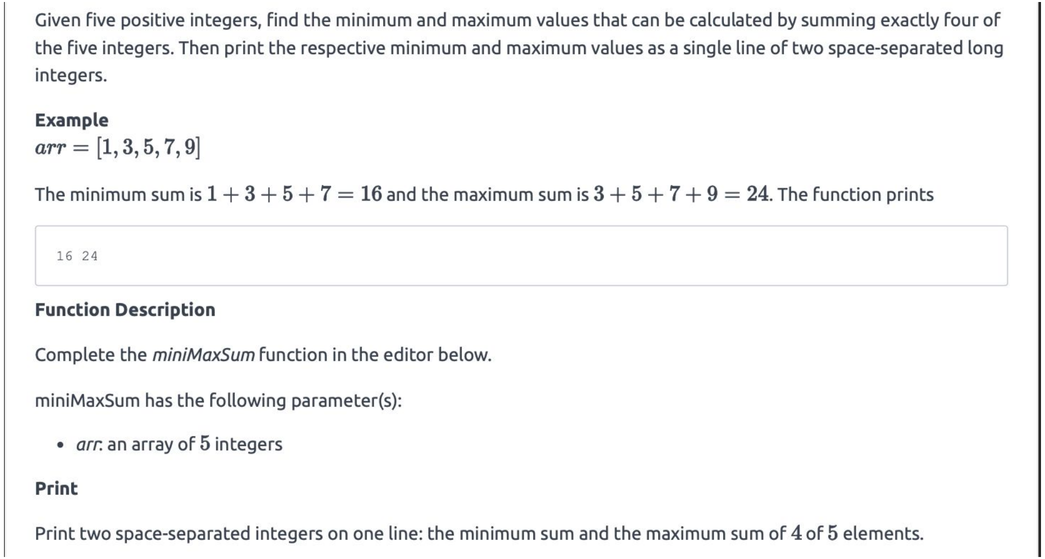
Waizly

Engineer Assessment

Keterangan

- **Penilaian akan didasarkan pada** :
  - Seberapa rapih struktur aplikasi
  - Seberapa rapih penulisan kode (Clean Code)
  - Seberapa tepat utilisasi teknologi, styling, dan plugin dalam menyelesaikan masalah
  - Penggunaan algoritma yang efisien
- **Soal implementasi** kerjakan dengan aturan sebagai berikut :
  1. Backend Role : Implementasi Backend 1 dan 2
  2. Frontend Role : Implementasi Frontend 1
  3. Mobile Role : Implementasi Mobile 1
  4. Fullstack Role : Implementasi Backend 1-2, Implementasi Frontend 1
  5. Techlead : Implementasi Backend 1-2, Implementasi Frontend 1, Implementasi Mobile 1
- **Mengerjakan soal implementasi diluar dari role yang dilamar merupakan nilai tambah**
- Semua jawaban berupa project Github yang bersifat public dan link di share sebagai jawaban ke tim HR

Problem Solving Basic - Test 1



Given five positive integers, find the minimum and maximum values that can be calculated by summing exactly four of the five integers. Then print the respective minimum and maximum values as a single line of two space-separated long integers.

Example

$arr = [1, 3, 5, 7, 9]$

The minimum sum is $1 + 3 + 5 + 7 = 16$ and the maximum sum is $3 + 5 + 7 + 9 = 24$. The function prints

```
16 24
```

Function Description

Complete the *miniMaxSum* function in the editor below.

