E-GENTING
PROGRAMMING COMPETITION 2013

General instructions:

1. Answer one or more of the questions.
2. The competition is an open book test.
3. The duration of the competition is 8 hours.
4. Do not discuss matters related to the questions with other contestants.
5. To receive credit for answering a question, your answer must be a credible response to the question. A credible response is an answer that solves the problem or would be likely to solve the problem with a little additional effort.
6. Provided your answer is a credible response, you will receive credit for the products of a methodical approach. For example, data flow diagrams and state transition diagrams and tables.
7. Your total score is the sum of the credit you receive from each credible response.
8. Your programs will be assessed on the ease with which they can be read and understood.
    - Indenting must be clean and consistent.
    - Variable names should describe the contents of the variables.
    - Coupling between modules should be visible.
    - Each module should do one thing well.
9. The questions are worth the following marks:

No	Name	Marks
1.	Rental Profile Report	200
2.	Select Card Data	120
3.	Effect of Double Buffering	500
4.	Conversion to HTML	60

10. Unless otherwise stated, your programs may be written in any mainstream programming language under any mainstream operating system.
11. Unless otherwise stated, you may make use of all the standard library functions of your chosen language and operating system.
12. The words ‘must’, ‘must not’, ‘required’, ‘should’, ‘should not’, and ‘may’ are to be interpreted as described in RFC 2119[1].
13. You are NOT expected to answer all questions.

---

[1] *Key words for use in RFCs to Indicate Requirement Levels*, RFC 2119, S. Bradner, March 1997.

1 RENTAL PROFILE REPORT

Kay's Rend-a-Car is a car rental business that operates several outlets around the city. Each outlet rents out a variety of cars. Kay's fleet management system creates a log file that records the departure and return of each rented vehicle from the outlet. Kay's would like to analyse the log file to produce a Rental Profile Report, which they can use to better allocate resources between the outlets.

Your task is to write a program to produce the Rental Profile Report.

The reporting program must accept the following parameters:

1. from-date (in DD-MM-YY format);
2. to-date (in DD-MM-YY format).

The Rental Profile Report must contain the following information:

1. reporting period (from-date and to-date);
2. for each outlet:
    a. outlet name;
    b. for each type of vehicle:
        i. vehicle name;
        ii. number of distinct customers who rented the type;
        iii. total vehicle-days rented;
        iv. total distance driven;
        v. rental received;
    c. totals for the outlet of:
        i. number of distinct customers who rented from the outlet;
        ii. total vehicle-days rented from the outlet;
        iii. total distance driven in vehicles rented from the outlet;
        iv. total rental received by the outlet;
3. grand totals for:
    a. number of distinct customers who rented cars from any outlet;
    b. total vehicle-days rented;
    c. total distance driven;
    d. rental received.

The Rental Profile Report should be laid out in accordance with the example in Figure 1.

Note that the number of distinct customers who rented from an outlet may not equal the sum of the number of distinct customers who rented each type of vehicle because some customers rent more than one type of vehicle. Similarly the number of distinct customers who rented cars from any outlet may not equal the sum of the numbers of distinct customers who rented from each outlet.

The log file is a text file containing tab-separated columns of the following information:

1. date of the transaction in YYYY-MM-DD format;
2. time of the transaction in HH:MM:SS format;
3. outlet name;
4. transaction type code ('OUT' for a vehicle being taken by a customer and 'IN' for a vehicle being returned by the customer);

5. vehicle registration number;
6. vehicle name;
7. vehicle's odometer reading;
8. customer's driver's licence number;
9. customer's name;
10. amount paid by or refunded to the customer.

A sample of the log file is presented in Figure 2.

If a rental period straddles a reporting period boundary, the reporting program must pro-rate the rental statistics between the parts of the rental period that are inside and outside the reporting period in proportion to the number of days that are inside and outside the reporting period respectively. For the purpose of pro-rating the rental statistics, the reporting program may disregard the times when the vehicle was rented out and returned. For example, if the rental period is 8 September to 19 September and the reporting period is 12 September to 1 October, 4 days of the rental period fall outside the reporting period and 8 days of the rental period fall inside the reporting period, so if the total rent received were RM2,400, then the pro-rated rent received in the reporting period would be 8/12 of 2,400 or RM1,600.

If a vehicle is rented and returned the same day, the reporting program should treat the vehicle as having been rented for a single day. If a vehicle is rented on one day and then returned on the next day, the reporting program should treat the vehicle as having been rented for two days.

RENTAL PROFILE REPORT
For DD-MM-YY to DD-MM-YY

Outlet	Vehicle	No of Customers	Days Rented	Distance Driven	Rental Received
KL Main	Proton Waja	27	80	4282	16000
	Proton Persona	9	50	2873	9000
	Toyota Camry	15	95	4146	52250
	Honda Accord	12	72	3817	39600
		52	297	15118	116850
KLIA	Proton Waja	48	165	8952	33000
	Proton Persona	21	98	5842	17640
	Toyota Camry	32	205	10801	112750
	Honda Accord	25	123	6102	67650
		108	591	31697	231040
Grand total		158	888	46815	347890

Figure 1 - Rental Profile Report Layout

```
2013-09-14<t>09:15:27<t>KL Main<t>OUT<t>KZA-1819<t>Honda Accord<t>16423<t>W428174<t>JOE BLOGS<t>1650
2013-09-12<t>09:30:04<t>KL Main<t>IN<t>FGA-9823<t>Proton Waja<t>24171<t>W118723<t>ALICE JONES<t>0
2013-09-12<t>11:48:42<t>KL Main<t>OUT<t>ZTA-5217<t>Proton Waja<t>1218<t>W842571<T>JOHN HILL<t>1800

<t> is the tab character ('\t' in C, C++ and Java)
```

Figure 2 - Example Log File

The log file may contain mismatched 'IN' records, which indicate that the vehicle was being rented when the log file was started, and may contain mismatched 'OUT' records, which indicate that the vehicle was still being rented at the time when the log file was sampled. The reporting program may disregard mismatched 'IN' and 'OUT' records.

