E-GENTING
PROGRAMMING COMPETITION 2006

General instructions:

1. Answer one or more of the questions.
2. The competition is an open book test.
3. The duration for 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 will be the sum of the credit you received 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 marks allocated for the questions are as follows:

No	Name	Marks
1.	Microcontroller Emulator	200
2.	Cashier Activity Report	250
3.	Team Selection	300
4.	Loan Evaluator	100

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 MICROCONTROLLER EMULATOR

The Giveaway Toy Company manufactures small plastic toys in enormous numbers for fast food outlets to include in children's hamburger meals.

Giveaway is investigating the idea of using ultra low-cost microcontrollers to activate sounds, flash lights and turn motors on and off. Your task is to program an emulator for the proposed microcontroller. The emulator will be used to develop and test toy control programs.

The microcontroller has an 8-bit accumulator 'A', a stack pointer 'SP', a carry bit 'C' and a program counter 'PC'. It has an 8-bit data address space and a separate 16-bit program address space. It communicates to its peripheral devices via 8 input pins and 8 output pins.

The 8-bit data address space is structured as follows:

Address	Function
0x00 to 0xf0	240 bytes of general purpose random access memory (RAM)
0xf0 to 0xf7	8 input/output registers.
0xf8 to 0xff	Unused

The 8 input/output registers are used to read the state of the 8 input pins and set the state of the 8 output pins.

If the accumulator is loaded from any of the 8 input/output registers, the least significant bit of the accumulator is set to the state of the corresponding input pin. The other bits of the accumulator are reset.

If the accumulator is stored to any of the 8 input/output registers, the corresponding output pin is set to the state of the least significant bit of the accumulator. The other bits of the accumulator have no effect and may contain arbitrary data.

The 16-bit program address space maps to read-only memory that is pre-loaded with the executable program when the microcontroller is manufactured.

The microcontroller has the following instruction set:

Instruction	Program memory contents	Processing (C operators, all registers and data unsigned)	Description
LD A,#val	0x10 val	A = val; PC += 2;	Load accumulator, immediate

LD A,adr	0x11 adr	A = data[adr]; PC += 2;	Load accumulator, direct
LD SP,#val	0x12 val	SP = val; PC += 2;	Load stack pointer, immediate
ST adr,A	0x13 adr	data[adr] = A; PC += 2;	Store accumulator, direct
ADD A,adr	0x20 adr	C = A + data[adr] >= 256; A = A + data[adr] & 255; PC += 2;	Add, direct
SUB A,adr	0x21 adr	C = A – data[adr] < 0; A = A – data[adr] & 255; PC += 2;	Subtract, direct

BRA ofs	0x30 ofs	if (ofs < 128)
		  PC += 2 + ofs;
		else
		  PC += ofs – 254;
			Branch

BCS ofs	0x31 ofs	if (C) {
		  if (ofs < 128)
		  PC += 2 + ofs;
		  else
		  PC += ofs – 254;
		} else {
		  PC += 2;
		}
			Branch if carry set

JMP hi-lo	0x40 hi lo	PC = hi << 8 | lo;	Jump, direct

CALL hi-lo	0x41 hi lo	data[--SP] = PC+3 & 255;
		data[--SP] = PC+3 >> 8;
		PC = hi << 8 | lo;
			Call subroutine

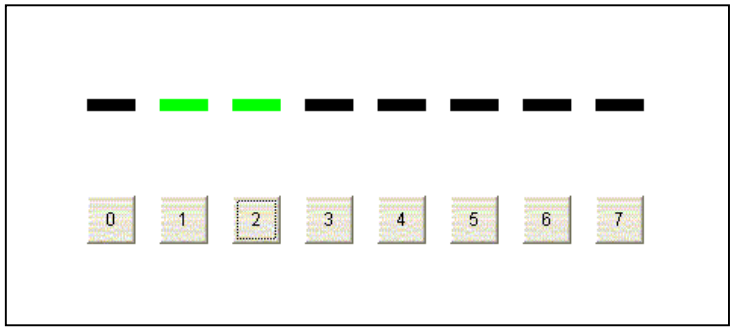
RET	0x42	PC = data[SP]<<8 | data[SP+1]; SP += 2;	Return from subroutine

In the preceding table, ‘data[x]’ is a reference to the contents of location ‘x’ in the 8-bit data address space.

When the microcontroller is initially powered up, it starts executing instructions starting at address zero in the 16-bit program address space.

The emulator must load the program to be executed from a binary disk file. The file may be smaller than the 64Kb capacity of the 16-bit program address space, in which case the contents of the file must be loaded starting at address zero.

The emulator must have a control panel with 8 indicator lights and 8 buttons similar to the following figure.



The 8 indicator lights must show the emulated state of the 8 output pins. When the program stores 1 to an input/output register, the corresponding indicator must turn green, and when the program stores 0 to an input/output register, the corresponding indicator must turn black.

The 8 buttons must control the emulated state of the 8 input pins. When a button is pressed, the program running in the emulator must read the value 1 from the corresponding input/output register, and when the button is released, the program must read the value 0 from the corresponding input/output register.

2 CASHIER ACTIVITY REPORT

Angina Marketing operates several chains of fast food outlets. Their brands include Fried Fowl, Ground Bull and Chease and Pastry, which market fried chicken, hamburgers and pizzas respectively. A significant number, but not all of Angina's outlets are located in airports and bus stations and are open 24 hours.

Angina maintains a database of information associated with each cashier shift at each cashier location at each of its restaurants. The information is stored in a database table with the following schema.

```
create table transactions (
     transNo             integer not null,
     tradingDate         char(10),
     locCode             char(6) not null,
     shiftNo             smallint not null,
     transType           smallint not null,
     transVal            integer not null
);
create index transNoInd on transactions(transNo);
```

`transNo`
: is a transaction number. Each new transaction is given the next transaction number in sequence.

