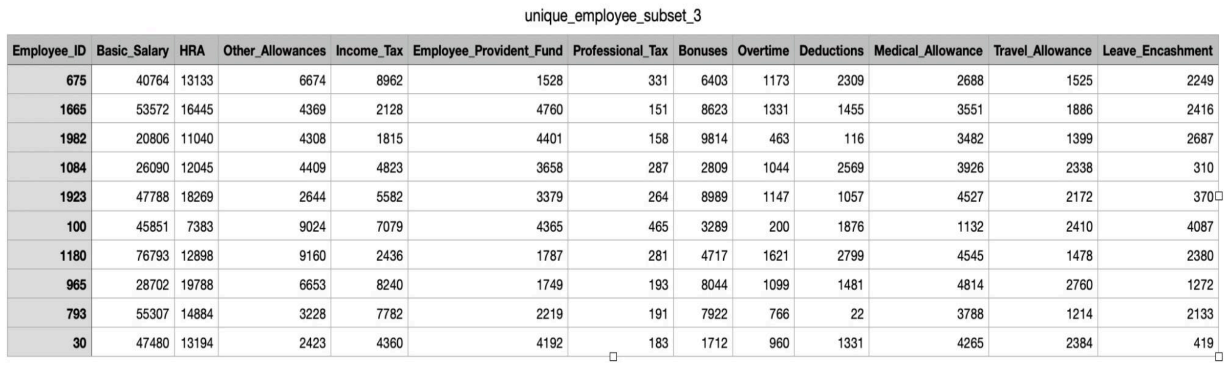
Test cases 4 positive:


unique_employee_subset_1

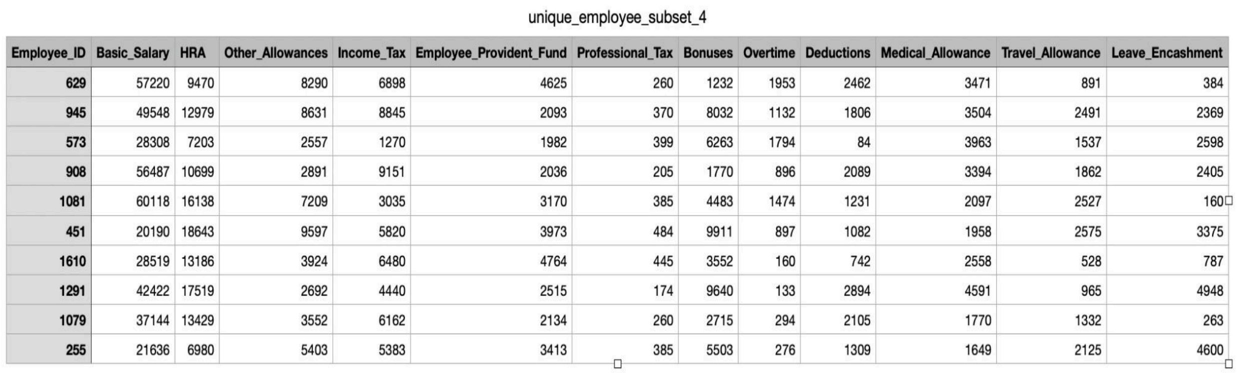
Employee_ID	Basic_Salary	HRA	Other_Allowances	Income_Tax	Employee_Provident_Fund	Professional_Tax	Bonuses	Overtime	Deductions	Medical_Allowance	Travel_Allowance	Leave_Encashment
1861	57198	12167	4556	2738	3116	349	8540	1787	1238	1366	2511	2127
354	71407	19056	9064	5134	1306	105	7684	739	2942	3234	2763	169
1334	50523	15408	4038	5658	1258	495	1023	1792	806	3614	786	62
906	66427	7349	7355	7337	4686	345	2285	1294	2808	2089	2294	2577
1290	36644	11420	4003	3999	1959	203	7319	479	2225	1282	2436	3190
1274	59001	15340	2979	9022	3624	167	7351	8	2686	3809	2830	1624
939	30136	16015	9170	3537	4783	398	5692	26	2421	1197	1574	4573
1732	28646	14879	9619	8607	1250	425	9348	28	1129	3209	1960	2456
66	75387	10370	4723	4904	3300	202	8833	580	2245	1720	830	4624
1324	71712	7376	2660	5683	1977	204	8098	1912	2219	4447	1415	4428