2 SELECT CARD DATA

A Customer Relationship Management (CRM) System uses plastic cards to identify customers. At least potentially, a customer's identity may be recorded on any combination of up to four places on each card. The four places are:

1. magnetic stripe track 1;
2. magnetic stripe track 2;
3. integrated circuit with contacts;
4. radio frequency identification circuit.

Some users of the CRM System like to record the customer's identity in more than one place on the card and then verify that the multiple recordings reconcile with each other, whereas other users use one of the storage locations for the CRM recording and the others for other data, such as an encoding to open a hotel room door.

The CRM System users would like to be able to specify a system parameter that determines how the four sources of data are to be used to identify the customer.

The system engineers have in mind that it should be possible for the users to configure a parameter string that determines how the identification data is to be used. For example, if the parameter string were 'M1 | M2', the CRM System would use the data on magnetic stripe track 1, if it existed and was valid. If there was no data on magnetic stripe track 1, or if the data was invalid, then the CRM System would use the data on magnetic stripe track 2.

Users who want to reconcile multiple recordings could define 'M1 & M2', which would be similar to 'M1 | M2', except that in situations where there are valid recordings on both magnetic stripe tracks, the CRM System would check that the two tracks contained the same data.

It would even be possible to define expressions of the form 'IC | (M1 & M2)', which would mean that the CRM System should use the integrated circuit data if existed and was valid, otherwise it should reconcile the two sources of magnetic stripe data.

Your task is to write a function with a calling syntax similar to that of `SelectCardData` in Figure 3. Your function must accept the parameter string and the card data from the four recordings and return the selected recording or an error code.

In the `CardData_t` structure, the error code, `NO_ERROR`, indicates that the recording is valid. If the error code is anything other than `NO_ERROR`, the recording is invalid and the error code explains the nature of the error.

```
enum ErrorCode_t {
     NO_ERROR,
     PARITY_ERROR,
     LRC_ERROR,
     READER_IO_ERROR,
     DATA_MISMATCH_ERROR,
     PARAMETER_STRING_ERROR
};

struct CardData_t {
     ErrorCode_t    errorCode;
     string         recording;
};

CardData_t SelectCardData (
     const char *parameterString,
     const CardData_t *m1CardData,
     const CardData_t *m2CardData,
     const CardData_t *icCardData,
     const CardData_t *rfCardData);
```

Figure 3 – SelectCardData Function Declaration

The full syntax of the parameter string is given by the `data_selection` production in the BNF in Figure 4. The meanings of the data source codes are given in Table 1 and the truth tables of the operators are given in Table 2 and Table 3.

If the function indentifies an invalid parameter string, it should return `PARAMETER_STRING_ERROR`.

```
data_selection : data_source
               | data_selection '|' data_source
               | data_selection '&' data_source
               ;

data_source    : 'M1'
               | 'M2'
               | 'IC'
               | 'RF'
               | '(' data_selection ')'
               ;
```

Figure 4 - Parameter String Syntax

Table 1 – Data Source Codes

Code	Data Source
M1	Magnetic stripe track 1
M2	Magnetic stripe track 2
IC	Integrated circuit with contacts
RF	Radio frequency identification circuit

Table 2 – Truth Table for Operator '|'

First operand (a)	Second operand (b)	Value of a | b
Valid data	Valid data	Valid data of the first operand
Valid data	Invalid data	Valid data of the first operand
Invalid data	Valid data	Valid data of the second operand
Invalid data	Invalid data	Error code of the first operand

Table 3 – Truth Table for Operator '&'

First operand (a)	Second operand (b)	Value of a & b
Valid data	Valid data	If the data associated with the two operands is the same: Valid data of the first operand Else DATA_MISMATCH_ERROR
Valid data	Invalid data	Vaid data of the first operand
Invalid data	Valid data	Valid data of the second operand
Invalid data	Invalid data	Error code of the first operand

3 EFFECT OF DOUBLE-BUFFERING

A twin host transaction processing system has the architecture shown in Figure 5.

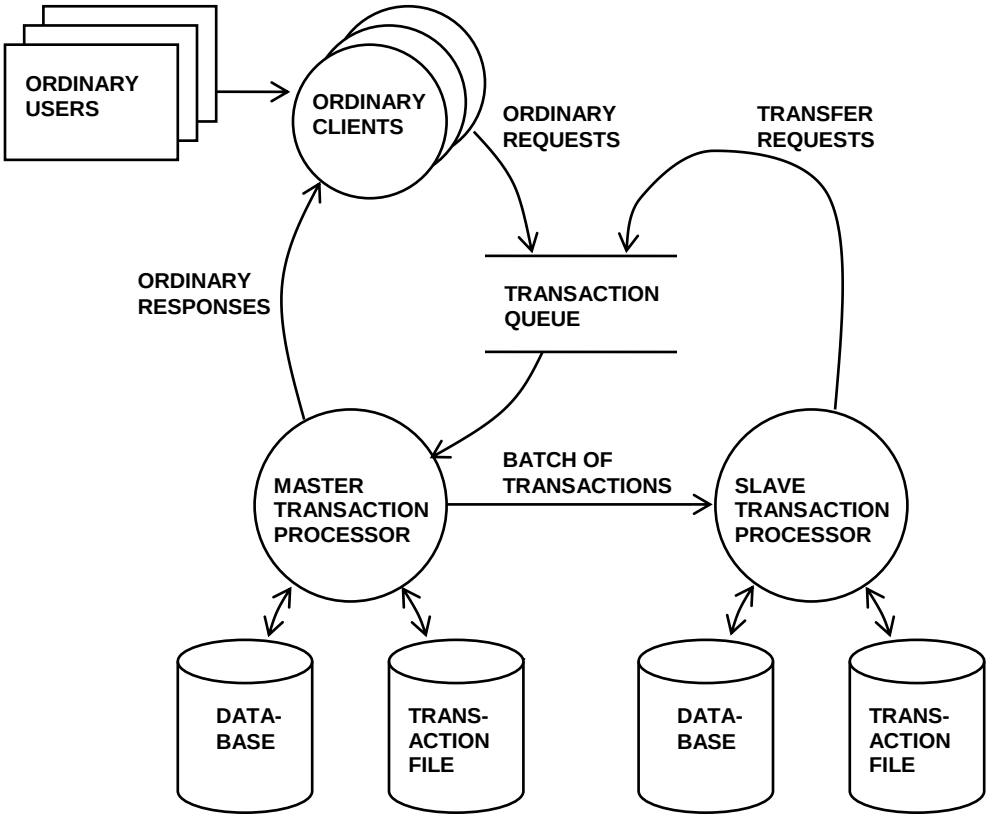


Figure 5 – Master/Slave System Architecture

Ordinary clients submit ordinary requests via a transaction queue. The Master Transaction Processor retrieves each request one-at-a-time from the transaction queue, processes the request, updating the database as necessary, and logging the request in the transaction file and returning a response back to the client.

