

СОФИЙСКИ УНИВЕРСИТЕТ
„СВ. КЛИМЕНТ ОХРИДСКИ“



ФАКУЛТЕТ ПО МАТЕМАТИКА
И ИНФОРМАТИКА

ДЪРЖАВЕН ИЗПИТ
ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР ПО ИНФОРМАТИКА”

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

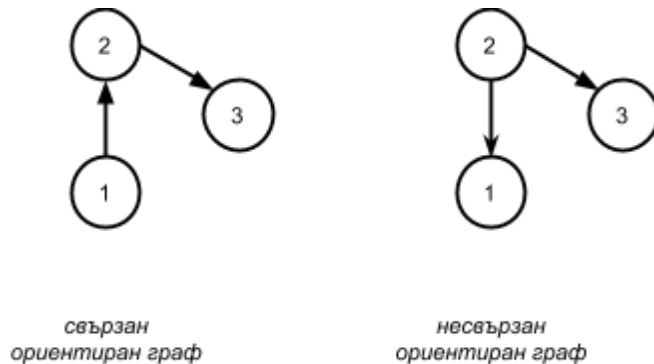
Задача 2 (10 точки). Следната задача да се реши на един от езиците за програмиране C++ или Java. Да се обозначи явно на кой от двата езика е решавана задачата. При решението на задачата да не се използват библиотеки за работа със структури от данни.

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

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 $G = \langle V, E \rangle$ казваме, че е свързан, ако за всяка двойка $u \in V, v \in V$ 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.



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 изборения от вас език.

Избран език:

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

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.

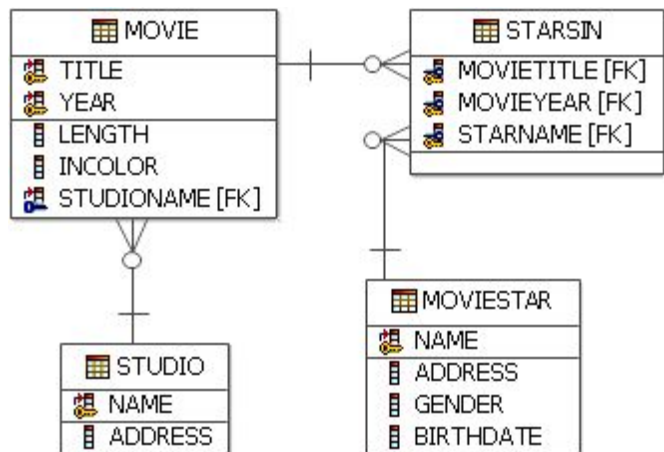
title – заглавие;

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;
```

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
FULL JOIN Studio ON address LIKE
'%Sofia%'
WHERE gender LIKE 'F'
ORDER BY 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;
```

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%';
```

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.

```
a) 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). Текстов файл с име `comprocl` съдържа зададената по-долу последователност от команди на `bash` за Linux. Напишете вдясно какво ще бъде изведено на стандартния изход след стартиране на файла с команден ред

```
bash comprocl 12 34 56
```

ако на стандартния вход бъде подадена последователността от символи 5 6

```
var=1
for i in 4 3 2 1
do for j
    do if test $i -gt $#
        then var=`expr $var \* $i`
            echo $var $j >> ff
        else continue
        fi
    done
done

while true
do echo $*
    break
done

read k1 k2

while cat ff | grep $k2

do set $k1 $var
    shift
    echo $2
    grep $i ff
    exit
    echo $1
done

wc -l < ff

echo END
```


Задача 6 (10 точки). Даден е крайният автомат

$$A = \langle \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}, \{a, b\}, q_0, \delta, \{q_6\} \rangle$$

със следната функция на преходите:

$\delta :$

q	a	b
q_0	$\{q_1, q_3, q_6\}$	$\{q_4, q_6\}$
q_1	\emptyset	$\{q_2\}$
q_2	$\{q_5, q_6\}$	$\{q_2\}$
q_3	$\{q_5, q_6\}$	\emptyset
q_4	\emptyset	$\{q_4, q_6\}$
q_5	\emptyset	$\{q_5, q_6\}$
q_6	\emptyset	\emptyset

Да се построи минимален детерминиран краен автомат, еквивалентен на дадения.

Задача 7 (10 точки). Нека $G = (V, E)$ е дърво. Говорейки за *път* в G имаме предвид прост път: такъв без повтаряне на върхове. Ако p е път в G , то $|p|$ означава дължината на p . Нека X е множеството от пътищата в G . Нека $Y \subseteq X$ е подмножеството на X , което се дефинира по следния начин.

За всеки път $p_1 \in Y$ и за всеки път $p_2 \in X$: $|p_1| \geq |p_2|$.

Докажете или опровергайте следните твърдения:

(а – 5 точки) За всеки два пътя $p, q \in Y$ съществува връх $u \in V$, такъв че u е общ за p и q .

(б – 5 точки) Съществува връх $u \in V$, такъв че за всеки два пътя $p, q \in Y$, u е общ за p и q .

Задача 8 (10 точки). Намерете неопределения интеграл

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

ЧЕРНОВА

ЧЕРНОВА

ЧЕРНОВА