LAB FAT SET-1

Q1. A parking garage charges a $2.00 minimum fee for stays up to three hours. For every hour or part thereof beyond three hours it charges an additional $0.50 per hour. The maximum charge for any single 24-hour period is $10.00. No car parks for longer than 72 hours.

Write a script (using JavaScript/JQuery) that:

1. Reads a CSV file named `yesterday.csv` where each row contains: `customer_id, entry_timestamp, exit_timestamp` (timestamps in ISO 8601).
2. For each record, computes the parking fee using a function `calculate_charges(duration_hours)` that:
   - o applies the base/minimum and per-hour rules,
   - o correctly handles fractional hours (partial hours count as a full hour),
   - o caps the fee at $10.00 per each 24-hour block (so a 30-hour stay is charged as `10.00 + charge_for_remaining_6_hours`, with no more than $30 for 72 hours),
   - o returns an error / flags the row if `exit_timestamp <= entry_timestamp`.
3. Produces an output report (CSV or table) listing: `customer_id, duration_hours (rounded up), charge, error_flag` and writes it to `charges_report.csv`.
4. Computes and prints summary statistics: total receipts, number of customers charged the daily maximum, average charge, and the `customer_id`(s) of the longest stay(s).

Q2. Write a PHP function named `lineSum` that takes a filename and a single line number and returns the sum of all integer values found on that line.

Requirements:

1. Signature: `function lineSum(string $filename, int $lineNumber): int` (or return `0` if no integers found).
2. The function should:
   - o open the file and read it line by line (do not load the whole file at once),
   - o ignore blank lines and lines beginning with `#`,
   - o split the target line on whitespace and sum tokens that are valid integers, ignoring non-integer tokens.
3. If the file does not exist or the requested line number is beyond the end of the file, return `0` (or optionally raise a simple error message).

Example file `sums.txt`:

```
# header
5
15 10
20 25
50
200
50 60
75 100
```

- `lineSum("sums.txt", 2)` ⟶ `25`
- `lineSum("sums.txt", 5)` ⟶ `200`