

Q2.

Read the following description of a relational database, and then answer Subquestions 1, 2 and 3.

A certain database consists of the following employee and department table. An employee works for a department and may or may not have a manager.

emp (employee) table

empno	ename	job	mgr	hiredate	sal	comm	deptno
-------	-------	-----	-----	----------	-----	------	--------

dept (department) table

deptno	dname	loc
--------	-------	-----

### Subquestion 1

A display of the highest earner of each job is required, wherein the employees of each job category would be compared to the highest salary within the category.

ename	job	Highest-Salary
-------	-----	----------------

From the answer group below, select the correct answers to be inserted into the blanks  in the following SQL statement.

```
SELECT      ename, job, 
FROM        emp
WHERE       sal IN (SELECT  FROM emp GROUP BY job)
ORDER BY    descending;
```

### Answer group:

- a) Highest-Salary
- b) MAX (sal)
- c) MAX (sal) Highest-Salary
- d) sal
- e) sal Highest-Salary
- f) salary

### Subquestion 2

Provided that the empno is also used to denote mgr for another employee, what would be the appropriate select statement to determine the number of distinct managers without listing them.

No of managers

### Answer group:

- a) select count (distinct (mgr)) "No of managers" from emp;
- b) select count (mgr (distinct)) "No of managers" from emp;
- c) select distinct (count (mgr)) "No of managers" from emp;
- d) select distinct (mgr) "No of managers" from emp;
- e) select mgr (distinct) "No of managers" from emp;

Q2.

Hãy đọc mô tả sau đây về một cơ sở dữ liệu quan hệ, và trả lời các câu hỏi con 1, 2 và 3.

Một cơ sở dữ liệu chứa bảng dữ liệu về nhân viên và phòng ban như dưới đây. Mỗi nhân viên làm việc cho một phòng, và có thể có hoặc không có trưởng phòng.

### Câu hỏi con 1

Cần phải hiển thị người có lương cao nhất của từng công việc, trong đó các nhân viên trong từng nhóm công việc sẽ được so sánh với mức lương cao nhất trong nhóm đó.

Trong nhóm các câu trả lời dưới đây, hãy chọn các câu trả lời đúng để điền vào các khoảng trống  trong câu lệnh SQL sau đây.

### Nhóm câu trả lời:

- ☒ a) Highest-Salary
- ☒ b) MAX (sal)
- ☒ c) MAX (sal) Highest-Salary
- ☒ d) sal
- ☒ e) sal Highest-Salary
- ☒ f) salary

### Câu hỏi con 2

Biết rằng empno cũng được dùng để phân biệt mgr với các nhân viên khác, câu lệnh nào dưới đây là thích hợp để tìm số trưởng phòng mà không cần phải liệt kê họ ra.

No of managers

### Nhóm câu trả lời:

F → I

- b) Move the top disk from peg FROM to peg TO
- c) Move the top disk from peg INTERMEDIATE to peg FROM
- d) Move the top disk from peg INTERMEDIATE to peg TO
- e) Move the top disk from peg TO to peg FROM
- f) Move the top disk from peg TO to peg INTERMEDIATE

**Subquestion 3**

From the answer group below, select the correct answers to be inserted into the blanks  J  through  L .

The Towers of Hanoi game moves the disks 7 times for 3 disks ( $n=3$ ).

This program moves the disks  J  times for 4 disks,  K  times for 5 disks, and  L  times for  $n$  disks.

**Answer group:**

- a) 11      b) 15      c) 31
- d) 32      e)  $2n + 1$       f)  $2n - 1$

- b) Move the top disk from peg FROM to peg TO
- c) Move the top disk from peg INTERMEDIATE to peg FROM
- d) Move the top disk from peg INTERMEDIATE to peg TO
- e) Move the top disk from peg TO to peg FROM
- f) Move the top disk from peg TO to peg INTERMEDIATE

**Câu hỏi con 3**

Trong nhóm các câu trả lời dưới đây, hãy chọn các câu trả lời đúng để điền vào các ô trống từ  J  đến  L .

Trò chơi Tháp Hanoi dịch chuyển đĩa 7 lần đối với 3 đĩa ( $n=3$ ).

Chương trình này dịch chuyển đĩa  J  lần đối với 4 đĩa,  K  lần đối với 5 đĩa, và  L  lần đối với  $n$  đĩa.

**Nhóm câu trả lời:**

- a) 11      b) 15      c) 31
- d) 32      e)  $2n + 1$       f)  $2n - 1$

$2^n - 1$

Q5.

Read the following description of a program design, and then answer Subquestion.

[Program Description]

One program is developed to classify the students into two classes X and Y at a training center. Class X contains the more competent students and Class Y contains less competent students who didn't pass the tests that organized in the center. The center has the appropriate training plans for each class to get best result.

- 1) There are 80 students registered for the training.
- 2) Each test consists of 80 questions. The questions are classified by the topics. The number of questions and minimum number of correct answers (pass criteria) in each topic for each test is shown in the Table1.

Table1 Topic\_Table (Number of questions for each topic and pass test criteria)

Row ↓	← Column	
	0	1
0	Number of questions	Minimum number of correct answers
0	6	4
1	40	32
2	10	6
3	6	4
4	6	4
5	6	4
6	6	4

- 3) 80 questions are ordered by the topics as shown below:

Questions 1 to 6 belong to Topic 1  
Questions 7 to 46 belong to Topic 2  
Questions 47 to 56 belong to Topic 3  
Questions 57 to 62 belong to Topic 4  
Questions 63 to 68 belong to Topic 5  
Questions 69 to 74 belong to Topic 6  
Questions 75 to 80 belong to Topic 7

- 4) StudentID and test results of 80 questions (1: correct, 0: incorrect) for all 80 students are stored in the Master\_Table as shown in the Table 2.

StudentID																pass	Column
Row	0	1	2	3	4	5	6	...	j	..	77	78	79	80	81		
0		1	1	1	0	1	0	0	..		1	1	1	1			
1		2	1	1	1	0	0	1	..		1	1	0	1			
2		3	1	1	1	1	1	1	..		1	1	1	1			
3		4	1	1	1	0	1	0	..		0	1	1	1			
		..															
i		i															
79		80	1	1	0	1	0	0	..	..	1	1	1	1			

- 5) An element Master\_Table [i,j], where (0 ≤ i ≤ 79, 1 ≤ j ≤ 80), corresponds to student "i" and the answer "j".

- 6) For clarity, an example is given by extraction of the last row from the Table2. After a test, the test answers made by the student whose StudentID is 80 are looked like this :

Q5.

Hãy đọc mô tả sau đây về thiết kế chương trình, rồi trả lời câu hỏi con.

[Mô tả chương trình]

Một chương trình được xây dựng để phân loại sinh viên thành hai lớp X và Y tại một trung tâm đào tạo. Lớp X gồm các sinh viên khá, lớp Y bao gồm các sinh viên kém đã thi trượt bài thi của trung tâm. Trung tâm có một số kế hoạch đào tạo phù hợp cho từng lớp để thu được kết quả tốt hơn.

- 1) Có cả thầy 80 học sinh đăng ký trong khoá đào tạo.

- 2) Mỗi bài thi có 80 câu hỏi. Các câu hỏi được phân loại theo chủ đề. Số câu hỏi và số câu trả lời đúng tối thiểu để đối với từng chủ đề được biểu diễn trong Bảng 1.