`tradingDate`
: is the trading day date of the cashier shift. Each cashier shift is given a nominal trading day date. The trading day date is in ISO 8601 format (i.e. YYYY-MM-DD). The trading day date is not necessarily the same as the

real time date. For example, a shift that starts at 10:00pm and finishes at 6:00am the next day will typically have the trading day date of the day when the shift starts, with the trading day rolling over at 6:00am the next day.

`locCode`
: is a cashier location code. Each cashier location has a distinct location code.

`shiftNo`
: is the shift number of the cashier shift. For example, if a cashier location had three 8-hour shifts, the first shift might be shift 1, the second, shift 2 and the third shift 3. Shift numbers may be skipped. For example, a location may have data for shifts 1 and 3, but not 2.

`transType`
: is a transaction type code. The transaction type codes are listed in the table below.

`transVal`
: is the transaction value in cents. For example, a transaction with a value of $12.34 would be stored as the integer 1234.

The transaction type codes are:

Value	Name	Meaning
1	Opening balance	Money passed forward from the previous cashier shift.
2.	Fill	Money transferred to the cashier location to top up the float in the event that the cashier location does not have enough of a particular denomination of banknote.
3	Credit	Money transferred from the cashier location to the safe.
4	Sale	Money received from patrons in exchange for purchases.
5	Closing balance	Money passed forward to the next cashier shift.

Angina now requires a report to summarise the information in the transactions table.

The reporting program must accept a reporting period expressed as a from-date and to-date and generate a report with the following contents:

1. date and time the report was printed;
2. reporting period (from-date and to-date);
3. for each cashier shift in the reporting period:
    a. cashier location code,
    b. trading day date,
    c. shift number,
    d. opening balance,
    e. total fills in the shift,
    f. total credits in the shift,
    g. total sales in the shift,

    h. closing balance;
4. totals for all shifts of:
    a. opening balance,
    b. fills,
    c. credits,
    d. sales,
    e. closing balance.

The totals for all shifts of the opening and closing balances are not a simple sum of the opening and closing balance columns. The total opening balance is the sum of the opening balances of the first shifts of each cashier location in the reporting period and the total closing balance is the sum of the closing balances of the last shifts of each cashier location in the reporting period. For example:

Location	**Date**	**Shift**	**Opening balance**	**Fills**	**Credits**	**Sales**	**Closing balance**
FF-1	24-09-06	1	1,200.00	100.00	2,500.00	2,700.00	1,500.00
FF-1	24-09-06	2	1,500.00	0.00	2,800.00	2,500.00	1,200.00
FF-2	24-09-06	1	800.00	200.00	1,700.00	1,950.00	1,250.00
FF-2	24-09-06	2	1,250.00	0.00	2,200.00	1,850.00	900.00
TOTAL			2,000.00 (1,200+800)	300.00	9,200.00	9,000.00	2,100.00 (1,200+900)

The report must be formatted in accordance with the following layout:

```
         1         2         3         4         5         6         7         8
1...5....0....5....0....5....0....5....0....5....0....5....0....5....0....5....0
DD-MM-YY HH:MM             CASHIER ACTIVITY REPORT                      PAGE X-X
                          for XX-XX-XX to XX-XX-XX

Cashier                   Opening                                     Closing
Location Date      Shift  balance       Fills    Credits      Sales   balance

X------X XX-XX-XX   X   XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX
X------X XX-XX-XX   X   XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX
X------X XX-XX-XX   X   XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX
X------X XX-XX-XX   X   XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX
X------X XX-XX-XX   X   XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX
X------X XX-XX-XX   X   XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX
                        ---------- ---------- ---------- ---------- ----------
TOTAL                   XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX
                        ========== ========== ========== ========== ==========
```

Although trading day dates are stored in YYYY-MM-DD format on the database, all user-level date input and output must be in DD-MM-YY format.

The report lines must be sorted by cashier location code, trading day date and shift number.

If a shift in the reporting period is still open (i.e. there is no closing balance transaction on the database) the reporting program must calculate the current balance at the location and display it in the closing balance column. For this purpose:

Closing balance = opening balance + fills – credits + sales

Angina intends to use the report to reconcile the balances at cashier locations at the end of each trading day. As such, the reporting program should generate the report for a single day in less than 10 seconds.

Because the transactions table contains one row for each sale of a chicken dinner, hamburger or pizza, it is very large. The last time the software engineers checked, it contained more than 200 million rows, representing a little over 5 year's data. Any attempt to search on a key other than the transaction number results in a serial scan of the table, which takes around 2.5 hours. Nevertheless, a search for the data associated with a particular transaction number is quite quick, taking around 0.5ms. In the past, software engineers at Angina have experimented with creating indexes on other keys, but the impact on foreground transaction processing was unacceptable.

The software engineers at Angina have discovered that the transaction number can be used as a 'fuzzy index' for trading day date. Although cashier locations move from one trading day to the next at different times, at any point in time when a cashier is trading in day 'd', all the other cashier locations are trading in either the day before d, the day d, or the day after d. They suggest that this phenomenon be used to reduce the scope of the search.

3 TEAM SELECTION

Nautical Defence Systems (NDS) is a large government-linked company working on top-secret naval projects. The government has instructed NDS to initiate a project to develop a fleet of state-of-the-art patrol boats.

The patrol boats are to be equipped with a number of highly sophisticated computer systems, the exact nature of which is classified, but it is known that the systems include tactical threat identification and response systems, weapons control systems and ship management systems.

NDS does not have the resources to develop the systems themselves and needs to hire outside teams to do the software development. Your company is one of the bidders.

NDS has defined a scheme for rating the teams bidding for the contracts. Your task is to write a program that selects the team members from among the staff in your company so that the rating of the overall team is maximised.

The rating of the team is given by the following expression:

$$Rt = \frac{\sum_{i=1}^{n} Rp_i}{Y^{(n-1)}}$$

Where:

Rt	is the rating of the team.
n	is the number of team members.