A Slave Transaction Processor, which runs on a physically different computer, maintains a mimic database and transaction file by sending a transfer request to the Master Transaction Processor via the transaction queue. On receipt of the transfer request, the Master Transaction Processor retrieves a batch of transactions from the transaction file and sends the batch to the Slave Transaction Processor. The Slave Transaction Processor re-processes the transactions, updating its database to keep it synchronised with that of the Master Transaction Processor, and recording the transactions in its own transaction file. When the Slave Transaction Procesor finishes processing a batch of transactions, it sends another transfer request to the Master Transaction Processor to retrieve another batch of transactions and so on.

Users have observed that during times of high load the Slave Transaction Processor is unable to keep up with the Master Transaction Processor and the slave lag (the number of

transactions the Slave Transaction Processor lags behind the Master Transaction Processor) progressively increases.

The system engineers responsible for maintaining the transaction processing system disagree as to what might be the best way to increase the efficiency of the master-slave interface to avoid progressively increasing slave lags.

Some of the engineers think the best solution might be to double-buffer the interface between the Master and Slave Transaction Processors. With this architecture, the Slave Transaction Processor would send a second transfer request as soon as it receives a batch of transactions from the Master Transaction Processor. It would then process the transactions in the first batch while the Master Transaction Processor is servicing the second transfer request. When the Slave Transaction Processor finishes processing the first transfer batch, it would only need to wait a short time, if at all, for the response to the second transfer request. In effect this optimisation would allow the transfer of data from the Master Transaction Processor to occur in parallel with the processing of transaction data by the Slave Transaction Processor.

Other engineers are of the opinion that simply increasing the capacity of the transfer batch might have much the same effect by distributing the transfer overhead over a larger number of transferred transactions. Increasing the capacity of the transfer batch is very much easier and less costly than overhauling the architecture of the system to facilitate double buffering, so this alternative needs to be investigated.

Your task is to program a computer simulation of the original design and the two alternatives to determine the threshold load (in units of ordinary transactions per second) at which the slave lag starts to progressively increase.

The ordinary requests are divided into Class A, B and C requests as listed in Table 4.

Table 4 – Ordinary Request Mix

Class	Proportion of request mix (%)	Minimum processing time (ms)	Maximum processing time (ms)	Minimum transaction record length (bytes)	Maximum transaction record length (bytes)
A	75	0.5	1.5	50	150
B	20	10.0	20.0	200	500
C	5	100.0	250.0	1,000	24,000

75 percent of the requests are Class A requests, which, for the purposes of the simulation, have a processing time evenly distributed between 0.5 and 1.5 milliseconds and a record length in the transfer batch evenly distributed between 50 and 150 bytes and similarly for the other transaction classes as listed in the table. The Master and Slave Transaction Processors take roughly the same amount of time to process a transaction.

The simulation program may assume that the processing time of a transfer request is negligible and that transfer requests are not sent in transfer batches.

The current system has a transfer batch capacity of 32,768 bytes. The engineers who suggest that increasing the size of the transfer batch might alleviate the slave lag problem suggest an increased transfer batch capacity of 131,072 bytes.

Your simulation program must model the three alternatives: the current system, double buffering and the increased transfer batch capacity, and determine the threshold load (in ordinary transactions per seconds) at which the slave lag starts to progressively increase.

The simulation program must model 8 ordinary clients submitting ordinary requests pseudo-randomly selected from the transaction mix listed in Table 4. The ordinary clients must simulate the submission of the requests with an inter-request time evenly distributed between 4.0 / L and 12.0 / L seconds, where 'L' is the simulated load in transactions per second. The inter-request time is the time between submission of one ordinary request and the submission of the next ordinary request. If the Master Transaction Processor takes longer than the scheduled inter-request time to process a request, then the client should model the submission of the next request immediately upon receipt of the response to the previous request.

The simulation program should assume that the data transfer times are negligible.

The simulation program should determine whether the slave lag is progressively increasing by sampling the slave lag after 5, 10, 15 and then 20 minutes of simulated transaction processing. If the slave lag for the three samples after the first is progressively increasing (i.e. the slave lag at each sample time is greater than the slave lag at the preceding sample time) then the simulation program should assume that the slave lag is progressively increasing.

The simulation program should determine the threshold load at which the slave lag progressively increases by a process of trial-and-error in the range 10 to 10,000 transactions per second. The simulation program should first test (10 + 10,000) / 2 or 5,005 transactions per second. If the simulation finds that the slave lag is progressively increasing at 5,005 transactions per second, it should restrict the range to 5,005 to 10,000 transactions per second and repeat the process. 12 trials should be sufficient to achieve adequate precision.

4 CONVERSION TO HTML

A computer system generates document files that are usually printed on a strip printer attached to a cash register. The documents are typically receipts and credit notes, although there are other types.

The operator of the computer system would like to be able to display the document files on an ordinary computer terminal, but the document files contain special codes to select bold and italic fonts, that corrupt the output when it is displayed 'as-is'.

The operator would like a program that translates the document files into HTML so that the files can be viewed using an ordinary web browser. Your task is to write the translation program.

The translation program must read a document file from its standard input stream and write the document file in HTML format to its standard output stream.

The document files are plain text, except for the special codes listed in Table 5, and are designed to be printed on a strip printer in a fixed-pitch font.