Bảng 1 Topic\_Table (Số câu hỏi của từng chủ đề và điểm chuẩn)

Topic's comment

Topic 1: Computer Science fundamentals  
Topic 2: Computer system  
Topic 3: System development and operation  
Topic 4: Network Technology  
Topic 5: Database Technology  
Topic 6: Security and Standardization  
Topic 7: Computerization and management

- 3) 80 câu hỏi xếp theo thứ tự chủ đề như sau:

Câu hỏi từ 1 đến 6 thuộc Chủ đề 1  
Câu hỏi từ 7 đến 46 thuộc Chủ đề 2  
Câu hỏi từ 47 đến 56 thuộc Chủ đề 3  
Câu hỏi từ 57 đến 62 thuộc Chủ đề 4  
Câu hỏi từ 63 đến 68 thuộc Chủ đề 5  
Câu hỏi từ 69 đến 74 thuộc Chủ đề 6  
Câu hỏi từ 75 đến 80 thuộc Chủ đề 7

- 4) StudentID và kết quả trả lời 80 câu hỏi (1: đúng, 0: sai) của cả 80 sinh viên được lưu trong Master\_Table như biểu diễn trong bảng Table2.

- 5) Phần tử Master\_Table [i,j], với (0 ≤ i ≤ 79, 1 ≤ j ≤ 80), tương ứng với sinh viên thứ "i" và câu trả lời thứ "j".

- 6) Để cho rõ ràng, một ví dụ được đưa ra nhờ trích xuất dữ liệu của dòng cuối cùng trong bảng Table2. Sau khi thi, các câu trả lời của sinh viên có StudentID bằng 80 trông như sau:

within a character constant, or to a piece of coding within a comment.

4) An appearance of `/*` and `*/` switches the "comment mode" between ON and OFF. However, this does not apply to a piece of coding within a character constant or to a piece of coding within a character string literal.

hoặc cho đoạn mã trong một chú thích.

4) Sự xuất hiện của `/*` và `*/` sẽ chuyển "chế độ chú thích" giữa ON và OFF. Tuy nhiên, điều này không áp dụng cho đoạn mã bên trong hằng ký tự hoặc cho đoạn mã bên trong hằng chuỗi.

**[Program 2]**

```
#include <stdio.h>
main()
{
    int c1, c2;
    int c_mode = 0; /* set comment mode to off */
    int quotel = 0; /* set character constant mode to off */
    int quote2 = 0; /* set character string literal mode to off */

    for ( c1 = getchar(); ( c2 = getchar()) != EOF; c1 = c2 ) {

        if ( !c_mode ) { /* when comment mode is off */
            /* '\ ' in character constant or character string literal? */
            if ( && c1 == '\\' ) {
                putchar(c1);
                putchar(c2);
                c2 = getchar();
                continue;
            }
            /* single quote which is not inside a character string literal? */
            else if ( !quote2 && c1 == '\'' )
                ;
            /* double quote which is not inside a character constant? */
            else if ( !quotel && c1 == '\"' )
                ;
            /* '/' and '*' which is not inside a character constant
               and character string literal? */
            else if ( && c1 == '/' && c2 == '*' ) {
                ;
                c2 = getchar();
                continue;
            }
            putchar(c1);
        }

        else {
            if ( c1 == '*' && c2 == '/' ) { /* end of comment? */
                ;
                c2 = getchar();
            }
        }
    }
    putchar(c1);
}
```

**[Chương trình 2]**

**Subquestion 2**

From the answer groups below, select the correct answers to be inserted into the blanks  in Program 2.

**Answer group for  A and  D:**

- a) !quotel
- b) !quote2
- c) (!quotel || !quote2)
- d) (!quotel && !quote2)
- e) (quotel || quote2)
- f) (quotel && quote2)

**Câu hỏi con 2**

Từ các nhóm câu trả lời dưới đây, lựa chọn các câu trả lời đúng để chèn vào các ô trống  trong chương trình 2.

**Nhóm câu trả lời cho  A và  D:**

- a) !quotel
- b) !quote2
- c) (!quotel || !quote2)
- d) (!quotel && !quote2)
- e) (quotel || quote2)
- f) (quotel && quote2)

**Answer group for ☐ B ☐ C and ☐ E:**

- a) c\_mode = !c\_mode
- b) c\_mode = quote1 && quote2
- c) quote1 = !quote1
- d) quote1 = !quote2
- e) quote1 = quote2
- f) quote2 = !quote1
- g) quote2 = !quote2
- h) quote2 = quote1

**Nhóm câu trả lời cho ☐ B ☐ C và ☐ E**

- a) c\_mode = !c\_mode
- b) c\_mode = quote1 && quote2
- c) quote1 = !quote1
- d) quote1 = !quote2
- e) quote1 = quote2
- f) quote2 = !quote1
- g) quote2 = !quote2
- h) quote2 = quote1

Q11.

Read the following description of a COBOL program and the program itself, and then answer Subquestion.

[Program Description]

Five television channels can be received in a certain geographical area. This program reads the view data file that records TV programs viewed by surveyed households in that area on a given day, and the TV program data file that records the TV program data on the same day. It then calculates the average audience rating and prints it. The program rounds down the time in units of one minute. For instance, the time 10:00 is the figure obtained by rounding down the time between 10:00:00 and 10:00:59.

1) The view data file (VIEW-FILE) is a sequential file with the following record format:

(i) The view data file stores the audience data for a given day collected from the surveyed 600 sample households.

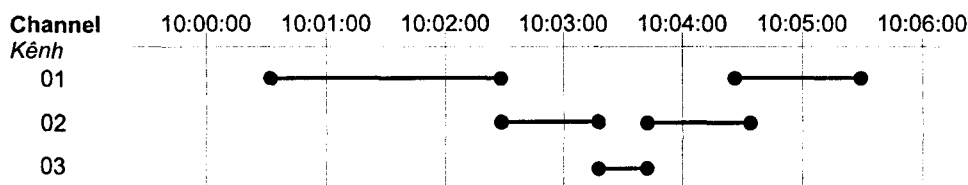
Channel No. 2 digits	Detection start time		Detection end time	
	Hour 2 digits	Minute 2 digits	Hour 2 digits	Minute 2 digits

(ii) Assume that there is one television set per household.

(iii) Assume that the channel numbers range from 01 to 05.

(iv) A record shows the channel viewed by a given household, indicating the "channel no.", the "detection start time" and the "detection end time."

(v) Audience data is obtained by detecting the channels viewed by surveyed households at 00 second of every minute. For instance, suppose that a certain household viewed TV from 10:00:30 to 10:05:30, during which it changed channels as follows:



Since channels are detected at 00 second of every minute, the data is recorded in the view file, rounded down in minute units as follows. Channel 03 was viewed but not recorded.

```
0110011002
0210031004
0110051005
```

(vi) The range of the detection start time and the detection end time is from 00:00 to 23:59. No data extends to the second day.

(vii) The detection start time is equal to or less than the detection end time.

2) The program data file (PROGRAM-FILE) is a sequential file with the following record format:

Q11.

Đọc phần mô tả sau và nội dung của một chương trình COBOL, và trả lời các câu hỏi con.

[Mô tả chương trình]

Năm kênh vô tuyến truyền hình có thể thu sóng được trong một vùng địa lý xác định. Chương trình này đọc tệp dữ liệu view, tệp này ghi lại các chương trình truyền hình (TV) được xem bởi các hộ gia đình được khảo sát trong vùng đó tại một ngày cụ thể, và tệp dữ liệu TV program, tệp này ghi lại dữ liệu chương trình truyền hình cùng ngày. Tiếp theo, nó tính toán tỷ lệ khán giả trung bình và in ra. Chương trình này làm tròn thời gian thấp xuống theo đơn vị phút. Ví dụ, thời gian 10:00 là con số thu được từ việc làm tròn thấp xuống của một thời điểm trong khoảng giữa 10:00:00 và 10:00:59.