Rp_i is the rating of the i-th programmer.
Y is the constant 1.1

The rating of an individual programmer is the simple sum of the value of the programmer's five best projects over the past five years. Project values are real numbers between 0 and 10. The method for valuing projects has already been completed and is out of the scope of the current task.

If two or more programmers undertook a project jointly, the project may only be included in the rating of one of the programmers. The ratings of the other programmers must be derived from alternative projects.

If a programmer undertook less than five projects in the past five years, or if, after joint projects included in others' ratings are excluded, a programmer is left with less than five projects, the sum of the value of that smaller number of projects is the programmer's rating.

The project values are stored on an SQL database with the following schema:

```
create table programmers (
     progNum        integer not null,
     progName       char(20) not null
);
create table projects (
     projNum        integer not null,
     projName       char(20) not null,
     projValue      double precision not null
);
create table assignments (
     asnProgNum     integer not null,
     asnProjNum     integer not null
);
```

`programmers`
is a table that contains one row for each programmer.

`progNum`
is a unique programmer number that identifies a programmer.

`progName`
is the programmer's name.

`projects`
is a table that contains one row for each project.

`projNum`
is a unique project number that identifies a project.

`projVal`
is the value of the project in the range 0 to 10.

`assignments`
is a table that relates a programmer to the projects that he undertook in the past 5 years.

`asnProgNum`
is a programmer number.

`asnProjNum`
: is a project number of a project that the programmer undertook in the past 5 years.

The database contains 75 programmers, 324 projects and 339 assignments.

Your program must select the highest rated team from among all the alternatives on the database. There is no restriction on the number of programmers that can be in a team, so long as the combination of team members in the team produces the highest rating. In addition to that, the program must be designed in a manner that will allow it to complete within one hour on a fast PC. It must display the following information about the team:

1. date and time the program was run;
2. for each team member:
    a. programmer number;
    b. programmer name;
    c. for each of the programmer's projects that was included in the programmer's individual rating:
        i. project number,
        ii. project name,
        iii. project value;
    d. programmer's individual rating;
3. number of team members;
4. overall rating of the team.

The information should be formatted in accordance with the following layout:

```
          1         2         3         4         5         6         7         8
1...5....0....5....0....5....0....5....0....5....0....5....0....5....0....5....0
DD-MM-YY HH:MM               TEAM SELECTION REPORT                      PAGE X-X

Programmer                    Project
Number  Name                  Number  Name                  Value

X----X  X------------------X  X----X  X------------------X  XX.X
                              X----X  X------------------X  XX.X
                                                            ----
                              Programmer total              XX.X

X----X  X------------------X  X----X  X------------------X  XX.X
                              X----X  X------------------X  XX.X
                                                            ----
                              Programmer total              XX.X
                                                           -----
Team total (X members)                                     XXX.X
```

4 LOAN EVALUATOR

A user needs to compare the cost of various financing arrangements for purchasing an item of capital equipment, but the proposals being made by the banks do not permit a convenient comparison of the various alternatives. Each bank has its own scheme of fees and repayments, which makes it difficult to compare the costs associated with one loan with the costs associated with the alternatives.

The user would like you to develop a program that receives the following parameters on a data entry screen:

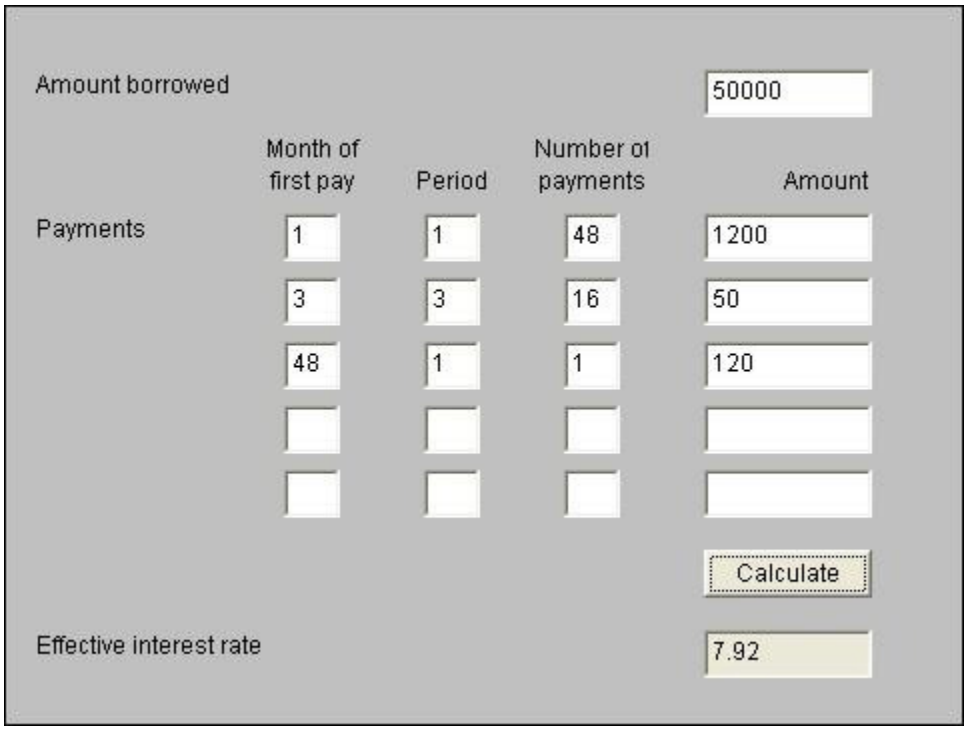
1. amount borrowed (the 'principal');
2. for each payment type (up to a maximum of 5 types):
    a. month number of the first payment,
    b. period between consecutive payments in months,
    c. number of payments,
    d. amount of each payment;

Payment types are establishment fees, account maintenance fees, repayments, account closure fees and so forth.

Month numbers start at zero, for payments made at the same time as the principal is borrowed, one for payments to be made one month after the principal is borrowed and so on.

