СОФИЙСКИ УНИВЕРСИТЕТ
„СВ. КЛИМЕНТ ОХРИДСКИ“

ФАКУЛТЕТ ПО МАТЕМАТИКА
И ИНФОРМАТИКА

ДЪРЖАВЕН ИЗПИТ
ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР ПО ИНФОРМАЦИОННИ СИСТЕМИ”

ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)
09.07.2019 г.

Драги абсолвенти:

- Попълнете факултетния си номер в горния десен ъгъл на всички листове.
- Пишете само на предоставените листове, без да ги разкопчавате.
- **Решението на една задача трябва да бъде на същия лист, на който е и нейното условие (т.е. може да пишете отпред и отзад на листа със задачата, но не и на лист на друга задача).**
- Ако имате нужда от допълнителен лист, можете да поискате от квесторите.
- На един лист не може да има едновременно и чернова, и белова.
- Черновите трябва да се маркират, като най-отгоре на листа напишете "ЧЕРНОВА".
- Ако решението на една задача не се побира на нейния лист, трябва да поискате нов бял лист от квесторите. Той трябва да се защипе с телбод към листа със задачата.
- Всеки от допълнителните листове (белова или чернова) трябва да се надпише най-отгоре с вашия факултетен номер.
- Черновите също се предават и се защипват в края на работата.
- Времето за работа по изпита е 3 часа.

Изпитната комисия ви пожелава успешна работа!

Задача 1. *Задачата да се реши на езика C++.*

Даден е двумерен масив от символи с размер 6 на 6 — малки и главни латински букви и цифри. Две клетки в него ще наричаме "съседни", ако имат обща стена (т.е. всяка клетка е съседна с най-много четири други, намиращи се под, над, вляво и вдясно от нея). Път с дължина N ще наричаме редица $a_0, a_1, \ldots a_{N-1}$ от клетки, за която:

1. **за всяко** `0 ≤ i < N − 1` **е изпълнено, че** a_i и a_{i+1} са съседни;
2. никоя от клетките не се среща повече от веднъж (т.е. няма цикли).

Да се попълнят празните места в кода на дадените по-долу функция `contains` и помощната ѝ функция `walk`. Функцията `contains` получава два аргумента – масив `arr` от дадения тип `char[6][6]` и символен низ `str`. Тя трябва да връща истина тогава и само тогава, когато в `arr` съществува път, чиито клетки образуват точно съдържанието на низа `str` (вижте примера по-долу). За определеност считаме, че функцията трябва да връща истина за празния низ.

Пример: За дадения по-долу двумерен масив `contains` трябва да върне истина, ако ѝ бъдат подадени низовете `"abcdefgh"`, `"A123B123C"` или `""`. За улеснение, за да може да ги видите по-лесно, те са маркирани в сиво.

y	u	f	a	b	c
G	o	p	g	B	1
c	b	a	h	3	2
d	k	j	i	2	3
e	f	Q	N	1	C
h	g	h	M	A	r

Кодът на двете функции е даден на следващия лист:

```
bool contains(char arr[6][6], const char* str)
{
```

```
    for (int row = 0; row < _____; _____________)
        for (int col = 0; col < _____; _____________)
            if (walk(arr, row, col, str))
                return _________;
    return _________;
}

bool walk(char arr[6][6], int row, int col, const char* str)
{
    if (*str == '\0')
        return ____________;

    if (row < 0 || col < 0 || row >= 6 || col >= 6)
        return ____________;

    if (arr[row][col] != *str)
        return ____________;

    arr[row][col] *= -1;

    bool result =
        walk(arr, row + ___, col,       str + 1) ||
        walk(arr, _________, _________, str + 1) ||
        walk(arr, _________, _________, str + 1) ||
        walk(arr, _________, _________, str + 1);

    arr[row][col] _________________;

    return result;
}
```

Задача 2. *Задачата да се реши на езика C++.*

А) Да се допълни кодът на класа `Base` така, че той да бъде синглетон (в даден момент от времето трябва да може да съществува най-много един негов обект). Местата, на които трябва да се попълни код, са обозначени с подчертаване. Функцията `main()` демонстрира работата с класа.

```
class Base
{
_________:
   ________ _______& getInstance()
   {
      static ________ instance;
      return instance;
   }


_________:
   Base()
   { }

   Base(const Base &);


   Base& operator=(const Base &);
};

int main()
{
   Base& b1 = Base::getInstance();
   Base& b2 = Base::getInstance();

   // prints 1
   std::cout << (&b1 == &b2);
   return 0;
}
```

Б) Нека класът `Derived` е дефиниран по следния начин:

```
class Derived : public Base
{
};
```

Да се определи дали в горната дефиниция има грешка.