Table 5 – Special Codes

Special Code	Meaning
\B	Print the succeeding text in a bold font of the same pitch as the normal font
\I	Print the succeeding text in an italic font of the same pitch as the normal font
\N	Return to the normal font after printing something in the bold or italic font
\\	Print a backslash character

The \B and \I codes are not cumulative in their effect and there is no bold italic font, only a bold font and an italic font. For example '`Normal \BBold \IItalic \NNormal`' is printed as '`Normal` **`Bold`** *`Italic`* `Normal`'.

Figure 6 is an example of a typical document file and Figure 7 is an example of that document file in HTML format.

```
\IGreen Valley
Department Store
\BRECEIPT\N
Location:    HOMEWARE
Cashier:     James
Date:        15-10-13
Time:        15:23:15
Document no: 17783225
Item:        LX7 Set
Amount:        782.50
Tendered:     1000.00
Balance:       217.50

\I782 points credited
to your bonus account\N
```

Figure 6 – Example Document File

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 3.0//EN">
<html>
<body>
<pre><i>Green Valley
Department Store
</i><b>RECEIPT</b>
Location:    HOMEWARE
Cashier:     James
Date:        15-10-13
Time:        15:23:15
Document no: 17783225
Item:        LX7 Set
Amount:        782.50
Tendered:     1000.00
Balance:       217.50

<i>782 points credited
to your bounus account</i>
</pre>
</body>
</html>
```

Figure 7 – Example HTML File

PERTANDINGAN PENGATURCARAAN E-GENTING 2013

Arahan-arahan am:

1. Jawab satu atau lebih soalan yang diberikan.
2. Pertandingan ini adalah satu ujian di mana peserta-peserta dibenarkan untuk merujuk kepada buku-buku atau bahan-bahan rujukan.
3. Masa yang diperuntukan untuk pertandingan ini adalah 8 jam.
4. Perbincangan sesama peserta tidak dibenarkan.
5. Untuk menerima kredit dalam menjawab sesuatu soalan, jawapan anda mestilah merupakan penyelesaian yang munasabah. Penyelesaian yang munasabah ialah jawapan yang menyelesaikan masalah tersebut atau jawapan yang mungkin menyelesaikan masalah tersebut dengan sedikit usaha tambahan.
6. Sekiranya jawapan anda merupakan penyelesaian yang munasabah, anda akan menerima kredit untuk hasilan methodikal seperti gambar rajah aliran data, gambar rajah peralihan keadaan, jadual dan sebagainya.
7. Jumlah markah anda adalah jumlah kredit yang anda perolehi bagi setiap penyelesaian yang munasabah.
8. Program-program anda akan dinilai berdasarkan kepada betapa mudah ianya boleh dibaca dan difahami.
    1. Takukan mestilah bersih dan sejajar.
    2. Nama-nama pembolehubah harus menggambarkan isi kandungan pembolehubah-pembolehubah berkenaan.
    3. Pergandingan di antara modul-modul mestilah nyata.
    4. Setiap modul harus melakukan satu perkara dengan baik.
9. Markah yang diperuntukan kepada setiap soalan adalah seperti berikut:

No	Tajuk	Markah
1.	Laporan Profil Penyewaan	200
2.	Pemilihan Data Kad	120
3.	Kesan Dwi-Penampan	500
4.	Pengubahan Ke HTML	60

10. Kecuali dinyatakan, program anda boleh ditulis dengan menggunakan mana-mana bahasa pengaturcaraan utama di bawah mana-mana sistem operasi utama.
11. Kecuali dinyatakan, anda boleh menggunakan semua fungsi piawaian perpustakaan dalam bahasa pengaturcaraan dan sistem operasi yang anda pilih.

12. Perkataan-perkataan ‘must’, ‘must not’, ‘required’, ‘should’, ‘should not’, dan ‘may’ adalah ditafsirkan seperti diterangkan dalam RFC 2119[2].

13. Anda TIDAK perlu menjawab semua soalan

[2] Key words for use in RFCs to Indicate Requirement Levels, RFC 2119, S. Bradner, March 1997.

1 LAPORAN PROFIL PENYEWAAN

Kay's Rent-a-Car merupakan satu perniagaan sewa kereta yang mengendalikan beberapa cawangan di sekitar bandar. Sistem pengurusan Kay menghasilkan satu fail log yang mencatat pelepasan dan pemulangan setiap kenderaan yang disewa dari cawangan-cawangannya. Kay ingin menganalisa fail log tersebut bagi menghasilkan satu Laporan Profil Penyewaan yang boleh digunakan untuk peruntukkan sumber di antara cawangan-cawangan secara lebih berkesan.

Tugas anda ialah menulis satu aturcara yang menghasilkan Laporan Profil Penyewaan tersebut.

Aturcara tersebut mesti menerima parameter-parameter berikut:

1. tarikh-dari (dalam format DD-MM-YY);
2. tarikh-hingga (dalam format DD-MM-YY).

Laporan Profil Penyewaan tersebut mesti mengandungi maklumat berikut:

- tempoh laporan (tarikh-dari dan tarikh-hingga);
- untuk setiap cawangan:
    - nama cawangan;
    - untuk setiap jenis kenderaan:
        - nama kenderaan;
        - bilangan pelanggan berbeza yang menyewa jenis tersebut;
        - jumlah kenderaan-hari disewa;
        - jumlah jarak dipandui;
        - bayaran sewa yang diterima;
    - jumlah bagi cawangan tersebut untuk:
        - bilangan pelanggan berbeza yang menyewa dari cawangan tersebut
        - jumlah kenderaan-hari disewa dari cawangan tersebut;
        - jumlah jarak dipandui untuk kenderaan yang disewa dari cawangan tersebut;
        - jumlah bayaran sewa yang diterima oleh cawangan tersebut;
- jumlah besar bagi:
    - bilangan pelanggan berbeza yang menyewa dari mana-mana cawangan;
    - jumlah kenderaan-hari disewa;
    - jumlah jarak dipandui;
    - bayaran sewa yang diterima;

Laporan Profil Penyewaan harus diatur seperti contoh dalam Gambarajah 1.

Perhatikan bahawa bilangan pelanggan berbeza yang menyewa dari sesebuah cawangan mungkin tidak bersamaan dengan jumlah bilangan pelanggan berbeza yang menyewa setiap jenis kenderaan sebab sesetengah pelanggan menyewa lebih dari satu jenis

kenderaan. Begitu juga dengan bilangan pelanggan berbeza yang menyewa kereta dari mana-mana cawangan mungkin tidak bersamaan dengan jumlah bilangan pelanggan berbeza yang menyewa dari setiap cawangan.

Fail log tersebut merupakan satu fail teks yang mengandungi tab sebagi pemisah ruangan bagi maklumat yang berikut:

1. tarikh transaksi dalam format YYYY-MM-DD;
2. masa transaksi dalam format HH:MM:SS;
3. nama cawangan;
4. kod jenis transaksi ('OUT' apabila sesebuah kenderaan diambil oleh pelanggan dan 'IN' apabila sesebuah kenderaan dipulangkan oleh pelanggan;
5. nombor pendaftaran kenderaan;
6. nama kenderaan;
7. bacaan odometer kenderaan;
8. nombor lesen memandu pelanggan;
9. nama pelanggan;
10. amaun dibayar oleh atau dibayar balik kepada pelanggan.

Salah satu contoh fail log tersebut dipaparkan dalam Gambarajah 2.

Jika tempoh penyewaan merentangi sempadan tempoh laporan, aturcara tersebut mesti mengira statistik penyewaan secara pro-rata di antara bahagian tempoh penyewaan yang berada di dalam dan di luar tempoh laporan dengan kadaran bilangan hari yang berada di dalam dan di luar tempoh laporan tersebut masing-masing. Untuk tujuan pengiraan statistik penyewaan secara pro-rata, aturcara tersebut boleh mengabaikan masa apabila kenderaan diambil keluar dan kenderaan dipulangkan. Sebagai contoh, jika tempoh penyewaan adalah dari 8 September sehingga 19 September dan tempoh laporan ialah 12 September sehingga 1 Oktober, 4 hari untuk tempoh penyewaan berada di luar tempoh laporan dan 8 hari untuk tempoh penyewaan berada di dalam tempoh laporan. Oleh yang demikian, jika jumlah bayaran sewa yang diterima adalah RM2,400, maka bayaran sewa pro-rata yang diterima dalam tempoh laporan tersebut ialah 8/12 untuk RM2,400 atau RM1,600.

Jika sesebuah kenderaan disewa dan dipulangkan dalam hari yang sama, aturcara tersebut harus menganggap kenderaan tersebut disewa untuk satu hari. Jika sesebuah kenderaan disewa pada hari pertama dan dipulangkan pada keesokan hari, aturcara tersebut harus menganggap kenderaan tersebut disewa untuk dua hari.