The data entry screen must include a 'Calculate' button. When the Calculate button is clicked, the program must calculate and display the effective interest rate of the loan.

The data entry screen should be laid out as follows:



The effective interest rate of the loan is the annual interest rate 'R' that satisfies the equation:

$$P = \sum_{i=1}^{n} \frac{v_i}{\left(1 + \frac{R}{1200}\right)^{t_i}}$$

Where:

P	is the amount borrowed (the principal);
n	is the number of individual payments;
v_i	is the amount of an individual payment;
t_i	is the month number (i.e. time) at which the payment is to be made;
R	is the effective, monthly compound interest rate as a percentage.

The program should calculate and display the effective interest rate to two decimal places.

The program must reject invalid parameters (e.g. non-numeric values). If a parameter is invalid, the program should highlight the invalid parameter by displaying it on a red background and displaying 'ERROR' in the effective interest rate field.

The program may reject calculate requests that would result in interest rates outside the range 0 to 100%. If the program rejects parameters that result in out-of-range interest rates, it should display '****' in the effective interest rate field.

PERTANDINGAN PENGATURCARAAN E-GENTING 2006

Arahan-arahan am:

1. Jawab satu atau lebih soalan yang diberikan.
2. Pertandingan ini adalah satu ujian di mana peserta-peserta dibenarkan merujuk kepada buku-buku atau bahan-bahan rujukan.
3. Masa yang diperuntukan untuk pertandingan ini adalah 8 jam.
4. Perbincangan sesama peserta adalah tidak dibenarkan.
5. Untuk menerima kredit bagi menjawab sesuatu soalan, jawapan anda mestilah merupakan penyelesaian yang munasabah. Penyelesaian yang munasabah ialah jawapan yang menyelesaikan masalah tersebut atau jawapan yang mungkin akan menyelesaikan masalah tersebut dengan sedikit usaha tambahan.
6. Sekiranya jawapan anda merupakan penyelesaian yang munasabah, anda akan menerima kredit untuk hasilan methodical 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.
    - Takukan mestilah bersih dan sejajar.
    - Nama-nama pembolehubah harus menggambarkan isi kandungan pembolehubah-pembolehubah berkenaan.
    - Pasangan (coupling) di antara modul-modul mestilah nyata.
    - Setiap modul harus melakukan satu perkara dengan baik.
9. Markah yang diperuntukan kepada setiap soalan adalah seperti berikut:

No	Nama	Markah
1.	Pelagak Pengawal Mikro	200
2.	Laporan Aktiviti Juruwang	250
3.	Pemilihan Pasukan	300
4.	Pengira Pinjaman	100

10. Selain dinyatakan, program anda boleh ditulis dengan menggunakan mana-mana bahasa pengaturcaraan utama di bawah mana-mana system operasi utama.
11. Selain dinyatakan, anda boleh menggunakan semua fungsi piawaian perpustakaan dalam bahasa pengaturacaraan 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 dijangka menjawab semua soalan.

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

5 PELAGAK PENGAWAL MIKRO

Syarikat Alat Mainan Giveaway merupakan pengeluar alat-alat mainan plastik dalam kuantiti yang banyak untuk dibekalkan kepada kedai-kedai makanan segera. Alat-alat mainan tersebut akan disertakan dalam bungkusan burger untuk kanak-kanak.

Giveaway sedang memikirkan cara bagaimana untuk menggunakan pengawal mikro yang berkos rendah untuk mengaktifkan bunyi, lampu sinar serta menghidup dan mematikan motor. Tugas anda adalah memprogramkan satu pelagak untuk pengawal mikro tersebut. Pelagak tersebut akan digunakan untuk membina dan menguji program pengawal alat mainan.

Pengawal mikro tersebut mempunyai satu pengumpul 8 bit 'A', satu penunjuk tindanan 'SP', satu bit pembawa 'C', dan satu pembilang program 'PC'. Pengawal mikro tersebut mempunyai ruang alamat data 8 bit dan ruang alamat program 16 bit yang berasingan. Pengawal mikro tersebut berkomunikasi dengan peranti persisian melalui 8 pin input dan 8 pin output.

Ruang alamat data 8 bit tersebut distrukturkan seperti di bawah:

Alamat	Fungsi
0x00 hingga 0xf0	Ingatan capaian rawak untuk gunaan umum yang bersaiz 240 bait.
0xf0 hingga 0xf7	8 pendaftar input/output.
0xf8 hingga 0xff	Tidak digunakan.

8 pendaftar input/output tersebut digunakan untuk membaca status 8 input pin dan menetapkan status 8 output pin.

Jika pengumpul dimuatkan dari mana-mana satu daripada 8 pendaftar input/output, bit terkanan pengumpul tersebut akan ditetapkan dengan status input pin yang sepadan. Bit yang lain pada pengumpul tersebut akan direset.

Jika pengumpul tersebut disimpan ke mana-mana satu daripada 8 pendaftar input/output, pin output yang sepadan akan ditetapkan kepada status bit terkanan pengumpul tersebut. Bit yang lain pada pengumpul tidak mempunyai sebarang kesan dan mungkin mengandungi sembarangan data.

Ruang alamat program 16 bit tersebut dipetakan kepada ingatan baca sahaja yang telah dimuatkan dengan program pelaksanaan ketika pengawal mikro itu dikilang.