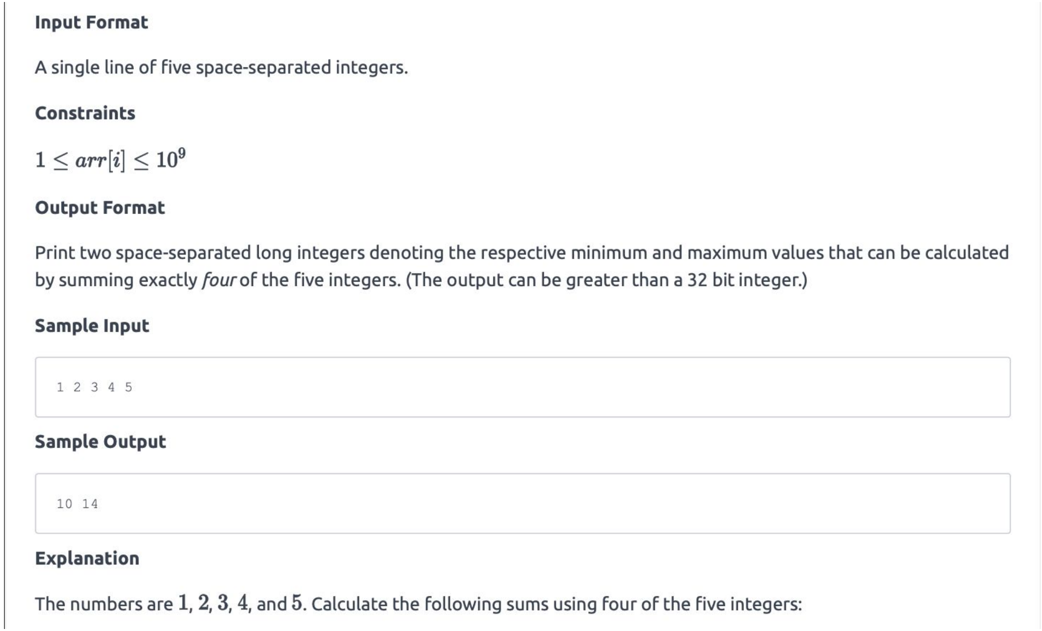
miniMaxSum has the following parameter(s):

- *arr*: an array of 5 integers

Print

Print two space-separated integers on one line: the minimum sum and the maximum sum of 4 of 5 elements.

Problem Solving Basic - Test 1



Input Format

A single line of five space-separated integers.

Constraints

$1 \leq arr[i] \leq 10^9$

Output Format

Print two space-separated long integers denoting the respective minimum and maximum values that can be calculated by summing exactly *four* of the five integers. (The output can be greater than a 32 bit integer.)

Sample Input

```
1 2 3 4 5
```

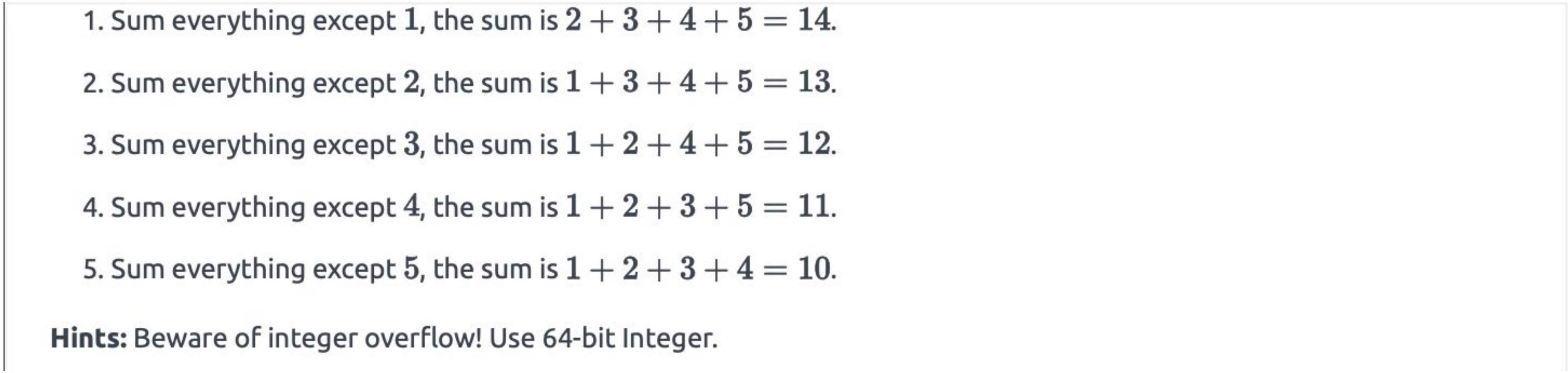
Sample Output

```
10 14
```

Explanation

The numbers are 1, 2, 3, 4, and 5. Calculate the following sums using four of the five integers:

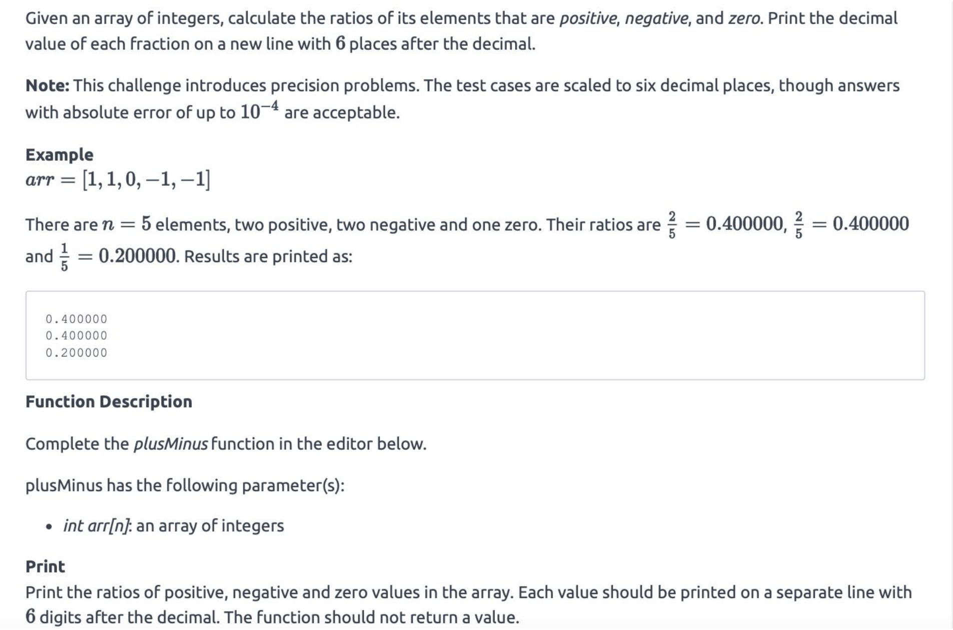
Problem Solving Basic - Test 1



1. Sum everything except 1, the sum is $2+3+4+5=14$.
2. Sum everything except 2, the sum is $1+3+4+5=13$.
3. Sum everything except 3, the sum is $1+2+4+5=12$.
4. Sum everything except 4, the sum is $1+2+3+5=11$.
5. Sum everything except 5, the sum is $1+2+3+4=10$.

Hints: Beware of integer overflow! Use 64-bit Integer.

Problem Solving Basic - Test 2



Given an array of integers, calculate the ratios of its elements that are *positive*, *negative*, and *zero*. Print the decimal value of each fraction on a new line with 6 places after the decimal.

Note: This challenge introduces precision problems. The test cases are scaled to six decimal places, though answers with absolute error of up to 10^{-4} are acceptable.

Example

$arr = [1, 1, 0, -1, -1]$

There are $n = 5$ elements, two positive, two negative and one zero. Their ratios are $\frac{2}{5} = 0.400000$, $\frac{2}{5} = 0.400000$ and $\frac{1}{5} = 0.200000$. Results are printed as:

```
0.400000
0.400000
0.200000
```

Function Description

Complete the *plusMinus* function in the editor below.

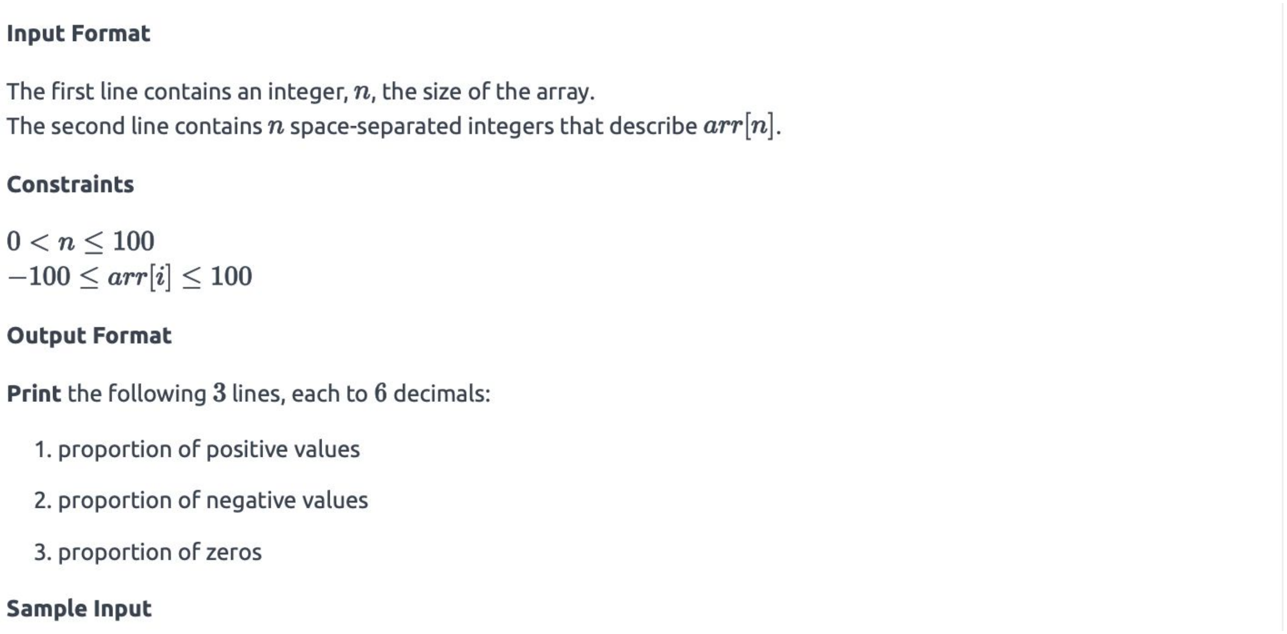
plusMinus has the following parameter(s):