```
RENTAL PROFILE REPORT
For DD-MM-YY to DD-MM-YY

                                 No of    Days Distance   Rental
Outlet     Vehicle           Customers  Rented   Driven Received

KL Main    Proton Waja              27      80     4282    16000
           Proton Persona            9      50     2873     9000
           Toyota Camry             15      95     4146    52250
           Honda Accord             12      72     3817    39600
                                   ----   -----  ------- --------
                                    52     297    15118   116850

KLIA       Proton Waja              48     165     8952    33000
           Proton Persona           21      98     5842    17640
           Toyota Camry             32     205    10801   112750
           Honda Accord             25     123     6102    67650
                                   ----   -----  ------- --------
                                   108     591    31697   231040
                                   ----   -----  ------- --------
Grand total                        158     888    46815   347890
                                   ====   =====  ======= ========
```

Gambarajah 1 - Susunan Laporan Profil Penyewaan

```
2013-09-14<t>09:15:27<t>KL Main<t>OUT<t>KZA-1819<t>Honda Accord<t>16423<t>W428174<t>JOE BLOGS<t>1650
2013-09-12<t>09:30:04<t>KL Main<t>IN<t>FGA-9823<t>Proton Waja<t>24171<t>W118723<t>ALICE JONES<t>0
2013-09-12<t>11:48:42<t>KL Main<t>OUT<t>ZTA-5217<t>Proton Waja<t>1218<t>W842571<T>JOHN HILL<t>1800


<t> is the tab character ('\t' in C, C++ and Java)
```

Gambarajah 2 - Contoh Fail Log

Fail log tersebut mungkin mengandungi rekod-rekod 'IN' yang salah padan, di mana ia menunjukkan bahawa kenderaan tersebut telah disewa semasa fail log tersebut dimulakan. Fail log tersebut juga mungkin mengandungi rekod-rekod 'OUT' yang salah padan, di mana ia menunjukkan bahawa kenderaan tersebut masih dalam sewaan apabila fail log tersebut dicatat. Aturcara tersebut boleh mengabaikan rekod-rekod 'IN' dan 'OUT' yang salah padan.

2 PEMILIHAN DATA KAD

Satu Sistem Pengurusan Perhubungan Pelanggan (CRM) menggunakan kad-kad plastik untuk mengenalpasti pelanggan-pelanggan. Identiti pelanggan boleh direkodkan pada mana-mana kombinasi empat kawasan pada setiap kad. Empat kawasan tersebut ialah:

1. trek 1 pada jalur magnetik
2. trek 2 pada jalur magnetik;
3. litar bersepadu dengan sentuhan;
4. litar pengenalan frekuensi radio.

Sesetengah pengguna sistem CRM suka merekodkan identiti pelanggan pada lebih daripada satu kawasan dalam kad dan kemudiannya mengesahkan rekod-rekod tersebut adalah sepadan. Manakala pengguna yang lain menggunakan salah satu kawasan untuk merekodkan data CRM dan kawasan yang lain untuk data yang lain seperti pengekodan untuk membuka pintu bilik hotel.

Pengguna-pengguna sistem CRM tersebut ingin menyatakan satu parameter sistem yang menentukan bagaimana keempat-empat sumber data tersebut akan digunakan untuk mengenalpasti pelanggan.

Jurutera-jurutera sistem mencadangkan bahawa adalah berkemungkinan untuk pengguna-pengguna menetapkan satu rentetan parameter yang menentukan bagaimana data identiti tersebut akan digunakan. Sebagai contoh, jika rentetan parameter tersebut ialah 'M1 | M2', sistem CRM akan menggunakan data pada trek 1 jalur magnetik jika data tersebut wujud dan sah. Jika tiada data wujud pada trek 1 jalur magnetik atau data tersebut tidak sah, sistem CRM akan menggunakan data pada trek 2 jalur magnetik.

Pengguna-pengguna yang ingin persepadanan antara pelbagai rekod boleh menggunakan 'M1 & M2', di mana ia adalah serupa dengan 'M1 | M2', kecuali dalam situasi ini apabila wujudnya rekod yang sah pada kedua-dua trek jalur magnetik, sistem CRM tersebut akan mengesahkan bahawa kedua-dua trek tersebut mengandungi data yang sama.

Adalah berkemungkinan juga untuk menetapkan ungkapan dalam bentuk 'IC | (M1 & M2)', yang bermaksud bahawa sistem CRM tersebut harus menggunakan data pada litar bersepadu jika ia wujud dan sah. Jika tidak, ia harus melakukan persepadanan antara dua sumber data pada jalur magnetik.

Tugas anda ialah menulis satu fungsi dengan sintaks bersamaan dengan `SelectCardData` dalam Gambarajah 3. Fungsi anda mesti menerima rentetan parameter serta data kad daripada empat kawasan rekod dan memulangkan rekod yang dipilih atau kod kesilapan.

Dalam struktur `CardData_t`, kod kesilapan, `NO_ERROR`, menunjukkan bahawa rekod tersebut adalah sah. Jika kod kesilapan adalah berlainan dengan `NO_ERROR`, rekod tersebut adalah tidak sah dan kod kesilapan tersebut menerangkan sifat kesilapan tersebut.