Pengawal mikro tersebut mempunyai set suruhan seperti berikut:

Suruhan	Kandungan ingatan program	Pemprosesan (operator C, semua pendaftar dan data adalah positif)	Penerangan
LD A,#val	0x10 val	A = val; PC += 2;	Muat pengumpul, segera

LD A,adr	0x11 adr	A = data[adr]; PC += 2;	Muat pengumpul, terus
LD SP,#val	0x12 val	SP = val; PC += 2;	Muat penunjuk tindanan, segera
ST adr,A	0x13 adr	data[adr] = A; PC += 2;	Simpan pengumpul, terus.
ADD A,adr	0x20 adr	C = A + data[adr] >= 256; A = A + data[adr] & 255; PC += 2;	Tambah, terus
SUB A,adr	0x21 adr	C = A – data[adr] < 0; A = A – data[adr] & 255; PC += 2;	Tolak, terus

BRA ofs	0x30 ofs	if (ofs < 128)
		  PC += 2 + ofs;
		else
		  PC += ofs – 254;
			Cabangan

BCS ofs	0x31 ofs	if (C) {
		  if (ofs < 128)
		  PC += 2 + ofs;
		  else
		  PC += ofs – 254;
		} else {
		  PC += 2;
		}
			Cabangan jika pembawa diset

JMP hi-lo	0x40 hi lo	PC = hi << 8 | lo;	Lompat, terus

CALL hi-lo	0x41 hi lo	data[--SP] = PC+3 & 255;
		data[--SP] = PC+3 >> 8;
		PC = hi << 8 | lo;
			Panggil sub-rutin

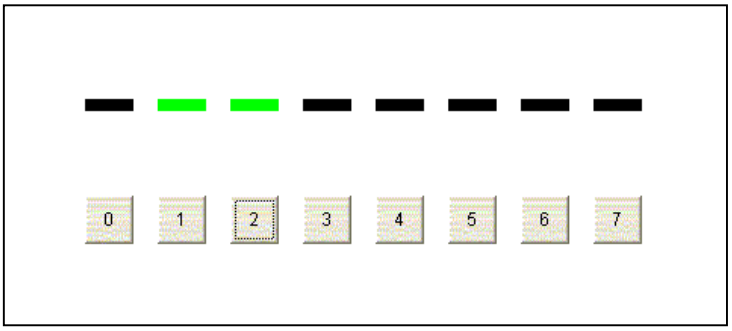
RET	0x42	PC = data[SP]<<8 | data[SP+1]; SP += 2;	Kembali dari sub-rutin

Dalam jadual di atas, ‘data[x]’ adalah satu rujukan kepada kandungan di lokasi ‘x’ dalam ruang alamat data 8 bit.

Apabila pengawal mikro diaktifkan, ia akan mula melaksanakan arahan di alamat sifar pada ruang alamat program 16 bit.

Pelagak tersebut mesti memuatkan program yang perlu dilaksanakan dari satu fail binari. Fail tersebut mungkin lebih kecil dari kapasiti 64Kb ruang alamat program 16 bit itu, di mana pada situasi begini, kandungan fail tersebut mesti dimuatkan bermula di alamat sifar.

Pelagak tersebut mesti mempunyai satu panel kawalan yang mempunyai 8 lampu penunjuk dan 8 butang seperti dalam gambar berikut.



8 lampu penunjuk tersebut mesti memaparkan status pelagakan 8 pin output tersebut. Apabila program tersebut menyimpan nilai 1 kepada sesebuah pendaftar input/output, lampu penunjuk yang betul mesti memaparkan warna hijau, dan apabila program tersebut menyimpan nilai 0 kepada sesebuah pendaftar input/output, lampu penunjuk yang betul mesti memaparkan warna hitam.

8 butang tersebut mesti mengawal status pelagakan bagi 8 pin input tersebut. Apabila butang ditekan, program yang dilaksanakan dalam pelagak mesti membaca nilai 1 daripada pendaftar input/output yang sepadan, dan apabila butang tersebut dilepaskan, program tersebut mesti membaca nilai 0 daripada pendaftar input/output yang sepadan.

6 LAPORAN AKTIVITI JURUWANG

Angina Marketing menguruskan beberapa rangkaian kedai makanan segera. Di antara jenama Angina Marketing termasuklah “Fried Fowl”, “Ground Bull and Chease” dan “Pastry”, yang masing-masing memasarkan ayam goreng, burger dan pizza. Sebilangan besar rangkaian kedai Angina dibukadi lapangan terbang dan stesen bas serta beroperasi selama 24 jam sehari.

Angina mengekalkan satu pangkalan data yang mengaitkan setiap giliran juruwang di setiap lokasi juruwang yang berada pada setiap restorannya. Maklumat tersebut disimpan dalam pangkalan data yang mempunyai skema seperti berikut:

```
create table transactions (
     transNo              integer not null,
     tradingDate          char(10),
     locCode              char(6) not null,
     shiftNo              smallint not null,
     transType            smallint not null,
     transVal             integer not null
);
create index transNoInd on transsactions(transNo);
```

`transNo`
: adalah nombor transaksi. Setiap transaksi baru akan diperuntukkan satu nombor transaksi yang berikutnya dalam turutan.

`tradingDate`

adalah tarikh urus niaga bagi giliran juruwang tersebut. Setiap giliran juruwang diperuntukkan satu tarikh urus niaga nominal. Tarikh urus niaga tersebut adalah dalam format ISO 8601 (iaitu YYYY-MM-DD). Tarikh urus niaga tersebut tidak semestinya sama dengan tarikh sebenar. Sebagai contoh, satu giliran yang bermula pada 10:00pm dan tamat pada 6:00am hari seterusnya akan mempunyai tarikh urus niaga di mana giliran tersebut bermula, dengan hari urus niaga ditutup pada 6:00am hari seterusnya.

`locCode`

adalah kod lokasi juruwang. Setiap lokasi juruwang mempunyai kod lokasi yang berbeza.

`shiftNo`

adalah nombor giliran untuk giliran juruwang. Sebagai contoh, jika sesuatu lokasi juruwang mempunyai 3 giliran yang setiap satu bertempoh 8 jam, giliran pertama mungkin adalah giliran 1, giliran kedua giliran 2, dan ketiga giliran 3. Nombor giliran boleh dilangkau. Sebagai contoh, satu lokasi boleh mempunyai data bagi giliran 1 dan 3, tetapi tanpa data untuk giliran 2.

`transType`

adalah kod untuk jenis transaksi. Jenis kod transaksi disenaraikan pada jadual di bawah.

