СОФИЙСКИ УНИВЕРСИТЕТ
„СВ. КЛИМЕНТ ОХРИДСКИ“

ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

ДЪРЖАВЕН ИЗПИТ

ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР ПО ИНФОРМАЦИОННИ СИСТЕМИ”

ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)
10.09.2015 г.

Моля, не пишете в тази таблица!			
Зад. 1		Зад. 5	
Зад. 2		Зад. 6	
Зад. 3		Зад. 7	
Зад. 4		Зад. 8	
Крайна оценка:			

Драги абсолвенти:

- Попълнете факултетния си номер в горния десен ъгъл на всички листа;
- Пишете само на предоставените листове без да ги разкопчавате;
- Ако имате нужда от допълнителен лист, можете да поискате от квесторите;
- Допълнителните листа трябва да се номерират, като номерата продължават тези от настоящия комплект;
- Всеки от допълнителните листа трябва да се надпише най-отгоре с вашия факултетен номер;
- **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 решението на задачата не се побира в един лист, трябва да поискате нов бял лист от квесторите. В такъв случай отново трябва да започнете своето решение на листа с условието на задачата и в края му да напишете „Продължава на лист № X”, където X е номерът на допълнителния лист, на който е вашето решение;
- Черновите трябва да бъдат маркирани, като най-отгоре на листа напишете „ЧЕРНОВА“;
- На един лист не може да има едновременно и чернова и белова;
- Времето за работа по изпита е 3 часа;

Изпитната комисия ви пожелава успешна работа!

Задача 1 (10 точки).

А) Следните програмни фрагменти са съответно от булева функция на C++ и статичен булев метод на Java, проверяващи дали в даден масив `a` от цели числа, подредени в нарастващ ред, се съдържа числото `x`. Функцията/методът прилагат алгоритъма за двоично търсене. Липсващите части от фрагментите са обозначени с _____. Попълнете липсващите части така, че функцията или съответно методът да са коректно дефинирани спрямо това описание. *Решете задачата за един от двата езика по избор!*

(C++)
```
bool member (int x, int a[], int size){
   if (size == 0) return false;
   return a[size/2] == x  ____
      (a[size/2] < x && member (_______________________)) ___
      (_____________________________________________);
}
```

(Java)
```
static boolean member (int x, int[] a){
   if (a.length == 0) return false;
   return a[a.length/2] == x ___
      (a[a.length/2] < x && member (_______________________)) ___
      (_____________________________________________);
}
```

Б) (C++ и Java) Нека е дефиниран масив `nums`, в който се съдържат `N` целочислени стойности. Попълнете празните полета, за да бъде коректна програмната реализация на алгоритъма за сортиране във възходящ ред чрез пряка селекция (selection sort).

```
for (int i = 0; i < _________; i++) {
   int min = i;
   for (int j = ________; j < ________; j++) {
      if(nums[____] < nums[____]) _____ = ____;
   }
   if(min != _________) {
      int x = nums[i];
      nums[____] = _________;
      _____________________;
   }
}
```

В) Каква ще бъде стойността на променливата `result` след изпълнение на следния програмен фрагмент на C++/Java:

```
int a = 0; int b = 15; int result = -1;
if (b < 10 && b / a < 10) result = 0;
else  result = 1;
```

а) -1 б) 0 в) 1 г) грешка при компилация делене на нула д) грешка при изпълнение делене на нула

Г) Каква е стойността на израза `6 | 11` на езиците C++ и Java:

а) 15 б) 13 в) 10 г) 11

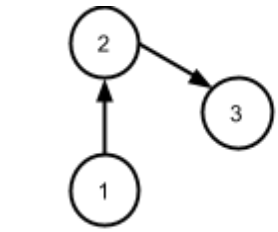
Задача 2 (10 точки). Следната задача да се реши на един от езиците за програмиране C++ или Java. Да се обозначи явно на кой от двата езика е решавана задачата. При решението на задачата да не се използват библиотеки за работа със структури от данни.

а) Да се дефинира подходяща структура от данни, позволяваща представянето в паметта на програмата на ориентиран граф от типа `G=<V,E>`, където `V` е множеството на целите положителни числа, не по-големи от `1000` (представено чрез типа данни int), а `E=V×V`.

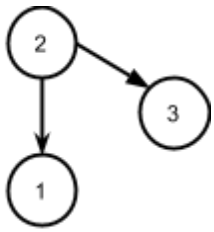
б) За така дефинираната структура от данни да се дефинира функция (или статичен метод)

```
[булев тип] isConnected ([подходящ тип] g)
```

чиято стойност е истина точно за тези графи `g`, които са свързани. За един граф `G=<V,E>` казваме, че е свързан, ако за всяка двойка `u∈V, v∈V` има път от `u` до `v` или от `v` до `u`. На примера са показани два графа, първият от които е свързан, а вторият – не.