```
enum ErrorCode_t {
     NO_ERROR,
     PARITY_ERROR,
     LRC_ERROR,
     READER_IO_ERROR,
     DATA_MISMATCH_ERROR,
     PARAMETER_STRING_ERROR
};

struct CardData_t {
     ErrorCode_t    errorCode;
     string         recording;
};

CardData_t SelectCardData (
     const char *parameterString,
     const CardData_t *m1CardData,
     const CardData_t *m2CardData,
     const CardData_t *icCardData,
     const CardData_t *rfCardData);
```

Gambarajah 3 – Deklarasi Fungsi SelectCardData

Sintaks penuh bagi rentetan parameter tersebut diberikan oleh `data_selection` dalam BNF di Gambarajah 4. Penerangan kod-kod sumber data boleh didapati di Jadual 1 and jadual kebenaran untuk operator boleh didapati di Jadual 2 dan Jadual 3.

Jike fungsi tersebut mengenalpasti sesuatu rentetan parameter yang tidak sah, ia harus mengembalikan `PARAMETER_STRING_ERROR`.

```
data_selection : data_source
               | data_selection '|' data_source
               | data_selection '&' data_source
               ;

data_source    : 'M1'
               | 'M2'
               | 'IC'
               | 'RF'
               | '(' data_selection ')'
               ;
```

Gambarajah 4 – Sintaks Rentetan Parameter

Jadual 1 – Kod-kod Sumber Data

Kod	Sumber Data
M1	Trek 1 jalur magnetik
M2	Trek 2 jalur magnetic
IC	Litar bersepadu dengan sentuhan
RF	Litar pengenalan frekuensi radio

Jadual 2 – Jadual Kebenaran bagi Operator '|'

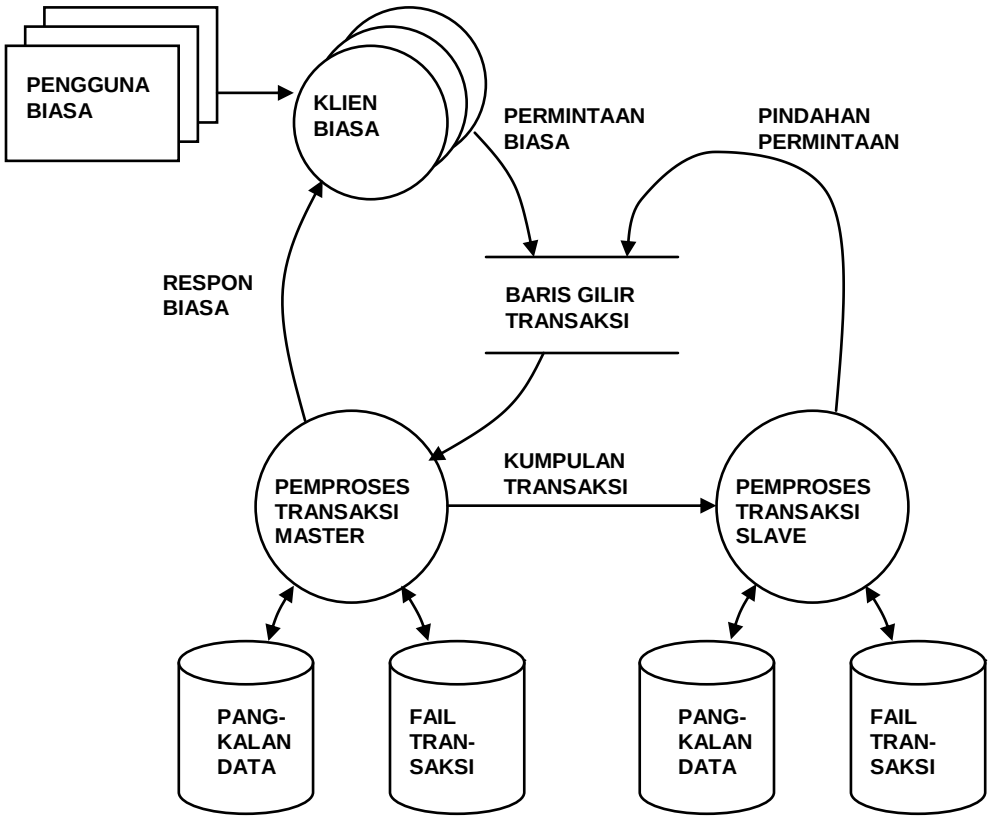
Operan pertama (a)	Operan kedua (b)	Nilai untuk a | b
Data yang sah	Data yang sah	Data yang sah bagi operan pertama
Data yang sah	Data yang tidak sah	Data yang sah bagi operan pertama
Data yang tidak sah	Data yang sah	Data yang sah bagi operan kedua
Data yang tidak sah	Data yang tidak sah	Kod kesilapan bagi operan pertama

Jadual 3 – Jadual Kebenaran bagi Operator '&'

Operan pertama (a)	Operan kedua (b)	Nilai bagi a & b
Data yang sah	Data yang sah	Jika data berkaitan dengan kedua-dua operan adalah sama: Data yang sah bagi operan pertama Jika tidak `DATA_MISMATCH_ERROR`
Data yang sah	Data yang tidak sah	Data yang sah bagi operan pertama
Data yang tidak sah	Data yang sah	Data yang sah bagi operan kedua
Data yang tidak sah	Data yang tidak sah	Kod kesilapan bagi operan pertama

3 KESAN DWI-PENAMPAN

Satu sistem pemprosesan transaksi berkembar mempunyai seni bina seperti dalam Gambarajah 5.



Gambarajah 5 – Seni Bina Sistem Master/Slave

Klien menghantar permintaan melalui satu giliran transaksi. Pemproses Transaksi Master tersebut mencapai setiap permintaan satu-per-satu dari giliran transaksi, memproses permintaan tersebut, mengemaskini pangkalan data jika perlu, mencatat permintaan tersebut dalam fail transaksi dan mengembalikan satu respon kepada klien.

Satu Pemproses Transaksi Slave yang beroperasi di fizikal komputer yang berbeza, mengekalkan satu pangkalan data dan fail transaksi yang serupa dengan menghantar permintaan pemindahan kepada Pemproses Transaksi Master tersebut melalui giliran transaksi. Apabila permintaan pemindahan tersebut diterima, Pemproses Transaksi Master mencapai satu kumpulan transaksi dari fail transaksi dan menghantarnya kepada Pemproses Transaksi Slave. Pemproses Transaksi Slave tersebut akan memproses transaksi-transaksi tersebut sekali lagi, mengemaskini pangkalan datanya supaya sepadan dengan pangkalan data di Pemproses Transaksi Master, dan kemudian mencatat transaksi tersebut pada fail transaksi yang tersendirinya. Apabila Pemproses Transaksi Slave selesai memproses satu kumpulan transaksi, ia menghantar satu lagi permintaan pemindahan kepada Pemproses Transaksi Master untuk mencapai satu lagi kumpulan transaksi dan seterusnya.

Pengguna mendapati semasa beban yang tinggi, Pemproses Transaksi Slave sukar dalam mengikuti Pemproses Transaksi Master dan ketinggalan Slave (bilangan transaksi di mana Pemproses Transaksi Slave ketinggalan dengan Pemproses Transaksi Master) bertambah secara beransur-ansur.

