

СОФИЙСКИ УНИВЕРСИТЕТ
„СВ. КЛИМЕНТ ОХРИДСКИ“ФАКУЛТЕТ ПО МАТЕМАТИКА
И ИНФОРМАТИКАДЪРЖАВЕН ИЗПИТ
ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР ПО ИНФОРМАТИКА”

ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)

12.07.2016 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. Дадена е квадратна матрица от дробни числа с размерност 11×11 . Казваме, че два елемента на матрицата са **симетрични**, ако те са разположени симетрично относно (1) главния диагонал, (2) вторичния диагонал, или (3) пресечната точка на двата диагонала на матрицата. На всеки елемент от матрицата съответстват най-много три симетрични елемента. Съвкупност, състояща се от елемент на матрицата заедно с всички негови симетрични елементи, наричаме **симетрична група**.

Пример: В матрицата с размерност 3×3

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix}$$

$\{a_{01}, a_{10}, a_{21}, a_{12}\}$ и $\{a_{00}, a_{22}\}$ са симетрични групи.

Да се състави функция, която променя матрицата така, че **всеки** елемент a_{ij} , $0 \leq i, j \leq 10$ от матрицата се заменя със средно-аритметичната стойност на елементите в неговата симетрична група.

Задача 2. Задачата да се реши с използване на език за процедурно или обектно-ориентирано програмиране (C, C++ или Java).

Нека в текстов файл се съдържа единствен ред текст, в който е описано представяне на дърво по следната схема:

$\langle \text{дърво} \rangle ::= (\langle \text{корен} \rangle (\langle \text{празен низ} \rangle | \langle \text{списък от наследници} \rangle)),$

където

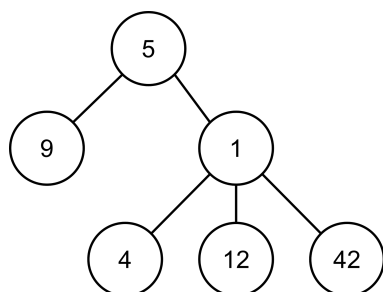
$\langle \text{корен} \rangle ::= \langle \text{цяло число} \rangle$

$\langle \text{списък от наследници} \rangle ::= \langle \text{дърво} \rangle | \langle \text{дърво} \rangle, \langle \text{списък от наследници} \rangle$

В текста се допуска наличието на произволен брой интервали между отделните елементи.

Да се напише функция, която получава като параметър низ, задаващ име на текстов файл, описващ дърво съгласно горната схема. Функцията да прочита съдържанието на файла, да построява в паметта дърво, съответстващо на текстовото описание, и да извежда на конзолата максималната сума от стойностите по път от корена на дървото до някое от листата му. Конкретното представяне на дървото в паметта е по Ваш избор.

Пример: Дървото, показано на долната диаграма, се представя по следния начин:



$(5 ((9 ()), (1 ((4 ()), (12 ()), (42 ())))))$

За този пример функцията следва да изведе на екрана числото 48
 $= 5 + 1 + 42.$

Задача 3. Нека са дадени следните изрази на езиците Haskell и Scheme. Да се посочи каква е оценката на изразите. 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:

Haskell:	
1.	<code>[x : [x] x <- [[1,2], [3,4]]]</code> Оценка:
2.	<code>[map (f 5) [1,2,3] f <- [(+), (-), (*)]]</code> Оценка:
3.	<code>"a" : [['b', 'c'], "d"]</code> Оценка:
Scheme:	
1.	<code>(map (lambda (x) (append (list x) x)) '((1 2) (3 4)))</code> Оценка:
2.	<code>(map (lambda (f) (map (lambda (x) (f 5 x)) '(1 2 3))) (list + - *))</code> Оценка:
3.	<code>(apply list (list (quote +) (quote 5) 8))</code> Оценка:

Задача 4. Дадена е базата от данни Movies, в която се съхранява информация за филми, филмови студиа, 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;

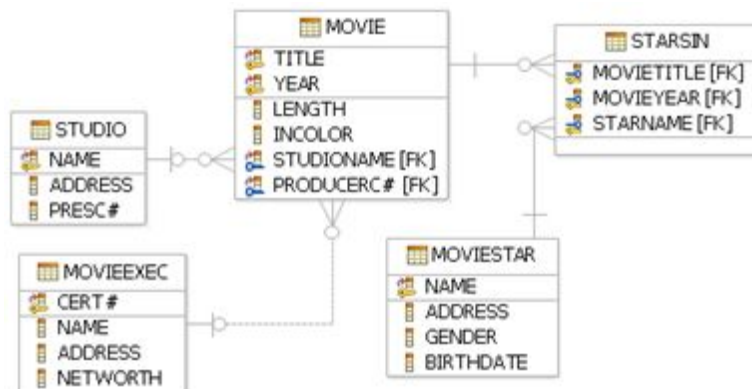
length – дължина в минути;

incolor – 'Y' за цветен филм и 'N' за

чернобыл;

*studio**name* – име на студио, външен ключ към таблицата Studio;

producer# – име на продуцент, външен ключ към таблицата MovieExec.