свързан ориентиран граф



несвързан ориентиран граф

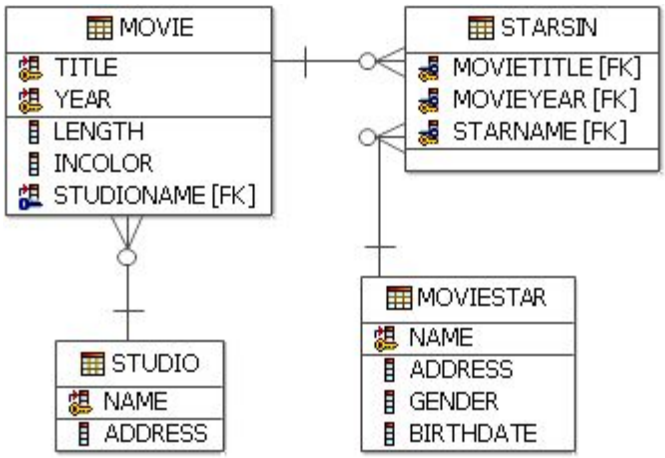
Забележка: При избор на Java за език за програмиране е достатъчно да се дефинира статичен метод, който решава задачата.

Задача 3 (10 точки). Дадени са следните дефиниции на програмните езици Haskell и Scheme, от програмния код на които липсват части. Попълнете полетата, обозначени с _____ с необходимия програмен код така, че да се получат посочените желани оценки. Изберете само един от двата езика за решението на задачата и напишете името му в даденото за целта поле. Точки за задачата се дават само за избрания от вас език.

Избран език:

	Език Haskell:
1	**merge x [] = ____** **merge ___ ___ = ___** **merge (x:xs) (y:ys) = if x < y** **then _____** **else _____** израз: **merge [1,3,5,7] [2,2,6,10]** желана оценка: **[1,2,2,3,5,6,7,10]**
2	израз: **(____->[y|y<-_____,even ____])[1,2,3,4]** желана оценка: **[2,4]**
	Език Scheme:
1	**(define (merge l1 l2)** **(cond ((null? l1) ___)** **(_____)** **((< (car l1) (car l2)) _____)** **(else _____)))** израз: **(merge '(1 3 5 7) '(2 2 6 10))** желана оценка: **(1 2 2 3 5 6 7 10)**
2	израз: **((lambda (___) (filter _____)) '(1 2 3 4))** желана оценка: **(2,4)**

Задача 4 (10 точки). Дадена е базата от данни Movies.



Таблицата ***Studio*** съдържа информация за филмови студиа:
name – име, първичен ключ;
address – адрес.

Таблицата ***Movie*** съдържа информация за филми. Колоните *title* и *year* заедно формират първичния ключ.
title – заглавие;
year – година, в която филмът е заснет;
length – дължина в минути;
incolor – **'Y'** за цветен филм и **'N'** за черно-бял;
studioname – име на студио, външен ключ.

Таблицата ***MovieStar*** съдържа информация за филмови звезди:
name – име;
address – адрес;
gender – пол, **'M'** за мъж и **'F'** за жена;
birthdate – рождена дата.

Таблицата ***StarsIn*** съдържа информация за участието на филмовите звезди във филмите. Трите колони заедно формират първичния ключ. Колоните *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.

Условие 1. Да се посочи заявката, която извежда имената и адресите на всички актриси от София, както и на всички филмови студиа от София. Резултатите да са сортирани по адрес.

а)
```
SELECT MS.name, MS.address
FROM MovieStar AS MS
JOIN Studio S ON MS.address =
S.address
WHERE MS.address LIKE '%Sofia%'
AND gender = 'F'
ORDER BY MS.address;
```

б)
```
SELECT *
FROM (SELECT name, address
      FROM MovieStar
      WHERE gender = 'F'
      UNION
      SELECT name, address
      FROM Studio) T
WHERE T.address LIKE '%Sofia%'
ORDER BY T.address;
```

в)
```
SELECT name, address
FROM MovieStar MS
WHERE gender = 'F'
ORDER BY address
UNION ALL
SELECT name, address
FROM Studio
WHERE address LIKE '%Sofia%'
ORDER BY address;
```

г)
```
SELECT DISTINCT name, address
FROM MovieStar INTERSECT Studio
WHERE address IS NOT NULL
AND gender LIKE 'F'
GROUP BY address
HAVING address LIKE '%Sofia%';
```

д)
```
SELECT DISTINCT name, address
FROM MovieStar
FULL JOIN Studio ON address LIKE
'%Sofia%'
WHERE gender LIKE 'F'
ORDER BY address;
```

Условие 2. Да се посочи заявката, която за всяко студио с най-много три черно-бели филма извежда името му, адреса и средната дължина на филмите (без значение дали са цветни) на това студио. Студиа без филми също да се извеждат.

