**Code Test**

**Sorting dan Alokasi**

Diberikan data set sebagai berikut:

|  |  |  |
| --- | --- | --- |
| Tagihan#1 | Due: 10 Jan 23 | 165,000 |
| Tagihan#2 | Due: 15 Feb 23 | 80,000 |
| Tagihan#3 | Due: 20 Jan 23 | 130,000 |
| Tagihan#4 | Due: 25 Mar 23 | 416,000 |
| Tagihan#5 | Due: 10 Feb 23 | 95,500 |
| Tagihan#6 | Due: 17 Aug 23 | 523,000 |

**Buatkan algoritma alokasi payment**. Input adalah nominal payment. Contoh Output adalah sebagai berikut:

1. Input Payment = 200,000

|  |  |  |  |
| --- | --- | --- | --- |
| Tagihan#1 | Due: 10 Jan 23 | 165,000 | 165,000 |
| Tagihan#3 | Due: 20 Jan 23 | 130,000 | 35,000 |
| Tagihan#5 | Due: 10 Feb 23 | 95,500 |  |
| Tagihan#2 | Due: 15 Feb 23 | 80,000 |  |
| Tagihan#4 | Due: 25 Mar 23 | 416,000 |  |

1. Input Payment = 500,000

|  |  |  |  |
| --- | --- | --- | --- |
| Tagihan#1 | Due: 10 Jan 23 | 165,000 | 165,000 |
| Tagihan#3 | Due: 20 Jan 23 | 130,000 | 130,000 |
| Tagihan#5 | Due: 10 Feb 23 | 95,500 | 95,500 |
| Tagihan#2 | Due: 15 Feb 23 | 80,000 | 80,000 |
| Tagihan#4 | Due: 25 Mar 23 | 416,000 | 29,500 |

**Berikan juga segala abnormal scenario** yang mungkin terjadi, isi dengan message box sesuai dengan scenario nya. Misal:

* input payment bernilai < 0
* apa lagi scenario yang mungkin terjadi?

**Buat Query DB** untuk mendapatkan Total Undue (belum jatuh tempo) dan Total Overdue (sudah jatuh tempo). Dengan kondisi tanggal hari ini adalah 25 Mar 23. Dalam 1 query saja.

**Algoritma Perhitungan Penalty**

Diberikan data set sebagai berikut:

**Data Table Tagihan**

|  |  |  |
| --- | --- | --- |
| **No Tagihan** | **Due Date** | **Total Tagihan** |
| Tagihan#1 | Due: 10 Jan 23 | 165,000 |
| Tagihan#3 | Due: 20 Jan 23 | 130,000 |
| Tagihan#5 | Due: 10 Feb 23 | 95,500 |
| Tagihan#2 | Due: 15 Feb 23 | 80,000 |
| Tagihan#4 | Due: 30 Mar 23 | 416,000 |
|  |  |  |

**Data Table Pembayaran**

|  |  |  |  |
| --- | --- | --- | --- |
| **No Payment** | **No Tagihan** | **Pmt Date** | **Pmt Amount** |
| Payment#1 | Tagihan#1 | 10 Jan 23 | 165,000 |
| Payment#2 | Tagihan#3 | 20 Feb 23 | 130,000 |
| Payment#2 | Tagihan#5 | 20 Feb 23 | 95,500 |
| Payment#3 | Tagihan#2 | 25 Feb 23 | 30,000 |
| Payment#4 | Tagihan#2 | 30 Mar 23 | 50,000 |
| Payment#4 | Tagihan#4 | 30 Mar 23 | 50,000 |

Dari dataset di atas, hitunglah Penalty “keterlambatan pembayaran”.

**Rumus Penalty:** A \* 2permill \* B

A = Total Tagihan yang terlambat bayar

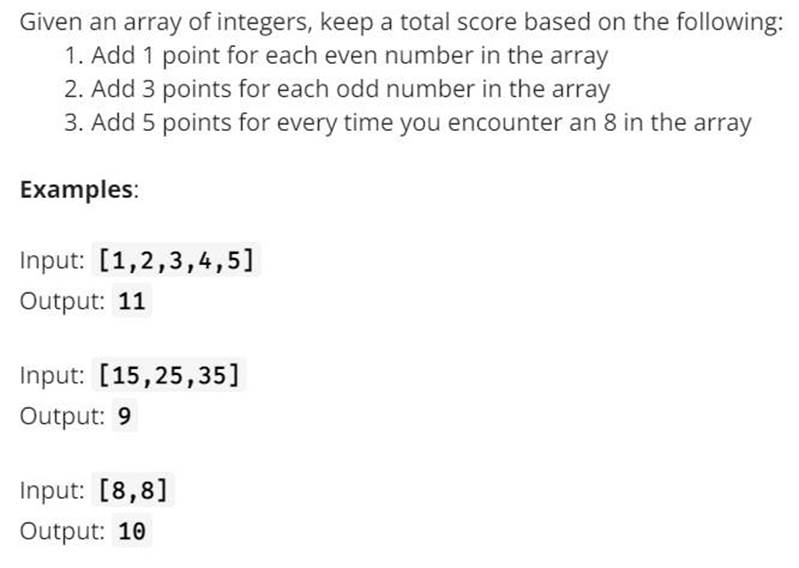
B = Jumlah hari keterlambatan sampai lunas

**Hasilkan dataset sebagai berikut: Hari ini = 29 Apr 22**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No Tagihan | No Penalty | Tagihan Overdue | Hari Keterlambatan | Amount Penalty |
| Tagihan#1 | 1 | 0 | 0 | 0 |
| Tagihan#3 | 1 | 130,000 | 31 | 800 |
| Tagihan#5 | 1 | 95,500 | 10 | 1,910 |
| Tagihan#2 | 1 | 30,000 | 10 | 600 |
| Tagihan#2 | 2 | 50,000 | 15 | 1,500 |
| Tagihan#4 | 1 | 366,000 | 30 | 21,960 |

**Algoritma #1**

Buatlah code untuk menghasilkan output sesuai dengan logic berikut:



**Algoritma #2**

Buatlah algoritma untuk empat soal di bawah ini, sesuai dengan nilai “n”

Contoh output di bawah adalah ketika nilai “n” = 5

**Database Query**

Buatlah dalam 1 query untuk mendapatkan duplicate data berdasarkan 1 field.

Buatlah dalam 1 query antara 2 table (misal TableA dan TableB), untuk menemukan data yang missing dari TableB, berdasarkan 1 field.