`transVal`

adalah nilai transaksi dalam sen. Sebagai contoh, satu transaksi dengan nilai $12.34 akan disimpan sebagai integer 1234.

Kod jenis transaksi tersebut adalah:

Nilai	Nama	Makna
1	Imbangan pembukaan	Wang yang dihantar dari giliran juruwang yang sebelumnya.
2.	Pengisian	Wang yang dipindah ke lokasi juruwang untuk menambah nilai apungan pada ketika di mana lokasi juruwang tersebut tidak mempunyai kuantiti yang mencukupi bagi sesuatu wang kertas.
3	Kredit	Wang yang dipindahkan dari lokasi juruwang ke peti keselamatan.
4	Jualan	Wang yang diterima daripada pelanggan sebagai pertukaran untuk pembelian.
5	Imbangan penutupan	Wang yang dibawa ke depan kepada juruwang yang seterusnya.

Angina sekarang memerlukan satu laporan untuk meringkaskan maklumat dalam jadual transaksi.

Program yang menghasilkan laporan tersebut mesti menerima satu tempoh laporan yang diungkapkan sebagai satu “tarikh-dari” dan “tarikh-hingga”, dan menjana laporan berdasarkan kandungan berikut:

1. tarikh dan masa laporan itu dicetak;
2. tempoh laporan (tarikh-dari dan tarikh-hingga);
3. bagi setiap giliran juruwang dalam tempoh laporan:
    a. kod lokasi juruwang,
    b. tarikh urus niaga,
    c. nombor giliran,
    d. imbangan pembukaan,
    e. jumlah pengisian dalam giliran,
    f. jumlah kredit dalam giliran,
    g. jumlah jualan dalam giliran,
    h. imbangan penutupan;
4. jumlah bagi semua giliran untuk:
    a. imbangan pembukaan,
    b. pengisian,
    c. kredit,
    d. jualan,
    e. imbangan penutupan.

Jumlah untuk semua giliran bagi imbangan pembukaan dan penutupan bukanlah hasil tambah bagi lajur imbangan pembukaan dan penutupan. Jumlah untuk imbangan pembukaan adalah hasil tambah semua imbangan pembukaan giliran pertama bagi setiap lokasi juruwang dalam tempoh laporan, dan jumlah untuk imbangan penutupan adalah hasil tambah semua imbangan penutupan giliran terakhir bagi setiap lokasi juruwang dalam tempoh laporan tersebut. Sebagai contoh:

Lokasi	**Tarikh**	**Giliran**	**Imbangan pembukaan**	**Pengisian**	**Kredit**	**Jualan**	**Imbangan penutupan**
FF-1	24-09-06	1	1,200.00	100.00	2,500.00	2,700.00	1,500.00
FF-1	24-09-06	2	1,500.00	0.00	2,800.00	2,500.00	1,200.00
FF-2	24-09-06	1	800.00	200.00	1,700.00	1,950.00	1,250.00
FF-2	24-09-06	2	1,250.00	0.00	2,200.00	1,850.00	900.00
JUMLAH			2,000.00 (1,200+800)	300.00	9,200.00	9,000.00	2,100.00 (1,200+900)

Laporan tersebut mesti diaturkan seperti di bawah:

```
         1         2         3         4         5         6         7         8
1...5....0....5....0....5....0....5....0....5....0....5....0....5....0....5....0
DD-MM-YY HH:MM              CASHIER ACTIVITY REPORT                     PAGE X-X
                           for XX-XX-XX to XX-XX-XX

Cashier                    Opening                                      Closing
Location Date     Shift    balance      Fills    Credits      Sales    balance

X------X XX-XX-XX   X   XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX
X------X XX-XX-XX   X   XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX
                       ---------- ---------- ---------- ---------- ----------
TOTAL                  XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX XXX,XXX.XX
                       ========== ========== ========== ========== ==========
```

Walaupun tarikh urus niaga disimpan dalam format YYYY-MM-DD dalam pangkalan data, semua input dan output tarikh yang diterima daripada pengguna mestilah dalam format DD-MM-YY.

Baris laporan mesti disusun mengikut kod lokasi juruwang, tarikh urus niaga dan nombor giliran.

Jika satu giliran dalam tempoh laporan masih dalam urus niaga (iaitu tiada transaksi imbangan penutupan dalam pangkalan data), program yang menghasilkan laporan tersebut mesti menghitung imbangan semasa di lokasi tersebut dan memaparkannya pada lajur imbangan penutupuan. Untuk tujuan ini:

Imbangan penutupan = imbangan pembukaan + pengisian – kredit + jualan

Angina ingin menggunakan laporan tersebut untuk menyelaraskan imbangan di semua lokasi juruwang pada akhir setiap hari urus niaga. Oleh yang demikian, program tersebut harus menjanakan laporan bagi satu hari dalam masa kurang daripada 10 saat.

Oleh sebab jadual transaksi tersebut mengandungi satu baris bagi setiap jualan makanan ayam, burger atau pizza, jadual tersebut amatlah besar. Pada suatu ketika yang lalu, jurutera perisian pernah memeriksa jadual tersebut dan mendapati ia mengandungi lebih daripada 200 juta baris, yang mewakili data untuk kira-kira 5 tahun. Sebarang pencarian yang menggunakan kunci yang selain daripada nombor transaksi akan menyebabkan satu pengimbasan bersiri ke atas jadual tersebut, dan akan mengambil masa kira-kira 2.5 jam. Walaupun demikian, pencarian data yang berkait dengan sesuatu nombor transaksi adalah agak cepat, iaitu mengambil masa lebih kurang 0.5ms sahaja. Dahulu, jurutera perisian di Angina pernah cuba membina indeks untuk kunci yang lain, tetapi kesannya terhadap pemprosesan transaksi latar depan adalah sukar diterima.