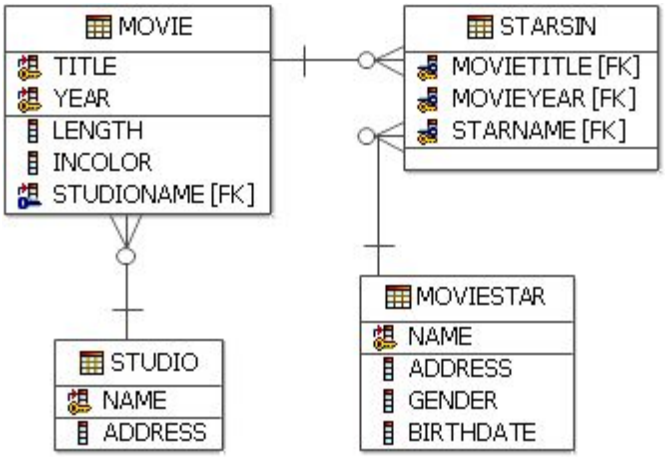
```
а)SELECT name, address, AVG(length) AS avgLength
FROM Studio
LEFT JOIN Movie ON name = studioName
GROUP BY studioName, address
HAVING COUNT(inColor = 'y') <= 3;

б)SELECT DISTINCT name, address, avgLength
FROM Studio, (SELECT studioName, AVG(length) AS avgLength
                  FROM Movie
                  GROUP BY studioName) Averages
WHERE NAME = ANY (SELECT studioName
             FROM Movie
                  WHERE inColor = 'n'
                  GROUP BY studioName
             HAVING COUNT(title) <= 3);

в)SELECT Studio.name, Studio.address, AVG(Movie.length) AS avgLength
FROM Movie
RIGHT JOIN Studio ON studioName = name
GROUP BY name, address
HAVING (SELECT COUNT(*) FROM Movie WHERE inColor = 'n') <= 3;

г)SELECT name, address, AVG(length) AS avgLength
FROM Studio
LEFT JOIN Movie ON name = studioName
WHERE NAME NOT IN (SELECT studioName
                         FROM Movie
                         WHERE inColor = 'n'
                         GROUP BY studioName
                   HAVING COUNT(*) > 3)
GROUP BY name, address;
```

Задача 5 (10 точки). Дадена е базата от данни Movies.



Информационна система за сервиз за битова техника съхранява информация за техници и ремонтите, извършени от тях.

За всеки ремонтиран от сервиза уред се съхранява задължително следната информация: уникален идентификатор, категория (напр. пералня, телевизор и т.н.), производител (до 50 символа), модел (до 20 символа), име на клиент, а ако е известна - и годината на производство (цяло число, по-голямо от 1900).

За всеки техник се съхранява задължително следната информация: уникален идентификатор, име (до 100 символа), ЕГН, както и категории уреди, които може да ремонтира.

Един техник може да извършва много ремонти на уреди. Един уред може да бъде поправян много пъти от различен техник. За всяка поправка се пази дата и цена. За дадена категория уреди може да има няколко техници в сервиза.

а) Създайте E/R модел на БД, която съхранява гореописаната информация.

б) Преобразувайте E/R диаграмата към релационни схеми. Премахнете излишествата, където това е възможно.

в) Напишете DDL код, съответстващ на релационните схеми. Реализирайте всички описани ограничения.

Задача 6 (10 точки). Представете си система за поръчване на такси, подобна на Uber. Системата трябва да позволява на клиента:

- да си поръча такси;
- да получи известяване, когато таксито пристигне на неговия адрес;
- да заплати за превоза през мобилния си телефон;
- да даде отзив за качеството на услугата и шофьора;
- да пази история на всички пътувания на клиента, заедно с информация, като например техния маршрут, дължина, време на превоза, крайна цена, шофьор и т.н.

Представете модел на потребителските случаи за тази система и го опишете текстово. Включете поне 7 потребителски случая с евентуални връзки между тях. Дефинирайте актьорите и обяснете техните роли, посочете границите на системата, евентуална връзка с външни системи.

Задача 7 (10 точки). Разгледайте отново условието на задача 5 (система за поръчване на такси). След това решете следните задачи:

А) Направете пълно описание на потребителски случай (use case), който описва поръчване на такси, пътуване до крайната дестинация и заплащане. Случаят да има поне две разширения.

Б) Нарисувайте UML диаграма на последователност (sequence), която описва стъпките в потребителския случай от точка А). Диаграмата да включва основния сценарий на случая и да съдържа поне един алтернативен блок (alt), в който се представя едно от разширенията му.

В) Опишете нефункционални изисквания към потребителския случай. Те трябва да бъдат организирани по FURPS, като за всяка точка от FURPS трябва да има поне едно изискване.

Задача 8 (10 точки). Намерете неопределения интеграл

$$\int \frac{x-2}{x(x^2+2)}\,dx.$$

ЧЕРНОВА

ЧЕРНОВА

ЧЕРНОВА