unique_employee_subset_2

Employee_ID	Basic_Salary	HRA	Other_Allowances	Income_Tax	Employee_Provident_Fund	Professional_Tax	Bonuses	Overtime	Deductions	Medical_Allowance	Travel_Allowance	Leave_Encashment
57	63323	17327	3531	4645	2545	396	2359	1327	1408	2401	1945	1194
1293	22219	15142	3105	1305	4993	298	7278	915	1277	2742	2731	3928
1119	36282	18828	9763	1599	3885	380	7859	1427	2559	4045	1088	1332
585	27805	14508	6959	4970	2733	335	1797	1904	1678	1070	2102	1909
375	22961	13508	3889	6462	2533	341	2677	928	2685	2048	2771	3630
276	24931	8633	5741	2135	1025	296	7633	1370	1268	2528	1723	1703
747	67280	9760	7686	4075	4803	133	1477	1319	1690	4408	661	1040
129	42002	14766	5136	3618	2961	343	9243	2	1100	2682	2731	4572
1647	33859	18975	2127	6953	1168	314	4810	1134	2975	1062	2069	1160
1853	61360	18937	2602	1888	1230	385	8887	637	1844	1780	929	2944

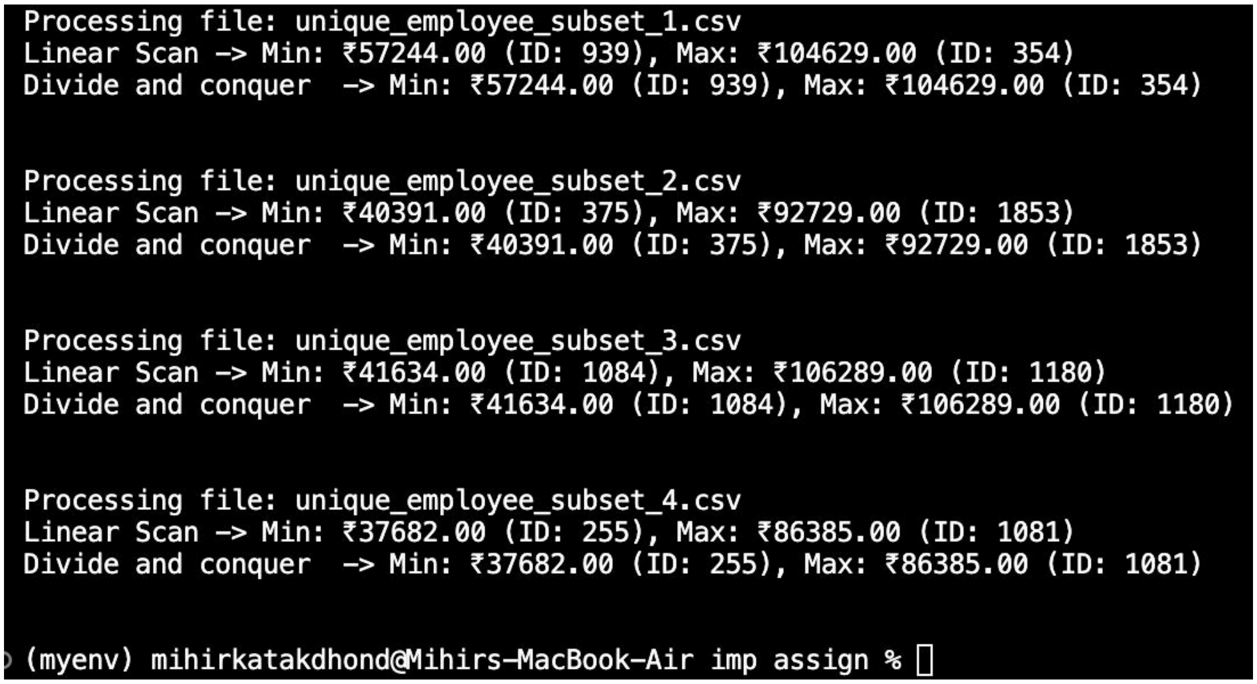

unique_employee_subset_3

Employee_ID	Basic_Salary	HRA	Other_Allowances	Income_Tax	Employee_Provident_Fund	Professional_Tax	Bonuses	Overtime	Deductions	Medical_Allowance	Travel_Allowance	Leave_Encashment
675	40764	13133	6674	8962	1528	331	6403	1173	2309	2688	1525	2249
1665	53572	16445	4369	2128	4760	151	8623	1331	1455	3551	1886	2416
1982	20806	11040	4308	1815	4401	158	9814	463	116	3482	1399	2687
1084	26090	12045	4409	4823	3658	287	2809	1044	2569	3926	2338	310
1923	47788	18269	2644	5582	3379	264	8989	1147	1057	4527	2172	370
100	45851	7383	9024	7079	4365	465	3289	200	1876	1132	2410	4087
1180	76793	12898	9160	2436	1787	281	4717	1621	2799	4545	1478	2380
965	28702	19788	6653	8240	1749	193	8044	1099	1481	4814	2760	1272
793	55307	14884	3228	7782	2219	191	7922	766	22	3788	1214	2133
30	47480	13194	2423	4360	4192	183	1712	960	1331	4265	2384	419