Ако грешка има, да се обясни каква е тя.

Ако грешка няма, да се определи дали класът `Derived` също ще бъде синглетон и да се обясни защо.

Верен отговор без коректно обяснение не се оценява.

Задача 3. *Задачата да се реши на един от езиците Scheme или Haskell. По-долу оградете името на езика, който сте избрали за вашето решение.*

Големият онлайн магазин *Siberia* търси начин да увеличи продажбите като препоръчва на клиентите си подходящи продукти. За целта изследователският екип на *Siberia* експериментира с различни реализации на функция `bestFit`, която приема като параметър код на продукт `a` и връща код на друг продукт `b`, който клиентите на магазина най-вероятно биха си купили заедно с `a`. Задачата пред разработчиците на *Siberia* е да реализират функция `recommended`, която получава като параметри потребителска кошница `basket` (списък от целочислени кодове на продукти), функция `bestFit` и списък от продуктите на магазина `products` (списък от наредени двойки от уникален код на продукт и цена — неотрицателно число).

Да се попълнят по подходящ начин празните полета по-долу така, че функцията `recommended` да връща списък от кодовете на всички възможни препоръчани продукти. Допуска се в резултата някои кодове да се срещат повече от веднъж. Препоръчан продукт е такъв, който:

- все още не е в `basket`, но се получава като резултат от прилагането на функцията `bestFit` над някой от продуктите, които вече са в `basket`;
- има цена, която не надвишава общата цена на потребителската кошница, дефинирана като сумата от цените на продуктите в `basket`.

Помощните дефиниции `findPrice` и `basketCost` намират съответно цената на даден продукт `product` в списъка `products` и цената на потребителската кошница. Да се приеме, че `basket` съдържа само кодове на продукти в `products` и `bestFit` също връща само такива кодове.

Упътване: могат да се използват наготово функциите `apply`, `assoc`, `elem`, `filter`, `foldr`, `lookup`, `map`, `member`, `sum` и стандартните функции в R^{5}RS за Scheme и в Prelude за Haskell.

Scheme

```
(define (recommended basket bestFit products)
  (define (findPrice product)
    ____________________________________________________ products ____
  (define basketCost
    ____________________________________________________ basket ____

  (_____________________
    (lambda (product)
      ____________________________________________________________________)
        (_____________________________________________________ basket)))
```

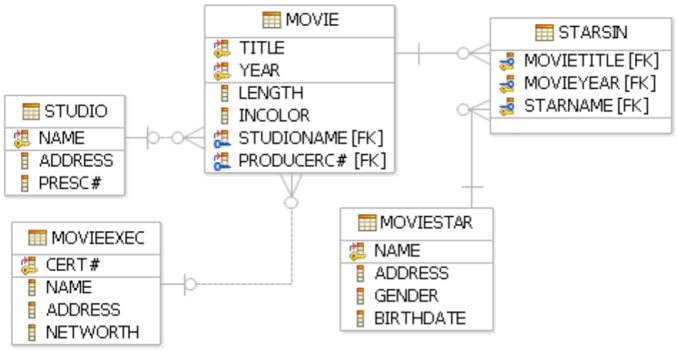
Haskell

```
recommended basket bestFit products =

  ____________________
  (\product ->
     ____________________________________________________________________)
  (_____________________________________________________________ basket)

  where findPrice product =
          ____________________________________ products _________________
        basketCost =
          ____________________________________________________ basket ________
```

Задача 4. Дадена е базата от данни **Movies**, в която се съхранява информация за филми, филмови студия, които ги произвеждат, продуцентите на филмите, както и актьорите, които участват в тях.



Таблицата **Movie** съдържа информация за филми. Атрибутите `title` и `year` заедно формират първичния ключ.

- `title` – заглавие;
- `year` – година, в която е заснет филмът;
- `length` – дължина в минути;
- `incolor` – `'Y'` за цветен филм и `'N'` за чернобял;
- `studioname` – име на студио, външен ключ към `Studio.name`;
- `producerc#` – номер на сертификат на продуцента, външен ключ към `MovieExec.cert#`.

Таблицата **StarsIn** съдържа информация за участието на филмовите звезди във филмите. Трите атрибута заедно формират първичния ключ. Атрибутите `movietitle` и `movieyear` образуват външен ключ към `Movie`.

- `movietitle` – заглавие на филма;
- `movieyear` – година на заснемане на филма;
- `starname` – име на филмовата звезда, външен ключ към `MovieStar.name`.

Таблицата **MovieStar** съдържа информация за филмови звезди:

- `name` – име, първичен ключ;
- `address` – адрес;
- `gender` – пол, `'M'` за мъж (актьор) и `'F'` за жена (актриса);
- `birthdate` – рождена дата.