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

name – име, първичен ключ;

address – адрес;

presc# – номер на сертификат на президента на студиото.

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

name – име, първичен ключ;

address – адрес;

gender – пол, 'М' за мъж и 'F' за жена;

birthdate – рождена дата.

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 таблицата **Movie**.

movietitle – заглавие на филма;

movieyear – година на заснемане на филма;

starname – име на филмовата звезда, външен ключ към таблицата MovieStar.

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

cert# – номер на сертификат, първичен ключ;

name – ИМЕ;

address – адрес;

networth – нетни активи;

birthdate – рождена дата.

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

1. Да се посочи заявката, която извежда името на продуцента и имената на филмите, продуцирани от продуцента на 'Pretty Woman'. Възможно е името на продуцента на филма да не е известно.

А)

```
SELECT t.name, title
FROM movie m JOIN (SELECT name, cert#
                    FROM movieexec
                    WHERE cert# IN (SELECT producerc#
                                    FROM movie
                                    WHERE title='Pretty Woman')) t
ON m.producerc#=t.cert#;
```

Б)

```
SELECT name, title
FROM (SELECT cert#
      FROM movieexec
      INTERSECT
      SELECT producerc#
      FROM movie
      WHERE title ='Pretty Woman') t
```

В)

```
SELECT name, title
FROM movie m JOIN movieexec me ON m.producerc#=me.cert#
WHERE title = 'Pretty Woman';
```

Г)

```
SELECT name, title
FROM movie m LEFT JOIN movieexec me ON m.producerc#=me.cert#
WHERE me.cert# NOT IN (SELECT producerc#
                      FROM movie
                      WHERE title='Pretty Woman');
```


Задача 5. Текстов файл с име **comproc1** съдържа зададената по-долу последователност от команди на bash за Linux. Да се подчертаят операторите, които извеждат текст на стандартния изход и за всеки от тях да се напише вдясно какво ще бъде изведено след стартиране на файла със следния команден ред:

bash comprow1 1 3 5

ако на стандартния вход бъде подадена последователността от символи **c d**

```
br=1
br=`expr $br \* $2`
a=$3
set ab bc cd de
shift
while true
do echo $*
    for j
    do if test $# -lt $br
        then br=`expr $br / 2`
            echo $br $j >> file
        else br=`expr $br + $a`
            echo $br $j >> file
        fi
    done
break
done
read a1 a2
while cat file | grep $a1
do echo $a $a2
    wc -l file
    tail -5c file
    exit
    echo FIN
done
echo $a $a1
wc -c file
tail -2l file
```

Задача 6. Даден е детерминиран краен автомат, определен както следва:

- входна азбука $\{0, 1\}$
- множество от състояния $\{A, B, C, D, E\}$
- начално състояние A
- множество от крайни състояния $\{C, E\}$
- функция на преходите δ , определена по следния начин:

$$\begin{aligned} \delta(A, 0) = B, \delta(A, 1) = D, \delta(B, 0) = B, \delta(B, 1) = C, \delta(C, 0) = D, \delta(C, 1) = E, \\ \delta(D, 0) = D, \delta(D, 1) = E, \delta(E, 0) = B, \delta(E, 1) = C \end{aligned}$$

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

Задача 7. Нека е даден е неориентиран граф $G = (V, E)$. Нека $W = \{U \subseteq V : |U| = 2\}$.

Графът-допълнение на G , който бележим с \overline{G} , се дефинира като $\overline{G} = (V, \overline{E})$, където $\overline{E} = W \setminus E$ е множеството от точно тези възможни ребра, които не присъстват в G . Казваме, че графът $G = (V, E)$ е *изоморфен* на графа $G' = (V', E')$, ако съществува биекция $f: V \rightarrow V'$ такава, че за всеки два върха $x, y \in V$ е изпълнено следното: (x, y) е ребро в G тогава и само тогава, когато $(f(x), f(y))$ е ребро в G' . *Самодопълнителен граф* е всеки граф, който е изоморфен на своето допълнение.

Да се докаже, че ако $G = (V, E)$ е самодопълнителен граф, то $|V| \equiv 0 \pmod{4}$ или $|V| \equiv 1 \pmod{4}$.

Задача 8. Да се пресметне определеният интеграл

$$\int_1^2 x \log_2 \frac{1}{x} dx$$

ЧЕРНОВА