Jurutera-jurutera sistem yang menyelenggarakan sistem pemprosesan transaksi tersebut bercanggah dalam cara terbaik untuk meningkatkan kecekapan antara-muka master-slave supaya mengurangkan pertambahan ketinggalan Slave.

Sesetengah jurutera berpendapat bahawa cara terbaik ialah memperkenalkan dwi-penampan dalam antara-muka Pemproses Transaksi Master dan Slave. Dalam seni bina tersebut, Pemproses Transaksi Slave akan menghantar permintaan pemindahan kedua sebaik sahaja ia menerima satu kumpulan transaksi dari Pemproses Transaksi Master. Ia akan memproses transaksi-transaksi dalam kumpulan yang pertama dan dalam masa yang sama Pemproses Transaksi Master akan memproses permintaan pemindahan yang kedua. Apabila Pemproses Transaksi Slave habis memproses transaksi dalam kumpulan pertama, ia hanya perlu menunggu seketika sahaja untuk menerima respon bagi permintaan pemindahan yang kedua. Dalam erti kata lain, cara ini akan membenarkan pemindahan data dari Pemproses Transaksi Master berlaku secara serentak dengan pemprosesan data oleh Pemproses Transaksi Slave.

Jurutera yang lain berpendapat bahawa penambahan kapasiti kumpulan pemindahan akan menghasilkan kesan yang sama dengan menyebarkan beban pemindahan melalui bilangan transaksi yang lebih besar. Penambahan kapasiti kumpulan pemindahan adalah lebih mudah dan lebih murah daripada membaik-pulih seni bina sistem tersebut untuk memperkenalkan dwi-penampan. Oleh yang demikian, alternatif ini perlu dikaji.

Tugas anda ialah mengaturcara satu simulasi komputer bagi reka bentuk asal dan kedua-dua alternatif tersebut untuk menentukan beban ambang (dalam unit transaksi biasa per saat) di mana ketinggalan Slave mula bertambah secara beransur-ansur.

Permintaan-permintaan biasa dibahagikan kepada permintaan Kelas A, B dan C seperti dalam Jadual 4.

Jadual 4 – Taburan Permintaan Biasa

Kelas	Kadar taburan permintaan (%)	Masa pemprosesan minimum (ms)	Masa pemprosesan maksimum (ms)	Saiz rekod transaksi minimum (bytes)	Saiz rekod transaksi maksimum (bytes)
A	75	0.5	1.5	50	150
B	20	10.0	20.0	200	500
C	5	100.0	250.0	1,000	24,000

75 peratus daripada permintaan tersebut adalah permintaaan Kelas A, di mana untuk tujuan simulasi, mempunyai masa pemprosesan yang bertabur rata antara 0.5 sehingga 1.5 milisaat dan saiz rekod dalam kumpulan pemindahan bertabur rata antara 50 sehingga 150 bytes, dan sama juga dengan permintaan Kelas yang lain seperti diterangkan pada jadual di atas. Pemproses Transaksi Master dan Slave mengambil masa yang lebih kurang sama untuk memproses sesuatu permintaan.

Program simulasi tersebut boleh menganggap bahawa masa pemprosesan untuk permintaan pemindahan boleh diabaikan dan permintaan-permintaan pemindahan tidak dihantar dalam kumpulan pemindahan.

Sistem semasa mempunyai kapasiti pemindahan kumpulan yang bernilai 32,768 bytes. Jurutera-jurutera yang bercadang untuk menambah saiz kumpulan pemindahan untuk menyelesaikan masalah ketinggalan Slave mencadangkan supaya menambah kapasiti kumpulan pemindahan kepada 131,072 bytes.

Program simulasi anda mesti memodelkan tiga alternatif: sistem semasa, dwi-penampan dan penambahan kapasiti kumpulan pemindahan, serta menentukan beban ambang (dalam bilangan transaksi biasa per saat) di mana ketinggalan Slave mula bertambah secara beransur-ansur.

Program simulasi tersebut mesti memodelkan 8 klien biasa menghantar permintaan biasa yang dipilih secara pseudo-rawak daripada taburan transaksi seperti dalam Jadual 4. Klien-klien tersebut mesti menghantar permintaan-permintaan dengan jurang masa yang bertabur rata antara 4.0 / L sehingga 12.0 / L saat, di mana 'L' ialah beban simulasi dalam bilangan transaksi per saat. Jurang masa ialah masa di antara penghantaran satu permintaan biasa dengan penghantaran permintaan biasa yang seterusnya. Jika Pemproses Transaksi Master mengambil masa yang lebih lama daripada jurang masa yang dijadualkan, maka klien tersebut harus menghantar permintaan yang seterusnya sebaik sahaja menerima respon bagi permintaan yang sebelumnya.