1) Tệp dữ liệu view (VIEW-FILE) là một tệp tích lũy liên tục có dạng bản ghi lại như sau:

(i) Tệp dữ liệu view lưu trữ dữ liệu xem của một ngày cụ thể được thu thập từ 600 hộ gia đình mẫu được khảo sát.

Kênh số 2 chữ số	Thời điểm bắt đầu dò		Thời điểm kết thúc dò	
	Giờ 2 chữ số	Phút 2 chữ số	Giờ 2 chữ số	Phút 2 chữ số

(ii) Giả thiết rằng mỗi hộ gia đình có một vô tuyến truyền hình.

(iii) Giả thiết rằng các số kênh trải từ 01 đến 05.

(iv) Một bản ghi chỉ ra kênh được xem bởi một hộ gia đình cụ thể, chỉ rõ "Kênh số", "Thời gian bắt đầu bắt sóng" và "Thời gian kết thúc bắt sóng".

(v) Dữ liệu thu được bằng cách dò ra các kênh đang được xem bởi các hộ gia đình tại thời điểm 00 giây mỗi phút. Chẳng hạn như có một hộ gia đình đã xem vô tuyến từ 10:00:30 đến 10:05:30, trong thời gian đó hộ này đã thay đổi các kênh như sau:

Vì các kênh được bắt sóng tại thời điểm 00 giây mỗi phút, dữ liệu được ghi lại trong tệp view, và được làm tròn xuống theo đơn vị phút như sau. Kênh 03 đã được xem nhưng không ghi được.

```
0110011002
0210031004
0110051005
```

(vi) Phạm vi thời gian bắt đầu bắt sóng đến thời gian kết thúc bắt sóng từ 00:00 tới 23:59. Không có dữ liệu nào kéo dài đến ngày thứ hai.

(vii) Thời gian bắt đầu bắt sóng trùng hoặc trước thời gian kết thúc bắt sóng.

2) Tệp dữ liệu TV program (PROGRAM-FILE) là một tệp tích lũy liên tục có dạng bản ghi như sau:

Channel No. 2 digits	Detection start time		Detection end time	
	Hour 2 digits	Minute 2 digits	Hour 2 digits	Minute 2 digits

(i) The program data file stores program data for 5 channels, on the same day as in the view data file.

(ii) The range of the program start time and the program end time is from 00:00 to 23:59. No data extends to the second day.

(iii) The program start time is equal to or less than the program end time.

(iv) A record shows that a program is broadcast from the "program start time" to the "program end time."

3) The program executes as follows:

(i) The view count table (VIEW-COUNT-TABLE) is a two-dimensional table consisting of 5 (channels) à 1,440 (minutes). It summarizes the number of households by channel viewed in minute units and enters the data in each category of the table.

(ii) The average audience rating of a TV program is obtained using the following equation:

Average audience rating of program (%) =  
(Sum of the time during which each household viewed that program) / (Number of sample households × broadcast time of the program) × 100

(iii) The channel, program start time, program end time, average audience rating (in percent, to the first decimal place), and the program name are printed in the order in which programs were read from the program data file, in the following format:

channel	from - to	rating(%)	program-title
02	2320 - 2359	8.3	Sports News
04	0430 - 0439	1.7	Good Morning
:	:	:	:

[Program]

```
DATA DIVISION.
FILE SECTION.
FD VIEW-FILE.
01 VIEW-REC.
   05 VIEW-CHANNEL                PIC 99.
   05 VIEW-START-HHMM.
      10 VIEW-START-HH            PIC 99.
      10 VIEW-START-MM            PIC 99.
   05 VIEW-END-HHMM.
      10 VIEW-END-HH              PIC 99.
      10 VIEW-END-MM              PIC 99.
FD PROGRAM-FILE.
01 PROGRAM-REC.
   05 PROG-CHANNEL                PIC 99.
   05 PROG-START-HHMM.
      10 PROG-START-HH            PIC 99.
      10 PROG-START-MM            PIC 99.
```

Kênh số 2 chữ số	Thời điểm bắt đầu đỏ		Thời điểm kết thúc đỏ	
	Giờ 2 chữ số	Phút 2 chữ số	Giờ 2 chữ số	Phút 2 chữ số

(i) Tập dữ liệu chương trình lưu trữ dữ liệu cho 5 kênh trong cùng một ngày như trong tập dữ liệu view.

(ii) Phạm vi thời gian bắt đầu chương trình đến thời gian kết thúc chương trình là từ 00:00 tới 23:59. Không có dữ liệu kéo dài sang ngày thứ hai.

(iii) Chương trình bắt đầu thời gian trùng hoặc trước thời gian kết thúc chương trình.

(iv) Một bản ghi chỉ ra rằng một chương trình được phát sóng từ "Thời gian bắt đầu chương trình" tới "Thời gian kết thúc chương trình".

3) Chương trình này được thực hiện như sau

(i) Bảng VIEW-COUNT-TABLE là bảng 2 chiều gồm 5 (kênh)×1,440(phút). Nó tổng hợp số hộ gia đình theo kênh được xem theo đơn vị phút và điền dữ liệu đó vào mỗi mục trong bảng này.

(ii) Tỷ lệ khán giả trung bình của một chương trình TV tính được theo phương trình sau đây:

Tỷ lệ khán giả trung bình của một chương trình (%) =  
(tổng thời gian mỗi hộ gia đình xem chương trình đó) / (số hộ mẫu×thời gian phát sóng của chương trình đó) × 100

(iii) Kênh, thời gian bắt đầu chương trình, thời gian kết thúc chương trình, tỷ lệ khán giả trung bình (theo phần trăm tới vị trí thập phân đầu tiên), và tên chương trình được in theo thứ tự. Các chương trình được đọc ra từ tập dữ liệu chương trình theo dạng sau:

[Chương trình]

```

05  PROG-END-HHMM.
    10  PROG-END-HH          PIC 99.
    10  PROG-END-MM          PIC 99.
05  PROGRAM-TITLE           PIC X(50).
FD  OUT-FILE.
01  OUT-REC                  PIC X(100).
WORKING-STORAGE SECTION.
01  SAMPLE-SIZE              PIC 9(4)  VALUE 600.
01  M                        PIC 9(4).
01  END-OF-FILE              PIC X.
01  START-MMMM               PIC 9(4).
01  END-MMMM                 PIC 9(4).
01  SUMMATION                PIC 9(9)  BINARY.
01  PROG-RATING              PIC ZZ9.9.
01  VIEW-COUNT-TABLE.
    05  CHANNEL              OCCURS 5.
    10  COUNT-OF-MINUTE      OCCURS 1440 PIC 9(4)  BINARY.
01  O-DATA.
    05  FILLER                PIC X(3)  VALUE SPACE.
    05  O-PROG-CHANNEL        PIC 99.
    05  FILLER                PIC X(3)  VALUE SPACE.
    05  O-PROG-START-HHMM.
    10  O-PROG-START-HH       PIC 99.
    10  O-PROG-START-MM       PIC 99.
    05  FILLER                PIC X(3)  VALUE " - ".
    05  O-PROG-END-HHMM.
    10  O-PROG-END-HH         PIC 99.
    10  O-PROG-END-MM         PIC 99.
    05  FILLER                PIC X(2)  VALUE SPACE.
    05  O-PROG-RATING         PIC ZZ9.9.
    05  FILLER                PIC X(6)  VALUE SPACE.
    05  O-PROGRAM-TITLE       PIC X(50).
01  TITLE-LINE               PIC X(100)
    VALUE "channel from - to   rating(%)  program-title".
PROCEDURE DIVISION.
MAIN-PARAGRAPH.
    PERFORM EXPAND-VIEW-COUNT.
    PERFORM CALCULATE-RATING-AND-PRINT.
    STOP RUN.
EXPAND-VIEW-COUNT.
    A
    OPEN INPUT VIEW-FILE.
    MOVE "N" TO END-OF-FILE.
    PERFORM UNTIL END-OF-FILE = "Y"
        READ VIEW-FILE AT END
        MOVE "Y" TO END-OF-FILE
    NOT AT END
        PERFORM SET-VIEW-COUNT
    END-READ
    END-PERFORM.
    CLOSE VIEW-FILE.
CALCULATE-RATING-AND-PRINT.
    OPEN INPUT PROGRAM-FILE  OUTPUT OUT-FILE.
    WRITE OUT-REC FROM TITLE-LINE AFTER ADVANCING 1.
    B
    PERFORM UNTIL END-OF-FILE = "Y"
        READ PROGRAM-FILE AT END
        MOVE "Y" TO END-OF-FILE
    NOT AT END
        PERFORM CALCULATE-RATING
        MOVE PROG-CHANNEL      TO O-PROG-CHANNEL
        MOVE PROG-START-HHMM   TO O-PROG-START-HHMM
        MOVE PROG-END-HHMM     TO O-PROG-END-HHMM
        MOVE PROG-RATING       TO O-PROG-RATING
        MOVE PROGRAM-TITLE     TO O-PROGRAM-TITLE
        WRITE OUT-REC FROM O-DATA AFTER ADVANCING 1
    END-READ
    END-PERFORM.
    CLOSE PROGRAM-FILE OUT-FILE.

```



SET-VIEW-COUNT.

```
COMPUTE START-MMMM = VIEW-START-HH * 60 + VIEW-START-MM + 1.  
COMPUTE END-MMMM   = VIEW-END-HH   * 60 + VIEW-END-MM + 1.  
PERFORM VARYING M FROM START-MMMM BY 1 UNTIL M > END-MMMM
```

**C**

END-PERFORM.

CALCULATE-RATING.

```
COMPUTE START-MMMM = PROG-START-HH * 60 + PROG-START-MM + 1.  
COMPUTE END-MMMM   = PROG-END-HH   * 60 + PROG-END-MM + 1.
```

**D**

PERFORM VARYING M FROM START-MMMM BY 1 UNTIL M > END-MMMM

**E**

END-PERFORM.

COMPUTE PROG-RATING ROUNDED

= **F**

### Subquestion

From the answer groups below, select the correct answers to be inserted into the blanks in the above program.

Answer group for **A**, **B** and **D**:

- a) COMPUTE SUMMATION = END-MMMM - START-MMMM
- b) INITIALIZE VIEW-COUNT-TABLE
- c) MOVE "N" TO END-OF-FILE
- d) MOVE "Y" TO END-OF-FILE
- e) MOVE SAMPLE-SIZE TO SUMMATION
- f) MOVE ZERO TO SUMMATION
- g) MOVE ZERO TO VIEW-COUNT-TABLE

Answer group for **C** and **E**:

- a) ADD 1 TO COUNT-OF-MINUTE(VIEW-CHANNEL M)
- b) ADD 1 TO COUNT-OF-MINUTE(VIEW-CHANNEL M + 1)
- c) ADD 1 TO SUMMATION
- d) ADD COUNT-OF-MINUTE(PROG-CHANNEL M) TO SUMMATION
- e) COMPUTE SUMMATION = COUNT-OF-MINUTE(PROG-CHANNEL M + 1)
- f) COMPUTE SUMMATION = COUNT-OF-MINUTE(VIEW-CHANNEL M) + 1
- g) MOVE 1 TO COUNT-OF-MINUTE(VIEW-CHANNEL M)

Answer group for **F**:

- a)  $100 * \text{SUMMATION} / ((\text{END-MMMM} - \text{START-MMMM}) * (\text{SAMPLE-SIZE} + 1))$
- b)  $100 * \text{SUMMATION} / ((\text{END-MMMM} - \text{START-MMMM} + 1) * \text{SAMPLE-SIZE})$
- c)  $100 * \text{SUMMATION} / (\text{END-MMMM} - \text{START-MMMM}) / \text{SAMPLE-SIZE}$
- d)  $100 * \text{SUMMATION} / (\text{END-MMMM} - \text{START-MMMM} + 1) / (\text{SAMPLE-SIZE} + 1)$

### Câu hỏi con

Từ các nhóm câu trả lời dưới đây, chọn các câu trả lời đúng để chèn vào các ô trống trong chương trình trên.

Nhóm câu trả lời cho **A**, **B** và **D**:

- a) COMPUTE SUMMATION = END-MMMM - START-MMMM
- b) INITIALIZE VIEW-COUNT-TABLE
- c) MOVE "N" TO END-OF-FILE
- d) MOVE "Y" TO END-OF-FILE
- e) MOVE SAMPLE-SIZE TO SUMMATION
- f) MOVE ZERO TO SUMMATION
- g) MOVE ZERO TO VIEW-COUNT-TABLE

Nhóm câu trả lời cho **C** và **E**:

- a) ADD 1 TO COUNT-OF-MINUTE(VIEW-CHANNEL M)
- b) ADD 1 TO COUNT-OF-MINUTE(VIEW-CHANNEL M + 1)
- c) ADD 1 TO SUMMATION
- d) ADD COUNT-OF-MINUTE(PROG-CHANNEL M) TO SUMMATION
- e) COMPUTE SUMMATION = COUNT-OF-MINUTE(PROG-CHANNEL M + 1)
- f) COMPUTE SUMMATION = COUNT-OF-MINUTE(VIEW-CHANNEL M) + 1
- g) MOVE 1 TO COUNT-OF-MINUTE(VIEW-CHANNEL M)

Nhóm câu trả lời cho **F**:

- a)  $100 * \text{SUMMATION} / ((\text{END-MMMM} - \text{START-MMMM}) * (\text{SAMPLE-SIZE} + 1))$
- b)  $100 * \text{SUMMATION} / ((\text{END-MMMM} - \text{START-MMMM} + 1) * \text{SAMPLE-SIZE})$
- c)  $100 * \text{SUMMATION} / (\text{END-MMMM} - \text{START-MMMM}) / \text{SAMPLE-SIZE}$
- d)  $100 * \text{SUMMATION} / (\text{END-MMMM} - \text{START-MMMM} + 1) / (\text{SAMPLE-SIZE} + 1)$

Q12.

Read the following description of a Java program and the program itself, and then answer Subquestions 1 and 2.

[Program Description]

This program simulates the processing by an automated ticket gate installed at each station of a railroad line comprising four stations A, B, C, and D, with A serving as the starting point and D as the terminal point. On this line, the following two types of fare certificates can be used: one-way ticket (hereinafter, "ticket"); and prepaid card (hereinafter, "card").

A fare on the line is decided based on the distance between stations. The fare for up to 4 kilometers is 120 yen (this fare is called the base fare). An amount of 30 yen is added for each additional 2 kilometers. Any additional distance less than 2 kilometers is rounded up to 2 kilometers. For example, if the distance is 7 kilometers, the fare is 180 yen.