Jurutera persisian di Angina mendapati bahawa nombor tansaksi boleh digunakan sebagai "indeks kabur" untuk tarikh urus niaga. Walaupun semua lokasi juruwang beralih dari satu tarikh urus niaga ke tarikh urus niaga yang seterusnya pada ketika yang berbeza, pada mana-mana ketika apabila seseorang juruwang berurus-niaga pada hari 'd', lokasi juruwang yang lain adalah berurus-niaga sama ada pada hari sebelum 'd', hari 'd' atau hari selepas 'd'. Mereka mencadangkan bahawa fenomena ini digunakan untuk mengurangkan skop pencarian.

7 PEMILIHAN PASUKAN

Nautical Defence Systems (NDS) ialah sebuah syarikat besar yang berkait dengan kerajaan dan mengusahakan projek-projek tentera laut yang berkesulitan tinggi. Kerajaan telah mengarahkan NDS supaya memulakan satu projek yang membina sebuah kapal rondaan yang canggih.

Kapal-kapal rondaan tersebut akan dilengkapkan dengan beberapa sistem komputer yang kompleks, yang di mana sifat sebenarnya belum dikategorikan. Akan tetapi, adalah diketahui bahawa sistem tersebut merangkumi sistem pengenalpastian dan gerak balas bagi ancaman taktikal, sistem kawalan senjata dan sistem pengurusan kapal.

NDS tidak mempunyai sumber yang mencukupi untuk membina sistem tersebut secara sendirian dan perlu menggaji pasukan dari luar untuk melaksanakan kerja pembangunan perisian. Syarikat anda adalah salah satu daripada syarikat yang menyertai tender tersebut.

NDS telah mentakrifkan satu skim untuk menilai pasukan yang menyertai tender kontrak tersebut. Tugas anda adalah mengaturcara satu program yang memilih ahli pasukan daripada staf dalam syarikat anda supaya pasukan tersebut mempunyai markah penilaian secara keseluruhan yang tertinggi.

Markah penilaian pasukan tersebut diungkapkan seperti berikut:

$$Rt = \frac{\sum_{i=1}^{n} Rp_i}{Y^{(n-1)}}$$

Di mana:

Rt	adalah markah penilaian untuk pasukan tersebut.
n	adalah bilangan ahli pasukan
Rp_i	adalah markah penilaian bagi pengaturcara yang ke-i
Y	adalah nilai malar 1.1

Markah penilaian bagi seseorang pengaturcara adalah jumlah nilai bagi 5 projek terbaik dalam 5 tahun yang lepas. Nilai projek adalah nombor nyata di antara 0 dan 10. Kaedah yang digunakan untuk menilai projek telah disempurnakan dan adalah di luar lingkungan tugas ini.

Jika dua atau lebih pengaturcara terlibat dalam satu projek bersama, projek tersebut hanya boleh dimasukkan ke dalam markah penilaian salah seorang pengaturcara sahaja. Markah penilaian bagi pengaturcara yang lain mesti diterbit dari projek yang lain.

Jika bilangan projek seseorang pengaturcara terlibat dalam lima tahun yang lepas adalah kurang daripada lima, atau jika selepas projek bersama yang telah diambilkira ke dalam markah penilaian pengaturcara lain dikecualikan, pengaturcara tersebut tinggal kurang daripada lima projek, maka jumlah nilai bagi bilangan projek yang lebih kecil itu merupakan markah penilaian bagi pengaturcara itu.

Nilai-nilai projek disimpan dalam satu pangkalan data SQL yang mempunyai skema seperti berikut:

```
create table programmers (
     progNum        integer not null,
     progName       char(20) not null
);
create table projects (
     projNum        integer not null,
     projName       char(20) not null,
     projValue      double precision not null
);
create table assignments (
     asnProgNum     integer not null,
     asnProjNum     integer not null
);
```

`programmers`
: ialah satu jadual yang mengandungi satu baris untuk setiap pengaturcara.

`progNum`
: ialah satu nombor pengaturcara unik untuk mengenalpasti seseorang pengaturcara.

`progName`
: ialah nama pengaturcara.

`projects`
: ialah satu jadual yang mengandungi satu baris untuk setiap projek.

`projNum`
: ialah satu nombor projek unik untuk mengenalpasti sesuatu projek.

`projVal`
: ialah nilai projek dalam julat 0 hingga 10.

`assignments`
: ialah satu jadual yang mengaitkan seseorang pengaturcara kepada projek-projek yang beliau mengambil bahagian dalam 5 tahun yang lepas.

`asnProgNum`
: ialah nombor pengaturcara.

`asnProjNum`
: ialah nombor projek bagi projek yang pengaturcara tersebut terlibat dalam 5 tahun yang lepas.

Pangkalan data tersebut mengandungi 75 orang pengaturcara, 324 buah projek dan 339 tugasan (assignments).

Aturcara anda mesti memilih pasukan yang mencapai markah penilaian yang tertinggi dari antara semua alternatif dalam pangkalan data. Bilangan pengaturcara di dalam sesuatu pasukan adalah tidak terhad asalkan gabungan peserta di dalam sesuatu pasukan dapat menghasilkan markah penilaian tertinggi. Sebagai tambahan, aturcara tersebut mesti direka supaya ia boleh menyempurnakan laporan dalam satu jam pada komputer yang pantas. Ia mesti memaparkan maklumat pasukan seperti berikut:

1. tarikh dan masa aturcara itu dilaksanakan;
2. bagi setiap ahli pasukan:
    a. nombor pengaturcara;
    b. nama pengaturcara;
    c. bagi setiap projek yang diambil kira dalam markah penilaian pengaturcara individu:
        i. nombor projek,
        ii. nama projek,
        iii. nilai projek;
    d. markah penilaian individu untuk pengaturcara;
3. bilangan ahli pasukan;
4. markah penilaian keseluruhan untuk pasukan tersebut.

