

СОФИЙСКИ УНИВЕРСИТЕТ  
„СВ. КЛИМЕНТ ОХРИДСКИ“



ФАКУЛТЕТ ПО МАТЕМАТИКА  
И ИНФОРМАТИКА

**ДЪРЖАВЕН ИЗПИТ**  
**ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР ПО КОМПЮТЪРНИ НАУКИ”**

**ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)**  
**9.09.2016 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. Задачата да се реши с използване на език за процедурно или обектно-ориентирано програмиране (C, C++ или Java).

Да се състави функция, която приема като параметър низ с произволна дължина и връща като резултат позициите на двойката **еднакви** символи, които са максимално отдалечени един от друг.

Ако в низа съществуват няколко двойки максимално отдалечени символи, функцията да връща позициите на най-ляво разположената двойка. Счита се, че номерата на позициите започват от 0.

Пример:

В символния низ "this is just a simple example" най-ляво и най-дясно разположените символи ' ' (интервали), са на позиции съответно 4 и 21, намират се на разстояние 17 символа един от друг и няма друга двойка еднакви символи, които са на по-голямо разстояние един от друг.

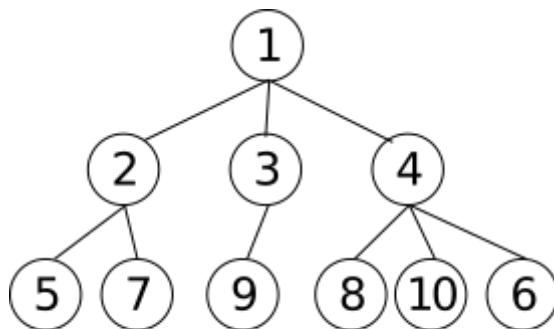
Задача 2. Задачата да се реши с използване на език за процедурно или обектно-ориентирано програмиране (C, C++ или Java).

Да се напише функция, която получава като параметри цяло число  $K$  и едномерен масив  $A$  с елементи различни цели числа. Функцията трябва да построи в паметта дърво  $T$ , съдържащо данните в масива  $A$ , като дървото  $T$  трябва да удовлетворява следните условия:

1. Всеки елемент на  $A$  се среща като възел в  $T$  точно веднъж.
2. Всеки възел в  $T$  има най-много  $K$  преки наследници (деца).
3.  $T$  е с възможно най-малка дълбочина.
4. Ако  $i < j$ , то  $A[i]$  да не се намира по-дълбоко от  $A[j]$  в  $T$  (т.е. да е на същата или по-малка дълбочина).

Като резултат функцията да връща построеното дърво и да извежда на стандартния изход неговата дълбочина. Конкретното представяне на дървото в паметта е по Ваш избор.

Пример: За  $A = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$  и  $K = 3$ , минималната дълбочина е **3**, а едно дърво  $T$ , удовлетворяващо горните условия, е:



Задача 3.

- Дадени да следните дефиниции на функция, съответно на програмните езици Haskell и Scheme, от програмния код на които липсват части. Да се попълнят полетата, обозначени с \_\_\_\_\_, с необходимия програмен код така, че да се получат посочените желани оценки.
- Дадени са следните изрази, съответно на програмните езици Haskell и Scheme. Да се посочи каква е оценката на израза.

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 избория от Вас език.

Избран език:

Haskell:	
1.	<p><code>filterByChar c ls = filter _____ ls</code></p> <p><u>израз:</u> <code>filterByChar 'o' ["cat", "cow", "dog"]</code></p> <p><u>желана оценка:</u> <code>["cow", "dog"]</code></p>
2.	<p><code>let (x:y):z = ["Curry"] in (x,y,z)</code></p> <p>Оценка:</p>
Scheme:	
1.	<p><code>(define (filterByElement x m)</code>  <code>(filter (lambda (__) _____) m))</code></p> <p><u>израз:</u> <code>(filterByElement 2 '( (1 2 3) (2 3 4) (3 4 5) ) )</code></p> <p><u>желана оценка:</u> <code>( (1 2 3) (2 3 4) )</code></p>
2.	<p><code>(apply + (map (lambda (l) (apply max l) ) '((5 -2) (1 9) (6 -8) )))</code></p> <p>Оценка:</p>

**Задача 4.** Дадена е базата от данни Ships, в която се съхранява информация за кораби и тяхното участие в битки по време на Втората световна война. Всеки кораб е построен по определен стереотип, определящ класа на кораба.

Таблицата **Classes** съдържа информация за класовете кораби:

*class* – име на класа, първичен ключ;

*type* – тип ('bb' за бойни кораби, 'bc' за бойни крайцери);

*country* – държава, която строи такива кораби;

*numGuns* – брой на основните оръдия, може да приема стойност *null*;

*bore* – калибър на оръдието (в инчове), може да приема стойност *null*;

*displacement* – водоизместимост (в тонове), може да приема стойност *null*.