The station of embarkation is recorded on a ticket when a passenger passes through the automated ticket gate in that station to enter the platform area. When he/she passes through the automated ticket gate to leave the platform area in the station of disembarkation, the fare is calculated, and if the amount paid for the ticket is insufficient, the gate is closed to prevent him/her from leaving the platform area. Once a ticket is used, it becomes invalid. On this line, a passenger can enter the platform area through the automated ticket gate in any station, thus being able to embark, regardless of the station that issued the ticket. For instance, with a ticket issued at Station A, a passenger can enter the platform area through the automated ticket gate in Station B.

The station of embarkation is recorded on a card when a passenger enters the platform area through the automated ticket gate in that station. At this time, if the balance on the card is zero, the gate is closed to prevent him/her from entering the platform area. When he/she leaves the platform area through the automated ticket gate in the station of disembarkation, a fare adjustment is processed. Namely, the fare is calculated and is subtracted from the balance on the card. At this time, if this balance is less than the amount of the fare, the gate is closed to prevent him/her from leaving the platform area.

Class Line represents the railroad line. Method getFare calculates the fare on the basis of a given distance, and returns this information.

Class Gate signifies an automated ticket gate. Fields A, B, C and D in class Line are instances of Gate, and represent the automated ticket gates installed at stations A, B, C and D, respectively. The constructor and each method perform the following processing.

1) The constructor generates an instance of Gate. A station name is specified as the first argument, and the distance from station A, which is the starting point of the line, is specified as the second argument.

2) Method enter performs processing when a passenger enters the platform area through an automated ticket

Q12.

Đọc phần mô tả sau và nội dung của một chương trình Java, và trả lời các câu hỏi con 1 và 2.

[Mô tả chương trình]

Chương trình sau giả lập hoạt động của một cửa soát vé tự động được lắp đặt tại mỗi ga dọc theo một tuyến đường xe lửa, có bốn ga A, B, C, và D, với A có vai trò là điểm bắt đầu và D có vai trò là điểm kết thúc. Trên tuyến này, có hai loại chứng nhận tiền vé có thể được sử dụng: vé một chiều (gọi tắt là "ticket"); thẻ trả trước (gọi tắt là "card").

Giá vé trên tuyến này được quyết định dựa trên khoảng cách giữa các ga. Giá vé cho 4 km đầu tiên là 120 yen (giá vé này được gọi là giá cơ bản). Giá vé tăng lên thêm 30 yen cho mỗi lần đi thêm 2 km. Khoảng cách đi thêm ít hơn 2 km được làm tròn là 2 km. Ví dụ nếu khoảng cách là 7 km, giá vé sẽ là 180 yen.

Ga lên tàu được ghi lại trên vé khi một người hành khách đi qua cửa soát vé tự động của ga đó để vào khu vực sân ga. Khi người khách đi qua cửa soát vé tự động để rời sân ga tại một ga xuống tàu, giá vé sẽ được tính, và nếu số tiền đã trả của vé đó không đủ, cửa sẽ đóng để ngăn người khách đó rời sân ga. Vé đã sử dụng một lần sẽ không còn giá trị. Trên tuyến này, một người khách có thể vào sân ga qua cửa soát vé tự động ở bất cứ ga nào, vì vậy có thể bắt đầu chuyến đi với vé được mua ở một ga bất kỳ. Ví dụ, với vé mua ở ga A, một người khách có thể vào khu vực sân ga qua cửa soát vé tự động của ga B.

Ga lên tàu được ghi lại trên thẻ khi một người khách vào khu vực sân ga qua cửa soát vé tự động tại ga đó. Tại thời điểm này, nếu số tiền dư trong thẻ còn lại là 0, cổng này sẽ đóng để ngăn người khách đó không vào được sân ga. Khi người khách rời sân ga qua cửa soát vé tự động tại ga xuống tàu, tiền dư sẽ được điều chỉnh. Cụ thể là giá vé được tính và được trừ khỏi số tiền dư trong thẻ. Nếu số tiền này ít hơn giá vé, cửa sẽ đóng lại để ngăn người hành khách rời khỏi khu vực ga.

Lớp Line thể hiện tuyến đường xe lửa. Phương thức getFare tính giá vé theo một khoảng cách được cho, và trả lại thông tin này.

Lớp Gate biểu thị cho một cửa soát vé tự động. Các trường A, B, C và D trong lớp Line là các thể hiện (instance) của Gate, và tượng trưng cho các cửa soát vé tự động lắp đặt tại A, B, C và D một cách lần lượt. Phương thức khởi tạo (constructor) và mỗi phương thức thực hiện quy trình sau đây:

1) Phương thức khởi tạo sinh ra một thể hiện của Gate. Tên ga được ghi rõ trong biến (argument) đầu tiên, và khoảng cách từ ga A (là điểm bắt đầu của tuyến) được ghi rõ trong biến thứ hai.

2) Phương thức enter tiến hành xử lý khi một hành khách vào khu vực sân ga qua một cửa soát vé tự động.

gate. If the fare certificate is not valid, the gate is closed. If entrance processing is carried out normally, information on the station of embarkation is recorded on this fare certificate.

3) Method `exit` performs processing when a passenger leaves the platform area through an automated ticket gate. If the fare certificate is invalid in that, for example, the amount (balance) on the fare certificate is insufficient, then the gate is closed.

4) Methods `open` and `close` output messages for opening and closing a gate, respectively.

Abstract class `Ticket`, which represents a fare certificate for this line, is inherited in defining a ticket and a card. The constructor and each method perform the following processing.

1) The constructor retains a purchase amount on the fare certificate as the initial value.

2) Method `getValue` returns the fare certificate amount (balance) as it is at the point in time when it is called.

3) Method `adjustValue` performs fare adjustment processing if necessary.

4) Method `deduct` deducts the amount specified as an argument from the amount (balance) of the fare certificate, and updates the amount (balance).

5) Method `setOrigin` records a specified Gate as the station of embarkation. If null is specified, the record of the station of embarkation is deleted.

6) Method `getOrigin` returns the station of embarkation that is recorded. If this station is not recorded, null is returned.

Class `OneWayTicket` represents a ticket, and class `PrepaidCard` represents a card. In the processing of either fare certificate, an abstract method is implemented, and the method in class `Ticket` is overridden, as necessary.

[Program 1]

```
public final class Line {
    public static final Gate A = new Gate("A", 0);
    public static final Gate B = new Gate("B", 5);
    public static final Gate C = new Gate("C", 8);
    public static final Gate D = new Gate("D", 14);

    public static int getFare(int distance) {
        return 120 + (Math.max(distance - 3, 0) / 2) * 30;
    }
}
```

Nếu chứng nhận tiền vé không có giá trị, cửa sẽ đóng lại. Nếu xử lý vào ga được thực hiện bình thường, thông tin tại ga lên tàu được ghi lại trên chứng nhận tiền vé này.?

3) Phương thức `exit` tiến hành xử lý khi một hành khách rời khu vực ga qua một cửa soát vé tự động. Nếu chứng nhận tiền vé không còn giá trị (ví dụ khi số tiền dư trên chứng nhận tiền vé không đủ), cửa sẽ đóng lại.

4) Các phương thức `open` và `close` cho ra các thông điệp về việc mở và đóng một cửa nào đó.

Lớp trừu tượng (abstract class) `Ticket`, thể hiện một chứng nhận tiền vé cho tuyến này, được kế thừa khi định nghĩa một ticket và một card. Phương thức khởi tạo và từng phương thức thực hiện quy trình sau đây.