- *int arr[n]*: an array of integers

Print

Print the ratios of positive, negative and zero values in the array. Each value should be printed on a separate line with 6 digits after the decimal. The function should not return a value.

Problem Solving Basic - Test 2



Input Format

The first line contains an integer, n, the size of the array.
The second line contains n space-separated integers that describe $arr[n]$.

Constraints

$0 < n \leq 100$
$-100 \leq arr[i] \leq 100$

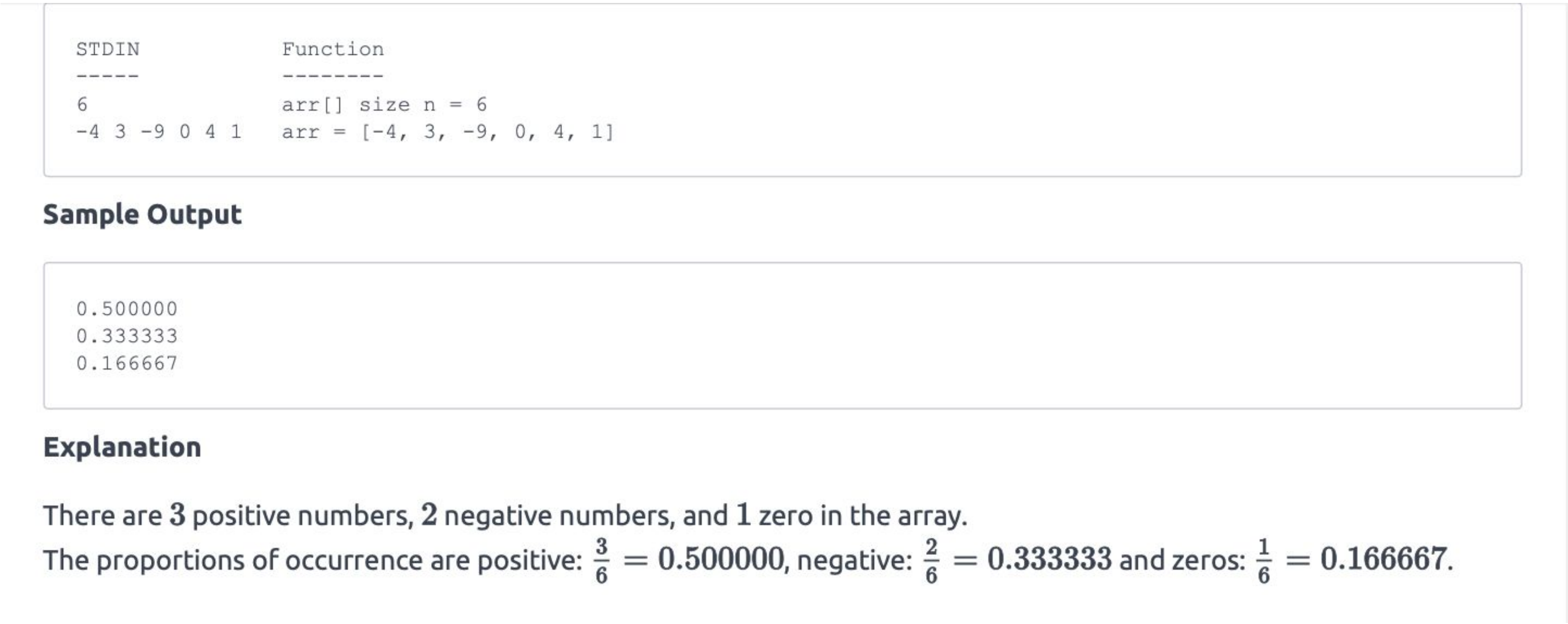
Output Format

Print the following 3 lines, each to 6 decimals:

1. proportion of positive values
2. proportion of negative values
3. proportion of zeros

Sample Input

Problem Solving Basic - Test 2



```
STDIN           Function
-----           --------
6               arr[] size n = 6
-4 3 -9 0 4 1   arr = [-4, 3, -9, 0, 4, 1]
```

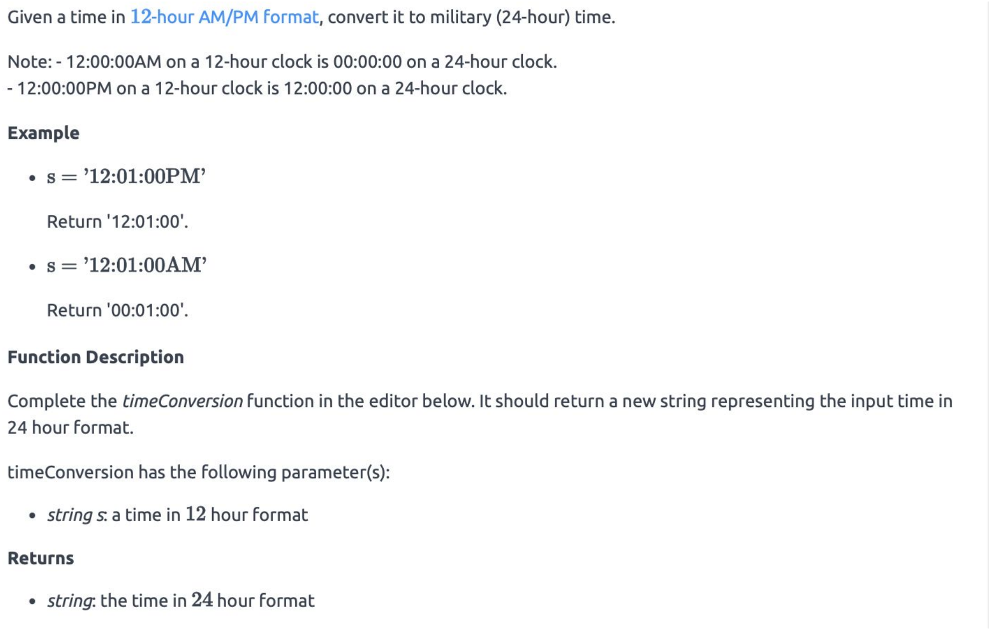
Sample Output

```
0.500000
0.333333
0.166667
```

Explanation

There are 3 positive numbers, 2 negative numbers, and 1 zero in the array.
The proportions of occurrence are positive: $\frac{3}{6} = 0.500000$, negative: $\frac{2}{6} = 0.333333$ and zeros: $\frac{1}{6} = 0.166667$.

Problem Solving Basic - Test 3



Given a time in 12-hour AM/PM format, convert it to military (24-hour) time.

Note: - 12:00:00AM on a 12-hour clock is 00:00:00 on a 24-hour clock.
- 12:00:00PM on a 12-hour clock is 12:00:00 on a 24-hour clock.

Example

- s = ’12:01:00PM’

  Return '12:01:00'.

- s = ’12:01:00AM’

  Return '00:01:00'.

Function Description

Complete the *timeConversion* function in the editor below. It should return a new string representing the input time in 24 hour format.