Program simulasi tersebut harus menganggap bahawa masa penghantaran data boleh diabaikan.

Program simulasi tersebut harus menentukan sama ada ketinggalan Slave adalah bertambah secara beransur-ansur dengan melakukan persampelan ke atas Slave pada 5, 10, 15 dan 20 minit selepas simulasi pemprosesan transaksi bermula. Jika ketinggalan Slave untuk 3 sampel selepas yang pertama adalah bertambah secara beransur-ansur (iaitu ketinggalan Slave pada setiap masa sampel adalah lebih besar daripada ketinggalan Slave pada masa sampel yang sebelumnya) maka program simulasi tersebut harus menganggap bahawa ketinggalan Slave adalah bertambah secara beransur-ansur.

Program simulasi tersebut harus menentukan beban ambang di mana ketinggalan Slave bertambah secara beransur-ansur dengan satu proses cuba-cuba dalam lingkungan 10 sehingga 10,000 transaksi per saat. Program simulasi tersebut harus bermula dengan (10 + 10,000) / 2 atau 5,005 transaksi per saat. Jika simulasi tersebut mendapati bahawa ketinggalan Slave bertambah secara beransur-ansur pada 5,005 per saat, ia harus menghadkan lingkungan kepada 5,005 sehingga 10,000 transaksi per saat dan mengulangi proses tersebut. 12 cubaan harus mencukupi untuk mencapai ketepatan yang sesuai.

4 PENGUBAHAN KE HTML

Satu sistem komputer menjana fail-fail dokumen yang biasanya dicetak oleh pencetak jalur yang disambungkan kepada satu pendaftar tunai. Dokumen-dokumen tersebut kebiasaannya terdiri daripada resit, nota kredit dan juga jenis-jenis yang lain.

Operator sistem komputer tersebut ingin memaparkan fail-fail dokumen tersebut pada satu terminal komputer yang biasa, tetapi fail-fail dokumen tersebut mengandungi kod-

kod khas yang menentukan bentuk huruf 'bold' dan 'italic', yang akan menjejaskan paparan apabila ia dipaparkan secara langsung.

Operator tersebut memerlukan satu aturcara yang menterjemahkan fail-fail dokumen tersebut kepada HTML supaya fail-fail tersebut boleh dibaca melalui pelayar laman yang biasa. Tugas anda ialah mengaturcara program penterjemahan ini.

Program penterjemahan tersebut mesti membaca satu dokumen fail dari aliran input standard dan menulis fail dokumen tersebut dalam format HTML ke aliran output standard.

Fail-fail dokumen adalah dalam teks biasa, kecuali kod-kod khas yang disenaraikan dalam Jadual 5, dan adalah bertujuan untuk dicetak pada satu pencetak jalur dalam bentuk huruf dengan tetapan yang sama.

Jadual 5 – Kod-kod Khas

Kod Khas	Makna
`\B`	Cetak teks yang seterusnya dalam bentuk huruf 'bold' dengan tetapan yang sama seperti bentuk huruf biasa
`\I`	Cetak teks yang seterusnya dalam bentuk huruf 'italic' dengan tetapan yang sama seperti bentuk huruf biasa
`\N`	Kembali kepada bentuk huruf biasa selepas mencetak sesuatu dalam bentuk huruf 'bold' atau 'italic'
`\\`	Cetak satu aksara backslash

Kod-kod `\B` dan `\I` adalah tidak bertindih dalam kesannya. Bentuk huruf yang ‘bold’ dan ‘italic’ pada masa yang sama tidak wujud. Hanya bentuk huruf yang ‘bold’ sahaja atau ‘italic’ sahaja wujud. Contohnya, ‘`Normal \BBold \IItalic \NNormal`’ akan dicetak sebagai ‘`Normal` **`Bold`** *`Italic`* `Normal`’.

Gambarajah 6 ialah satu contoh fail dokumen dan Gambarajah 7 ialah satu contoh fail dokumen tersebut dalam format HTML.

```
\IGreen Valley
Department Store
\BRECEIPT\N
Location:    HOMEWARE
Cashier:     James
Date:        15-10-13
Time:        15:23:15
Document no: 17783225
Item:        LX7 Set
Amount:        782.50
Tendered:     1000.00
Balance:       217.50

\I782 points credited
to your bonus account\N
```

Gambarajah 6 – Contoh Fail Dokumen

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 3.0//EN">
<html>
<body>
<pre><i>Green Valley
Department Store
</i><b>RECEIPT</b>
Location:    HOMEWARE
Cashier:     James
Date:        15-10-13
Time:        15:23:15
Document no: 17783225
Item:        LX7 Set
Amount:        782.50
Tendered:     1000.00
Balance:       217.50

<i>782 points credited
to your bounus account</i>
</pre>
</body>
</html>
```

Gambarajah 7 – Contoh Fail HTML