1) Phương thức khởi tạo duy trì số tiền mua trên chứng nhận tiền vé như giá trị khởi đầu.

2) Phương thức `getValue` trả lại số tiền trên chứng nhận tiền vé (số tiền dư) khi mà nó được gọi.

3) Phương thức `adjustValue` thực hiện quy trình điều chỉnh tiền vé nếu cần thiết.

4) Phương thức `deduct` khấu trừ số tiền được chỉ ra trong một biến ra khỏi số tiền (tiền dư) của chứng nhận giá vé, và sửa đổi số tiền (tiền dư) đó.

5) Phương thức `setOrigin` ghi lại một cửa xác định như một ga lên tàu. Nếu cửa có giá trị null, bản ghi về ga lên tàu đó bị xoá.

6) Phương thức `getOrigin` trả lại ga lên tàu mà đã được ghi lại. Nếu ga này không được ghi lại, giá trị null được trả lại.

Lớp `OneWayTicket` thể hiện một ticket, và lớp `PrepaidCard` thể hiện một card. Trong xử lý bất kỳ chứng nhận tiền vé nào, một phương thức trừu tượng được thực thi, và phương thức đó trong lớp `Ticket` bị ghi đè lên khi cần thiết.

[Chương trình 1]

[Program 2]

[Chương trình 2]

```
public class Gate {
    private final String name;
    private final int distance;

    public Gate(String name, int distance) {
        this.name = name;
        this.distance = distance;
    }

    public void enter(Ticket ticket) {
        if (ticket.isValid() && ticket.getOrigin() == null) {
            A;
            open();
        } else {
            close();
        }
    }

    public void exit(Ticket ticket) {
        Gate origin = ticket.getOrigin();
        if (origin != null) {
            int d = Math.abs(origin.distance - distance);
            int fare = Line.getFare(d);
            if (B) {
                ticket.adjustValue(fare);
                ticket.setOrigin(null);
                open();
                return;
            }
        }
        close();
    }

    private void open() { System.out.println(name + ": open"); }
    private void close() {
        System.out.println(name + ": closed");
    }
}
```

[Program 3]

[Chương trình 3]

```
public abstract class Ticket {
    private Gate origin;
    private int value;

    public Ticket(int value) {
        this.value = value;
    }

    public int getValue() { return value; }

    public void deduct(int amount) { value -= amount; }

    public void setOrigin(Gate gate) { origin = gate; }

    public Gate getOrigin() { return origin; }

    public abstract void adjustValue(int amount);

    public abstract boolean isValid();
}
```

[Program 4]

[Chương trình 4]

```
public class OneWayTicket extends Ticket {
    private boolean valid = true;

    public OneWayTicket(int value) {
        ☐ C;
    }

    public void setOrigin(Gate gate) {
        super.setOrigin(gate);
        if (gate == null)
            valid = false;
    }

    public void adjustValue(int amount) { }

    public boolean isValid() { return valid; }
}
```

[Program 5]

[Chương trình 5]

```
public class PrepaidCard extends Ticket {
    public PrepaidCard(int value) {
        ☐ C;
    }

    public void adjustValue(int amount) { deduct(amount); }

    public boolean isValid() {
        return getValue() > 0;
    }
}
```

**Subquestion 1**

From the answer groups below, select the correct answers to be inserted into the blanks ☐ in the above programs.

**Answer group for ☐ A:**

- a) ticket.setOrigin(Line.A)
- b) ticket.setOrigin(Line.D)
- c) ticket.setOrigin(null)
- d) ticket.setOrigin(this)
- e) ticket.setOrigin(ticket)

**Answer group for ☐ B:**

- a) ticket.getValue() < fare
- b) ticket.getValue() <= fare
- c) ticket.getValue() == fare
- d) ticket.getValue() > fare
- e) ticket.getValue() >= fare

**Answer group for ☐ C:**

- a) super()
- b) super(this)
- c) super(value)
- d) super(); this.value = value
- e) this(value)

**Câu hỏi con 1**

Từ các nhóm câu trả lời dưới đây, lựa chọn các câu trả lời đúng để chèn vào các ô trống ☐ trong các chương trình trên.

**Nhóm câu trả lời cho ☐ A:**

- a) ticket.setOrigin(Line.A)
- b) ticket.setOrigin(Line.D)
- c) ticket.setOrigin(null)
- A** d) ticket.setOrigin(this)
- e) ticket.setOrigin(ticket)

**Nhóm câu trả lời cho ☐ B:**

- a) ticket.getValue() < fare
- b) ticket.getValue() <= fare
- c) ticket.getValue() == fare
- d) ticket.getValue() > fare
- B** e) ticket.getValue() >= fare

**Nhóm câu trả lời cho ☐ C:**

- a) super()
- b) super(this)
- C** c) super(value)
- d) super(); this.value = value
- e) this(value)

### Subquestion 2

It was decided that a new type of fare certificate be sold. A fare certificate of this type permits a passenger to freely embark or disembark at all stations within 24 hours from the time that the fare certificate is issued. Twenty-four hours after such a fare certificate is issued, the passenger can leave the platform area in any station but cannot enter any such area. For this, it is desired that a new class that inherits abstract class Ticket be defined to permit the use of this type of certificate without making any modification to class Gate. Processing by the constructor and methods of this class are shown in the following table. From the answer group below, select the correct answers to be inserted into the blanks  in the table. Here, value in the answer group is a field value of class Ticket, and it is assumed that the initial value is set by the constructor and that the value is obtained by method getValue.

Constructor and methods	Processing
Constructor	<input type="text" value="D"/>
Method getValue	<input type="text" value="E"/>
Method setOrigin	The same as is defined in the superclass
Method getOrigin	The same as is defined in the superclass
Method adjustValue	No processing is performed (methods proper do not contain any statements).
Method isValid	<input type="text" value="F"/>

#### Answer group:

- a) 0 is returned at all times.
- b) An amount that is theoretically too large to be spent in 24 hours is set as the initial value of value.
- c) Method deduct is called, with amount as an argument.
- d) No processing is performed (methods proper do not contain any statements).
- e) The base fare on the line is returned at all times.
- f) The highest fare on the line (the maximum value among all fares) is set as the initial value of value, and the fare certificate issuance time is recorded.
- g) The same as is defined in the superclass
- h) true is returned only when the call time is within 24 hours from the fare certificate issuance time stored in the instance. In other cases, false is returned.
- i) true is returned only when value is not less than the base fare on the line. In other cases, false is returned.
- j) true is returned only when value is not less than the highest fare on the line (the maximum value among all fares). In other cases, false is returned.

### Subquestion 2

Có quyết định rằng một loại chứng nhận tiền vé mới được bán. Loại chứng nhận này cho phép một hành khách tự do lên hoặc xuống tàu tại tất cả các ga trong vòng 24 tiếng kể từ thời điểm chứng nhận vé tàu được phát hành. Hai mươi tư tiếng sau khi loại vé này được phát hành, người hành khách có thể rời khu vực sân ga của bất kỳ ga nào nhưng không thể vào lại bất kỳ khu vực ga nào như thế. Bởi vậy, mong muốn tạo ra một lớp mới kế thừa lớp trừu tượng Ticket và được phép sử dụng kiểu chứng nhận này mà không cần sửa lớp Gate. Từ nhóm câu trả lời dưới đây, lựa chọn các câu trả lời đúng để điền vào các ô trống  trong bảng. Ở đây, value trong nhóm câu trả lời này là một trường value của lớp Ticket, và giả thiết rằng giá trị được khởi tạo bởi hàm khởi tạo và rằng giá trị đó lấy được bằng phương thức getValue.