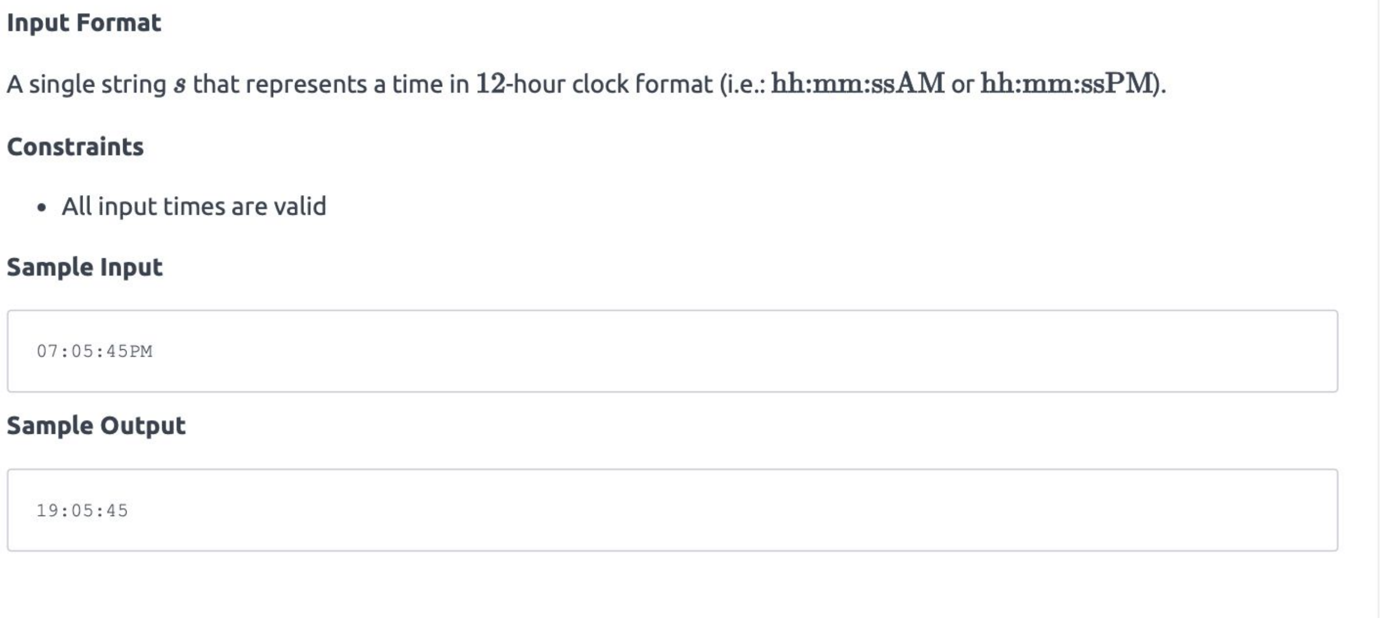
timeConversion has the following parameter(s):

- *string s*: a time in 12 hour format

Returns

- *string*: the time in 24 hour format

Problem Solving Basic - Test 3



Input Format

A single string s that represents a time in 12-hour clock format (i.e.: $\mathrm{hh:mm:ssAM}$ or $\mathrm{hh:mm:ssPM}$).

Constraints

- All input times are valid

Sample Input

```
07:05:45PM
```

Sample Output

```
19:05:45
```

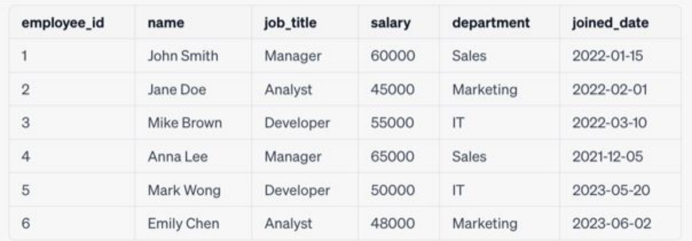
Implementation Test - Backend (1)

Create RESTFUL API with the programming language choose between PHP, Javascript, Java, or Golang (framework/library). You dont need to implement all of those Programming Language, just pick one you most like or fit. The API that you created **must at least have** CRUD features (**45 Points**). The API implementation can use any case you want, it will be a plus if you add features such as :

- Database Migration (**10 Points**)
- Authentication & Authorization (**20 Points)**
- Implement Logging (**10 Points**)
- Add Unit Testing (**15 Points**)
- Using Design Pattern (**Bonus 30 Points**)