Таблицата **Studio** съдържа информация за филмови студиа:

- `name` – име, първичен ключ;
- `address` – адрес;
- `presc#` – номер на сертификат на президента на студиото.

Таблицата **MovieExec** съдържа информация за продуцентите на филми.

- `cert#` – номер на сертификат, първичен ключ;
- `name` – име;
- `address` – адрес;
- `networth` – нетни активи.

*Забележка за всички таблици: Всички атрибути, които не участват във формирането на първичен ключ, могат да приемат стойност **NULL**.*

а) Да се огради буквата на заявката, която извежда име на студио и броя на филмите му, за тези студия с по-малко от два филма. Студиата, които нямат нито един филм, **НЕ** трябва да присъстват в резултата.

A)
```
SELECT S.NAME, COUNT(M.TITLE) as CNT
FROM STUDIO S JOIN MOVIE M
ON S.NAME = M.STUDIONAME
GROUP BY S.NAME
HAVING CNT < 2;
```

B)
```
SELECT S.NAME, COUNT(M.TITLE) as CNT
FROM STUDIO S LEFT JOIN MOVIE M
ON S.NAME = M.STUDIONAME
WHERE M.TITLE IS NULL
GROUP BY S.NAME
HAVING COUNT(M.TITLE) < 2;
```

C)
```
SELECT S.NAME, COUNT(M.TITLE) as CNT
FROM STUDIO S JOIN MOVIE M
ON S.NAME = M.STUDIONAME
GROUP BY S.NAME
HAVING COUNT(M.TITLE) < 2;
```

D)
```
SELECT S.NAME, COUNT(M.TITLE) as CNT
FROM STUDIO S JOIN MOVIE M
ON S.NAME = M.STUDIONAME
WHERE COUNT(M.TITLE) < 2
GROUP BY S.NAME;
```

б) Да се напише заявка, която да изведе имената на всички продуценти с минимален нетен актив.

Задача 5. Информационна система съхранява информация за клиентите на агенция, която отдава жилища под наем. Съхранява се информация за клиента: номер на клиент – цяло число, уникално за всеки клиент на агенцията, име на клиент – низ до 50 символа и номер на банкова карта за клиента – низ от точно 16 символа. Съхранява се информация за имота: номер на имот – цяло число уникално за всеки имот, адрес на имота – низ до 100 символа, описание на имота – низ до 50 символа, наем за имота – цяло число по-голямо от 0 и състояние на имота – цяло число, може да бъде само две стойности 0 – за свободен или 1 – за нает. Всеки имот принадлежи на точно един собственик, а един собственик може да притежава много имоти. В системата се пази информация за номер на собственик – цяло число, име на собственик – низ до 50 символа, IBAN на собственика – низ от точно 22 символа. В сила са следните ограничения: един клиент може да наема много имоти и един имот може да бъде наеман от много клиенти в течение на времето. В базата от данни трябва да се пази и информация за началната и крайната дата (периода) на наемане на имота от клиента.

Задание:
а) Да се направи E/R модел на БД, която съхранява гореописаната информация. Да се начертае E/R диаграма на модела.
б) Да се преобразува E/R диаграмата към релационни схеми. Да се премахнат излишествата, където това е възможно.
в) Да се напише DDL код, съответстващ на релационните схеми. Да се реализират всички описани ограничения.

Задача 6. а) В контекста на ***информационната система за отдаване на жилища под наем***, представена в Задача 5, разглеждаме процес по наемане на имот от клиент на агенцията. Имотът може да бъде нает за минимум един месец. Наемането се счита за успешно, ако е заплатен поне един месечен наем.

Да се опише в пълен формат ***потребителски случай „Наемане на имот“***. В описанието да се включат минимум: актьори, които участват; предусловия; следусловия (резултати); основен успешен сценарий; алтернативен сценарий; неуспешен сценарий. При описанието да се посочат също: взаимодействие с други потребителски случаи (ако е приложимо, но само с препратка); какви проверки за коректност се извършват при обработката. Да се посочат допълнителни нефункционални изисквания към този потребителски случай.

Ако са направени някакви допускания във връзка с представения сценарий, те трябва да бъдат описани явно.

б) В контекста на ***информационната система за отдаване на жилища под наем***, представена в Задача 5, и описанието на потребителския случай, изготвено в подусловие а), да се подготви ***диаграма на дейността (activity diagram)*** в термините на UML стандарта. Да се опише значението на елементите от нотацията, които са използвани в диаграмата.

Задача 7. Да се пресметне интегралът:

$$\int_1^8 \frac{dx}{\sqrt[3]{x}+x}$$

Чернова