Constructor and methods	Processing
Constructor	<input type="text" value="D"/>
Method getValue	<input type="text" value="E"/>
Method setOrigin	The same as is defined in the superclass
Method getOrigin	The same as is defined in the superclass
Method adjustValue	No processing is performed (methods proper do not contain any statements).
Method isValid	<input type="text" value="F"/>

#### Nhóm các câu trả lời:

- a) Trả lại giá trị 0 tại mọi điểm
- D** b) Một lượng là quá lớn theo lý thuyết để có thể tiêu trong 24 giờ được đặt là giá trị khởi tạo cho giá trị của value.
- c) Phương thức deduct được gọi, với amount như là một tham số.
- d) Không xử lý nào được thực hiện (các phương thức hợp lý không bao gồm bất kỳ phát biểu nào).
- E** e) Giá vé cơ bản trên tuyến được trả lại tại mọi thời điểm.
- F** f) Giá cao nhất của vé trên tuyến (giá lớn nhất giữa tất cả các loại vé) được đặt là giá trị khởi điểm của value, và thời điểm chứng nhận tiền vé được lưu lại.
- D** g) Điều tương tự được định nghĩa trong lớp trên.
- F** h) true được trả lại chỉ khi thời gian gọi teong vòng 24 giờ kể từ thời điểm chứng nhận tiền vé được lưu lại trong một thực thể. Trong các trường hợp khác, false được trả lại.
- i) true được trả lại chỉ khi value là không nhỏ hơn giá vé cơ bản trên tuyến. Trong các trường hợp khác, false được trả lại.
- j) true được trả lại chỉ khi value là không nhỏ hơn giá vé cao nhất trên tuyến (giá lớn nhất giữa tất cả các loại vé). Trong các trường hợp khác, false được trả lại.

**Q13.**

Read the following description of an assembler program and the program itself, and then answer Subquestions 1 through 3.


*[Program Description]*

This is a subprogram BPSRH in which one word is searched for a specified bit pattern.

1) The main program sets the first address of the parameter area in GR1, and calls BPSRH. The format of the parameter area is as follows:

(GR1)+0	Word to be searched
+1	n
+2	Bit pattern (right-justified)

2) In BPSRH, bits in the word to be searched are compared with the specified bit pattern in descending order of bit number, and the highest bit number in the portion that first matched is set in GR0; BPSRH then returns to the main program. If no portion is matched, -1 is set in GR0, and BPSRH returns to the main program. In the following example, the highest bit number in the portion that matched with the pattern is 12, which is set in GR0.

Comparison 	
	15                      ...                      0 (Bit number)
(GR1)+0	0001101011010110
+1	0000000000000100
+2	0000000000001101

3) When returning from the subprogram, the contents of general-purpose registers GR1 through GR7 are restored to the original values.

*[Program]*

```

1      BPSRH  START
2              RPUSH
3              LD      GR6,1,GR1
4              LAD     GR7,16
5              SUBA   GR7,GR6 ; GR7 <- (16 - n)
6              LD     GR2,2,GR1
7              SLL    GR2,0,GR7 ; With bit pattern left-justified
8              LAD     GR4,-1
9              A      ; Generation of mask pattern
10             LAD     GR0,-1 ; Initialization of return value
11             LAD     GR3,0 ; Initialization of comparison location pointer
12             LD      GR5,0,GR1
13     LOOP    LD      GR6,GR5 ; GR6 is for temporary use
14             AND     GR6,GR4
15             CPL     GR6,GR2 ; Comparison with bit pattern
16             JZE     FIND
17             LAD     GR3,1,GR3 ; Next comparison location is set.
18             CPA     GR3,GR7 ; Does uncomared portion contain n bits or more?
19             JPL     EXIT
20             B
21             JUMP    LOOP
22     FIND    LAD     GR0,15 ; Calculation of bit number
23             SUBA   GR0,GR3

```

**Q13.**

Hãy đọc phần mô tả và nội dung của chương trình hợp ngữ dưới đây, và trả lời các câu hỏi con từ 1 đến 3.

*[Mô tả Chương trình]*

Đây là một chương trình con BPSRH trong đó một từ được rà soát để tìm kiếm một mẫu bit xác định.

1) Chương trình chính tạo lập địa chỉ đầu tiên của vùng tham số trong GR1, và gọi BPSRH. Định dạng của vùng tham số này như sau:

n: Number of bits in bit pattern  
(1 ≤ n ≤ 16)

2) Trong BPSRH, các bit trong từ bị rà soát được so sánh với mẫu bit cụ thể sắp xếp theo thứ tự bit số (bit number) giảm dần, và bit số cao nhất trong đoạn mà phù hợp đầu tiên thì được cài đặt trong GR0; BPSRH tiếp theo sẽ quay lại chương trình chính. Nếu không đoạn nào phù hợp, -1 được cài đặt trong GR0, và BPSRH sẽ quay lại chương trình chính. Trong ví dụ sau đây, bit số cao nhất trong đoạn mà phù hợp với mẫu là 12, và nó được cài đặt trong GR0.

3) Khi quay lại từ chương trình con, các nội dung của các thanh ghi đa năng (general-purpose) GR1 đến GR7 được khôi phục các giá trị ban đầu.

*[Chương trình]*

```

24      EXIT   RPOP
25      RET
26      END

```

### Subquestion 1

From the answer groups below, select the correct answers to be inserted into the blanks in the above program.

Answer group for **A**:

- a) SLL GR4, 0, GR6
- b) SLL GR4, 0, GR7
- c) SRA GR4, 0, GR6
- d) SRA GR4, 0, GR7
- e) SRL GR4, 0, GR6
- f) SRL GR4, 0, GR7

Answer group for **B**:

- a) LD GR5, 1, GR2
- b) SLL GR3, 1
- c) SLL GR5, 1
- d) SRL GR3, 1
- e) SRL GR5, 0, GR2
- f) SRL GR5, 0, GR3
- g) SRL GR5, 1

### Subquestion 2

From the answer group below, select the correct answer as the hexadecimal notation of the value of GR5 when the following parameters are passed and control is just transferred to the instruction labeled FIND.

(GR1)+0	0001101011010110
+1	00000000000000100
+2	00000000000001101

Answer group:

- a) 000D      b) 1AD6      c) AD60
- d) D000      e) D6B0

### Subquestion 3

A subprogram RESERVE was created in which movie theater seats are reserved using a subprogram BP1SRH for performing searches specializing in bit patterns comprising consecutive 1's.

From the answer groups below, select the correct answers to be inserted into the blanks in RESERVE.

1) The number of seats for reservation in the movie theater is 1,024. Seat numbers are 0 through 1023. Moreover, seats for reservation are divided in such a way that every 16 seats with consecutive seat numbers constitute a group. A table for seat control consists of 64 consecutive words. Bit number 15 of the first word represents the state of seat number 0, and bit number 0 of the last word represents the state of seat number 1023. When the cor-

### Câu hỏi con 1

Từ các nhóm câu trả lời dưới đây, lựa chọn các câu trả lời đúng để điền vào các ô trống trong chương trình trên.