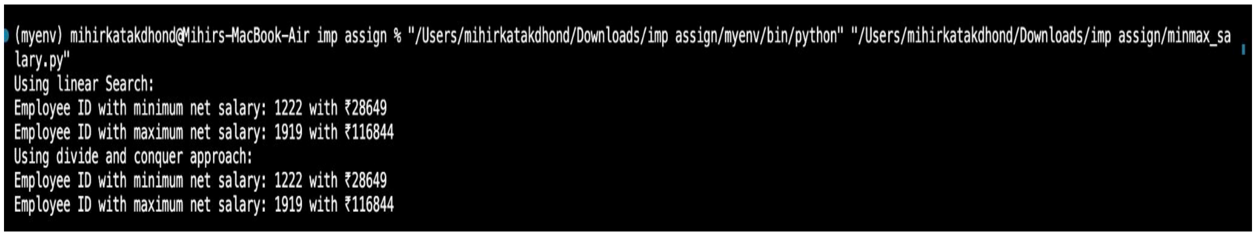
unique_employee_subset_4

Employee_ID	Basic_Salary	HRA	Other_Allowances	Income_Tax	Employee_Provident_Fund	Professional_Tax	Bonuses	Overtime	Deductions	Medical_Allowance	Travel_Allowance	Leave_Encashment
629	57220	9470	8290	6898	4625	260	1232	1953	2462	3471	891	384
945	49548	12979	8631	8845	2093	370	8032	1132	1806	3504	2491	2369
573	28308	7203	2557	1270	1982	399	6263	1794	84	3963	1537	2598
908	56487	10699	2891	9151	2036	205	1770	896	2089	3394	1862	2405
1081	60118	16138	7209	3035	3170	385	4483	1474	1231	2097	2527	160
451	20190	18643	9597	5820	3973	484	9911	897	1082	1958	2575	3375
1610	28519	13186	3924	6480	4764	445	3552	160	742	2558	528	787
1291	42422	17519	2692	4440	2515	174	9640	133	2894	4591	965	4948
1079	37144	13429	3552	6162	2134	260	2715	294	2105	1770	1332	263
255	21636	6980	5403	5383	3413	385	5503	276	1309	1649	2125	4600



```
Processing file: unique_employee_subset_1.csv
Linear Scan -> Min: ₹57244.00 (ID: 939), Max: ₹104629.00 (ID: 354)
Divide and conquer  -> Min: ₹57244.00 (ID: 939), Max: ₹104629.00 (ID: 354)


Processing file: unique_employee_subset_2.csv
Linear Scan -> Min: ₹40391.00 (ID: 375), Max: ₹92729.00 (ID: 1853)
Divide and conquer  -> Min: ₹40391.00 (ID: 375), Max: ₹92729.00 (ID: 1853)


Processing file: unique_employee_subset_3.csv
Linear Scan -> Min: ₹41634.00 (ID: 1084), Max: ₹106289.00 (ID: 1180)
Divide and conquer  -> Min: ₹41634.00 (ID: 1084), Max: ₹106289.00 (ID: 1180)


Processing file: unique_employee_subset_4.csv
Linear Scan -> Min: ₹37682.00 (ID: 255), Max: ₹86385.00 (ID: 1081)
Divide and conquer  -> Min: ₹37682.00 (ID: 255), Max: ₹86385.00 (ID: 1081)


(myenv) mihirkatakdhond@Mihirs-MacBook-Air imp assign % 
```