Maklumat tersebut harus dipaparkan dalam format seperti berikut:

```
          1         2         3         4         5         6         7         8
1...5....0....5....0....5....0....5....0....5....0....5....0....5....0....5....0
DD-MM-YY HH:MM               TEAM SELECTION REPORT                      PAGE X-X

Programmer                    Project
Number  Name                  Number  Name                  Value

X----X  X------------------X  X----X  X------------------X  XX.X
                              X----X  X------------------X  XX.X
                                                            ----
                              Programmer total              XX.X

X----X  X------------------X  X----X  X------------------X  XX.X
                              X----X  X------------------X  XX.X
                                                            ----
                              Programmer total              XX.X
                                                           -----
Team total (X members)                                     XXX.X
```

8 PENGIRA PINJAMAN

Seorang pengguna perlu membandingkan pelbagai kos pengurusan kewangan yang diperlukan untuk membeli sesuatu barangan permodalan. Akan tetapi, cadangan yang diberi oleh pelbagai bank tidak mengizinkan perbandingan yang mudah di antara alternatif-alternatif tersebut. Setiap bank mempunyai skim pembayarannya tersendiri, yang menyebabkan kesukaran dalam membandingkan kos berkaitan dengan sesuatu pinjaman dengan alternatif yang lain.

Pengguna tersebut meminta supaya anda membangunkan satu program yang menerima parameter-parameter seperti di bawah pada satu skrin input data:

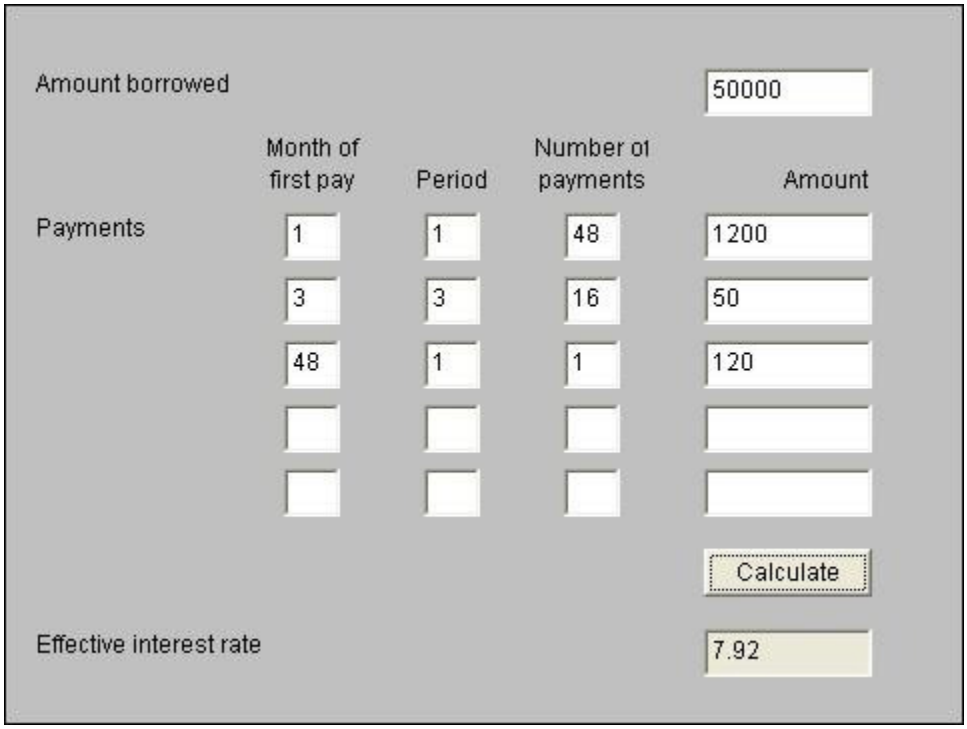
1. amaun dipinjam (wang pokok);
2. bagi setiap jenis pembayaran (maksima sehingga 5 jenis):
    a. nombor bulan bagi pembayaran pertama,
    b. tempoh di antara pembayaran yang berturutan dalam bulan,
    c. bilangan pembayaran,
    d. amaun bagi setiap pembayaran;

Jenis pembayaran termasuklah yuran penubuhan, yuran penyelenggaraan akaun, pembayaran balik, yuran penutupan akaun dan sebagainya.

Nombor bulan bermula pada sifar, bagi pembayaran yang dibuat pada ketika wang pokok dipinjam, satu untuk pembayaran yang perlu dibuat pada satu bulan selepas wang pokok dipinjam dan sebagainya.

Skrin input data tersebut mesti mempunyai satu butang ‘Calculate’. Apabila butang tersebut diklik, program tersebut mesti mengira dan memaparkan kadar faedah efektif bagi pinjaman tersebut.

Skrin input data tersebut harus diatur seperti di bawah:



Kadar faedah efektif tersebut bagi pinjaman itu adalah kadar faedah tahunan 'R' yang memuaskan persamaan berikut:

$$\mathrm{P} = \sum_{i=1}^{n} \frac{v_i}{\left(1 + \frac{\mathrm{R}}{1200}\right)^{t_i}}$$

Di mana:

P	ialah amaun dipinjam (wang pokok);
n	ialah nombor bagi pembayaran individu;
v_i	ialah amaun bagi pembayaran individu;
t_i	ialah nombor bulan (iaitu masa) di mana pembayaran itu perlu dibuat;
R	ialah kadar faedah kompaun bulanan yang efektif, sebagai satu peratusan.

Program tersebut harus mengira dan memaparkan kadar faedah efektif sehingga dua tempat perpuluhan.

Program tersebut mesti menolak parameter yang tidak sah (contohnya, nilai bukan angka). Jika sesuatu parameter adalah tidak sah, program tersebut harus menonjolkan parameter yang tidak sah tersebut dengan memaparkannya pada latar belakang yang berwarna merah dan memaparkan 'ERROR' pada ruang kadar faedah efektif.

Program tersebut boleh menolak pengiraan yang menghasilkan kadar faedah yang berada di luar julat di antara 0 hingga 100%. Jika program tersebut menolak parameter yang menghasilkan kadar faedah yang berada di luar julat, ia harus memaparkan '****' pada ruang kadar faedah efektif.