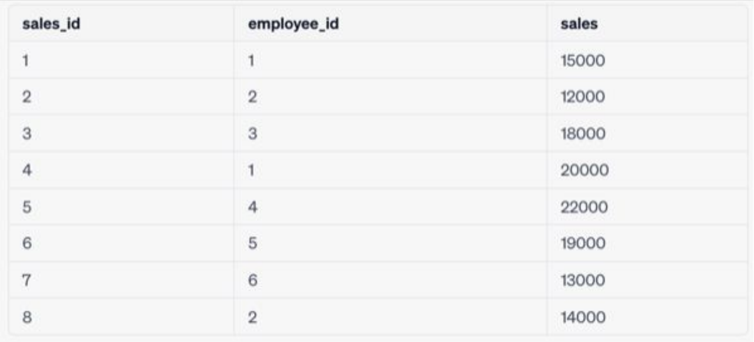
Note: Publish the application on Github for later the repository link that will be used as an answer

Implementation Test - Backend (2)



employee_id	name	job_title	salary	department	joined_date
1	John Smith	Manager	60000	Sales	2022-01-15
2	Jane Doe	Analyst	45000	Marketing	2022-02-01
3	Mike Brown	Developer	55000	IT	2022-03-10
4	Anna Lee	Manager	65000	Sales	2021-12-05
5	Mark Wong	Developer	50000	IT	2023-05-20
6	Emily Chen	Analyst	48000	Marketing	2023-06-02

Tabel **employees**



sales_id	employee_id	sales
1	1	15000
2	2	12000
3	3	18000
4	1	20000
5	4	22000
6	5	19000
7	6	13000
8	2	14000

Tabel **sales_data**

Note: Jawaban dalam bentuk .txt bisa disisipkan di repository git atau di upload terpisah, jawaban berupa query yang dibutuhkan untuk soal di halaman berikutnya

Implementation Test - Backend (2)

1. Tampilkan seluruh data dari tabel "employees" (**5 Points**)
2. Berapa banyak karyawan yang memiliki posisi pekerjaan (job title) "Manager"? (**5 Points**)
3. Tampilkan daftar nama dan gaji (salary) dari karyawan yang bekerja di departemen "Sales" atau "Marketing" (**10 Points**)
4. Hitung rata-rata gaji (salary) dari karyawan yang bergabung (joined) dalam 5 tahun terakhir (berdasarkan kolom "joined_date") (**10 Points**)
5. Tampilkan 5 karyawan dengan total penjualan (sales) tertinggi dari tabel "employees" dan "sales_data" (**10 Points**)
6. Tampilkan nama, gaji (salary), dan rata-rata gaji (salary) dari semua karyawan yang bekerja di departemen yang memiliki rata-rata gaji lebih tinggi dari gaji rata-rata di semua departemen (**15 Points**)
7. Tampilkan nama dan total penjualan (sales) dari setiap karyawan, bersama dengan peringkat (ranking) masing-masing karyawan berdasarkan total penjualan. Peringkat 1 adalah karyawan dengan total penjualan tertinggi (**25 Points**)
8. Buat sebuah **stored procedure** yang menerima nama departemen sebagai input, dan mengembalikan daftar karyawan dalam departemen tersebut bersama dengan total gaji (salary) yang mereka terima (**20 Points**)

Implementation Test - Frontend (1)

Create a simple ToDo App using Javascript (Framework/Library/Vanilla JS) pick just only one for the implementation. Just create the following feature :

- Minimum Feature (**45 Points**)
  - Add, Edit, Delete Task/Todo
  - Mark a task/todo is done/completed
- Will be plus if you add the following features :
  - Using animation (maybe when mark/delete/edit or insert data) (**10 Points**)
  - Using search bar for find specific task/todo (**10 Points**)
  - Get data from public API (its probably a time, weather or location or anything else) (**20 Points**)
  - Add Unit Testing (**15 Points**)
  - Using Design Pattern (**Bonus 30 Points**)

Note: Add a screenshot of the app and publish the application on Github for later the repository link that will be used as an answer

Implementation Test - Mobile (1)

Create a **todo application** with the relevant or mastered language or framework (Android/IOS/React Native/Flutter). Assessment will be based on several points, including :

1. Clarity and usability of the user interface (UI/UX) - 20 points
2. Application Functionality (Create, Read, Update, Delete) - 25 points
3. Quality of Code and Structure - 15 points
4. Security and Error Management - 10 points
5. Testing - 15 points
6. Using Design Patterns - 15 points

Note: Add a screenshot of the app and publish the application on Github for later the repository link that will be used as an answer