Nhóm câu trả lời cho **A**:

- a) SLL GR4, 0, GR6
- b) SLL GR4, 0, GR7
- c) SRA GR4, 0, GR6
- d) SRA GR4, 0, GR7
- e) SRL GR4, 0, GR6
- f) SRL GR4, 0, GR7

Nhóm câu trả lời cho **B**:

- a) LD GR5, 1, GR2
- b) SLL GR3, 1
- c) SLL GR5, 1
- d) SRL GR3, 1
- e) SRL GR5, 0, GR2
- f) SRL GR5, 0, GR3
- g) SRL GR5, 1

### Câu hỏi con 2

Từ nhóm câu trả lời dưới đây, lựa chọn câu trả lời đúng dưới dạng ký hiệu giá trị hệ thập lục phân (hexadecimal) của GR5 khi các tham số sau đây được truyền và điều khiển vừa được truyền tới chỉ lệnh có nhãn là FIND.

Nhóm trả lời:

- a) 000D      b) 1AD6      c) AD60
- d) D000      e) D6B0

### Câu hỏi con 3

Một chương trình con RESERVE được tạo cho việc đặt chỗ ngồi trong rạp chiếu phim. Nó sử dụng một chương trình con BP1SRH để thực hiện các tìm kiếm chuyên về các mẫu bit gồm các số 1 liên tiếp nhau.

Từ các nhóm câu trả lời dưới đây, lựa chọn các câu trả lời đúng để điền vào các ô trống trong RESERVE.

1) Số ghế đặt chỗ trong rạp chiếu phim là 1,024. Các số ghế là từ 0 đến 1023. Hơn nữa, các ghế đặt chỗ được phân chia theo cách mà cứ 16 ghế có số ghế liên tục thì tạo thành một nhóm. Một bảng kiểm soát chỗ gồm có 64 từ (word) liên tiếp. Bit số 15 của từ đầu tiên thể hiện trạng thái của ghế số 0, và bit số 0 của từ cuối cùng thể hiện trạng thái ghế số 1023. Khi bit tương ứng trong bảng kiểm soát chỗ là 1, chỗ ngồi đó là còn trống. Nếu



responding bit in the table for seat control is 1, the relevant seat is vacant. If the bit is 0, the relevant seat is booked.

bit đó là 0, chỗ ngồi đó đã được đặt.

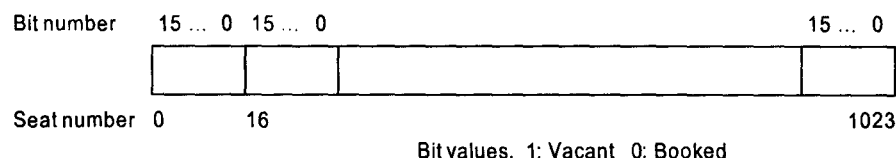


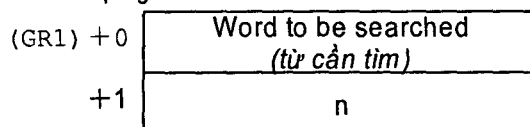
Fig. Format of table for seat control (64 words)  
Mẫu bảng kiểm soát chỗ (64 từ)

2) In the main program, the number of seats to be booked,  $n$  ( $1 \leq n \leq 16$ ), is set in GR1, the beginning address of the table for controlling seats for reservation is set in GR2, and RESERVE is called.

2) Trong chương trình chính, số chỗ được đặt,  $n$  ( $1 \leq n \leq 16$ ), được cài đặt trong GR1, địa chỉ bắt đầu của bảng kiểm soát đặt chỗ được cài đặt trong GR2, và RESERVE được gọi.

3) RESERVE secures a specified number of seats in such a way that all relevant seats are consecutive and that no group is skipped in between. Vacant seats are searched for in sequence starting with seat number 0. If consecutive vacant seats are found, these seats are placed in the "booked" state, the lowest seat number secured is set in GR0, and control is returned to the main program. If no vacant seat is secured, -1 is set in GR0, and control is returned to the main program.

3) RESERVE thắt chặt số lượng ghế theo cách mà tất cả các chỗ liên quan là liên tục và rằng không một nhóm nào bị bỏ qua giữa chúng. Các chỗ trống được tìm kiếm tuần tự bắt đầu từ ghế số 0. Nếu các ghế trống liên tục được tìm thấy, những ghế này được đặt trong trạng thái "được đặt", ghế số thấp nhất tìm thấy được cài đặt trong GR0, và kiểm soát được trả lại chương trình chính.



$n$ : Number of bits comprising consecutive 1's  
(số bit liên tục bằng 1)  
( $1 \leq n \leq 16$ )

4) When returning from the subprogram RESERVE, the contents of general-purpose registers GR1 through GR7 are restored to the original values.

4) Khi quay lại từ chương trình con RESERVE, các nội dung của các thanh ghi đa năng GR1 đến GR7 được khôi phục giá trị ban đầu.

5) The format of the parameters given to BP1SRH are as follows.

5) Định dạng các tham số được đưa cho BP1SRH như sau:

BP1SRH is a program in which line numbers 6 through 9 of BPSRH are replaced with the following three sets of instructions.

BP1SRH là một chương trình trong đó dòng số 6 đến 9 của BPSRH được thay thế với ba tập chỉ lệnh sau đây.

```
LAD GR2, #8000
SRA GR2, -1, GR6
LD GR4, GR2
```

```
LAD GR2, #8000
SRA GR2, -1, GR6
LD GR4, GR2
```

[Program]

[Chương trình]

```
RESERVE START
  RPUSH
  LD GR6, GR1 ; Save n
  LAD GR1, PARAM
  ST GR6, 1, GR1 ; Preparation of parameters for calling BP1SRH (1)
  ST GR2, TBLADD ; Save beginning address of table for seat control
  LAD GR4, 64, GR2
  LAD GR0, -1 ; Initialization of return value
LOOP CPL GR2, GR4 ; End of search?
  JZE EXIT
  LD GR5, 0, GR2 ; Load one word from table for seat control
  ST GR5, 0, GR1 ; Preparation of parameters for calling BP1SRH (2)
  CALL BP1SRH ; Searching for vacant seats in one word
  CPA GR0, --1
  JNZ FIND
  LAD GR2, 1, GR2 ; Going to search next word
  JUMP LOOP
FIND LAD GR3, 15
  SUBA GR3, GR0 ; GR3 <- (15 - GR0)
```

```

LAD    GR7,#8000
SRA    GR7,-1,GR6
C
XOR    GR7,=#FFFF ; GR7 <- 1110000111111111 (In case GR0 = 12, n = 4)
AND    GR5,GR7 ; Set at booked state.
ST     GR5,0,GR2
SUBL   GR2,TBLADD ; Calculation of seat number
D
ADDA   GR2,GR3
LD     GR0,GR2
EXIT   RPOP
RET
TBLADD DS    1
PARAM  DS    2 ; Parameter areas for calling BP1SRH
END

```

Answer group for **C**:

- a) AND GR5,GR3
- b) AND GR5,GR7
- c) OR GR5,GR3
- d) OR GR5,GR7
- e) SRA GR7,0,GR3
- f) SRL GR7,0,GR3

Nhóm câu trả lời cho **C**:

- a) AND GR5,GR3
- b) AND GR5,GR7
- c) OR GR5,GR3
- d) OR GR5,GR7
- e) SRA GR7,0,GR3
- f) SRL GR7,0,GR3

Answer group for **D**:

- a) SLL GR2,1
- b) SLL GR2,2
- c) SLL GR2,4
- d) SLL GR3,1
- e) SLL GR3,2
- f) SLL GR3,4

Nhóm câu trả lời cho **D**:

- a) SLL GR2,1
- b) SLL GR2,2
- c) SLL GR2,4
- d) SLL GR3,1
- e) SLL GR3,2
- f) SLL GR3,4