For 2000 employee’s:



```
(myenv) mihirkatakdhond@Mihirs-MacBook-Air imp assign % "/Users/mihirkatakdhond/Downloads/imp assign/myenv/bin/python" "/Users/mihirkatakdhond/Downloads/imp assign/minmax_sa
lary.py"
Using linear Search:
Employee ID with minimum net salary: 1222 with ₹28649
Employee ID with maximum net salary: 1919 with ₹116844
Using divide and conquer approach:
Employee ID with minimum net salary: 1222 with ₹28649
Employee ID with maximum net salary: 1919 with ₹116844
```

Mihir Katakdhond I 231070025 I SY BTECH COMPUTER ENGINEERING

Code:

```
import numpy as np
import pandas as pd

# Load the data
df = pd.read_csv('employee_salary_information_with_id.csv')

employee_id = df['Employee_ID'].values
basic_salary = df['Basic_Salary'].values
HRA = df['HRA'].values
other_allowances = df['Other_Allowances'].values
income_tax = df['Income_Tax'].values
employee_provident_fund = df['Employee_Provident_Fund'].values
professional_tax = df['Professional_Tax'].values
bonuses = df['Bonuses'].values
overtime = df['Overtime'].values
deductions = df['Deductions'].values
medical_allowance = df['Medical_Allowance'].values
travel_allowance = df['Travel_Allowance'].values
leave_encashment = df['Leave_Encashment'].values

columns_to_check = [basic_salary, HRA, other_allowances, income_tax,
employee_provident_fund, professional_tax, bonuses,
overtime, deductions, medical_allowance, travel_allowance, leave_encashment]

for i, column in enumerate(columns_to_check):
if (column < 0).any():
print(f"Error: Negative values found in column '{df.columns[i+1]}'")
print("Skipping salary calculations due to invalid data.\n")
exit()

gross_salary = basic_salary + HRA + other_allowances + bonuses + overtime +
medical_allowance + travel_allowance + leave_encashment

net_salary = gross_salary - (income_tax + employee_provident_fund + professional_tax +
deductions)

min_net_salary = net_salary[0]
max_net_salary = net_salary[0]
min_net_salary_employee_id = employee_id[0]
max_net_salary_employee_id = employee_id[0]
```

```
for i in range(1, len(net_salary)):
if net_salary[i] < min_net_salary:
min_net_salary = net_salary[i]
min_net_salary_employee_id = employee_id[i]
elif net_salary[i] > max_net_salary:
max_net_salary = net_salary[i]
max_net_salary_employee_id = employee_id[i]

print("Using linear Search:")
print(f"Employee ID with minimum net salary: {min_net_salary_employee_id} with
₹{min_net_salary}")
print(f"Employee ID with maximum net salary: {max_net_salary_employee_id} with
₹{max_net_salary}")

def MinMaxSalary(net_salary, left, right):
if left == right:
return net_salary[left], net_salary[left]
if right == left + 1:
return min(net_salary[left], net_salary[right]), max(net_salary[left],
net_salary[right])
mid = left + (right - left) // 2
min1, max1 = MinMaxSalary(net_salary, left, mid)
min2, max2 = MinMaxSalary(net_salary, mid + 1, right)
MinSalary = min(min1, min2)
MaxSalary = max(max1, max2)
return MinSalary, MaxSalary
print('Using divide and conquer approach:')
min, max = MinMaxSalary(net_salary, 0, len(net_salary) - 1)
print(f"Employee ID with minimum net salary: {min_net_salary_employee_id} with
₹{min}")
print(f"Employee ID with maximum net salary: {max_net_salary_employee_id} with
₹{max}")

test_files = [
"employee_invalid_1.csv",
"employee_invalid_2.csv",
"employee_invalid_3.csv",
"employee_invalid_4.csv",
"employee_invalid_5.csv",
"employee_invalid_6.csv",
"unique_employee_subset_1.csv",
"unique_employee_subset_2.csv",
"unique_employee_subset_3.csv",
"unique_employee_subset_4.csv"
]

def run_test_cases():
for file in test_files:
try:
print(f"Processing file: {file}")
df_test = pd.read_csv(file)

if df_test.isnull().all().all():
print(f"Error: The file '{file}' contains no values.")
continue

error_flag = False

for column in df_test.columns:
if df_test[column].isnull().any():
print(f"Error: Empty cells found in column '{column}' in {file}")
error_flag = True
if (df_test[column].fillna(0) == 0).all():
print(f"Warning: All values are zero in column '{column}' in {file}")
if df_test[column].dtype == 'object':
non_numeric_rows = df_test[pd.to_numeric(df_test[column], errors='coerce').isna()]
if not non_numeric_rows.empty:
for idx in non_numeric_rows.index:
print(f"Error: Non-numeric data found in column '{column}' for Employee ID
'{df_test.loc[idx, 'Employee_ID']}' in {file}")
error_flag = True
if (df_test[column] < 0).any():
print(f"Error: Negative values found in column '{column}' in {file}")
error_flag = True

if error_flag:
print("Skipping salary calculations due to data errors.\n")
continue

employee_id = df_test['Employee_ID'].values
basic_salary = df_test['Basic_Salary'].values
HRA = df_test['HRA'].values
other_allowances = df_test['Other_Allowances'].values
```

```
income_tax = df_test['Income_Tax'].values
employee_provident_fund = df_test['Employee_Provident_Fund'].values
professional_tax = df_test['Professional_Tax'].values
bonuses = df_test['Bonuses'].values
overtime = df_test['Overtime'].values
deductions = df_test['Deductions'].values
medical_allowance = df_test['Medical_Allowance'].values
travel_allowance = df_test['Travel_Allowance'].values
leave_encashment = df_test['Leave_Encashment'].values

gross_salary = basic_salary + HRA + other_allowances + bonuses + overtime +
medical_allowance + travel_allowance + leave_encashment
net_salary = gross_salary - (income_tax + employee_provident_fund + professional_tax +
deductions)

min_net_salary = net_salary[0]
max_net_salary = net_salary[0]
min_net_salary_employee_id = employee_id[0]
max_net_salary_employee_id = employee_id[0]

for i in range(1, len(net_salary)):
if net_salary[i] < min_net_salary:
min_net_salary = net_salary[i]
min_net_salary_employee_id = employee_id[i]
elif net_salary[i] > max_net_salary:
max_net_salary = net_salary[i]
max_net_salary_employee_id = employee_id[i]
print(f"Linear Scan -> Min: ₹{min_net_salary:.2f} (ID: {min_net_salary_employee_id}),
"
f"Max: ₹{max_net_salary:.2f} (ID: {max_net_salary_employee_id})")
def MinMaxSalary(net_salary, employee_id, left, right):
if left == right:
return (net_salary[left], employee_id[left]), (net_salary[left], employee_id[left])
if right == left + 1:
if net_salary[left] < net_salary[right]:
return (net_salary[left], employee_id[left]), (net_salary[right], employee_id[right])
else:
return (net_salary[right], employee_id[right]), (net_salary[left], employee_id[left])
mid = left + (right - left) // 2
(min1, min1_id), (max1, max1_id) = MinMaxSalary(net_salary, employee_id, left, mid)
(min2, min2_id), (max2, max2_id) = MinMaxSalary(net_salary, employee_id, mid + 1,
right)
```

```
MinSalary = (min1, min1_id) if min1 < min2 else (min2, min2_id)
MaxSalary = (max1, max1_id) if max1 > max2 else (max2, max2_id)
return MinSalary, MaxSalary
(min_net_salary, min_net_salary_employee_id), (max_net_salary,
max_net_salary_employee_id) = MinMaxSalary(net_salary, employee_id, 0, len(net_salary)
- 1)
print(f"Divide and conquer -> Min: ₹{min_net_salary:.2f} (ID:
{min_net_salary_employee_id}), "
f"Max: ₹{max_net_salary:.2f} (ID: {max_net_salary_employee_id})")
print('\n')
except Exception as e:
print(f" Error: {e}")

print('\nTest cases:\n')
run_test_cases()
```