Таблицата **Ships** съдържа информация за корабите:

*name* – име на кораб, първичен ключ;

*class* – име на класа на кораба, външен ключ към таблицата Classes;

*launched* – година, в която корабът е пуснат на вода, може да приема стойност *null*.

Таблицата **Battles** съхранява информация за битките:

*name* – име на битката, първичен ключ;

*date* – дата на провеждане.

Таблицата **Outcomes** съдържа информация за резултата от участието на даден кораб в дадена битка.

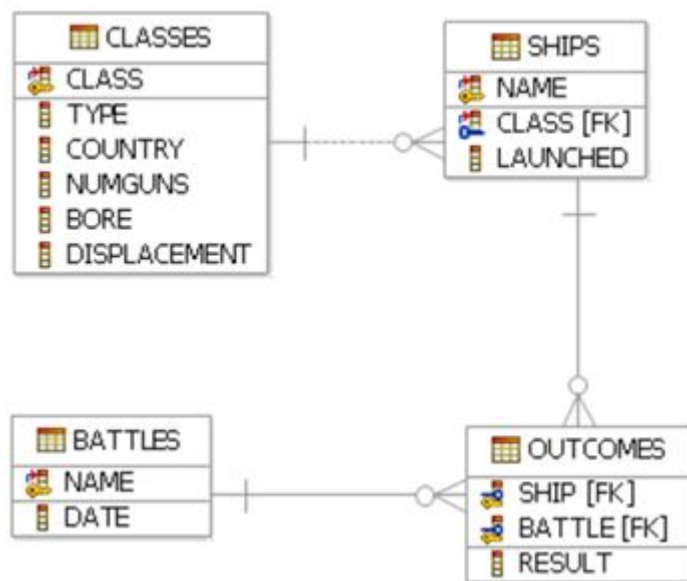
Атрибутите *ship* и *battle* заедно формират първичния ключ.

*ship* – име на кораба, външен ключ към таблицата Ships;

*battle* – име на битката, външен ключ към таблицата Battles;

*result* – резултат (потънал – 'sunk', повреден – 'damaged', победил – 'ok').

**Забележка за всички таблици:** За всички атрибути, за които не е посочено, че могат да приемат стойност *null*, да се счита, че съществува ограничение *not null*.



1. Да се посочи заявката, която извежда всички държави, които имат поне един кораб, участвал в битка, както и броя на потъналите кораби за всяка от държавите.

A)

```
SELECT c.country , COUNT(o.result)
FROM classes c left join ships s ON c.class=s.class
LEFT JOIN outcomes o ON s.name=o.ship
WHERE result='sunk' OR result IS NOT NULL
GROUP BY c.country;
```

Б)

```
SELECT c.country , COUNT(o.result)
FROM classes c JOIN ships s ON c.class=s.class
JOIN outcomes o ON s.name=o.ship
WHERE result='sunk'
GROUP BY c.country;
```

В)

```
SELECT c.country , COUNT(o.result)
FROM classes c join ships s ON c.class=s.class
JOIN outcomes o ON s.name=o.ship
JOIN battles b ON o.battle=b.name
ORDER BY c.country
HAVING result ='sunk';
```

Г)

```
SELECT DISTINCT c.country, (SELECT COUNT(o.result)
                             FROM classes c1 JOIN ships s
                             ON c1.class=s.class
                             JOIN outcomes o ON s.name=o.ship
                             WHERE result='sunk'
                             AND c1.country=c.country)
FROM classes c;
```

A)

Б)

B )

Г)

```
SELECT DISTINCT battle
FROM outcomes o CROSS JOIN classes c
GROUP BY battle
HAVING COUNT(DISTINCT country)>(SELECT COUNT(DISTINCT country)
                                FROM outcomes o, classes c
WHERE battle='Coral Sea');
```

Задача 5. Текстов файл с име **procA** съдържа зададената по-долу последователност от команди на **bash** за **Linux**. Да се напише вдясно какво ще бъде изведено на стандартния изход и какво ще бъде съдържанието на файловете **f1** и **f2** след стартиране на командната процедура със следния команден ред:

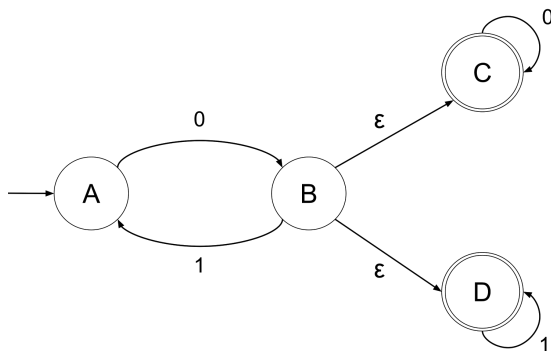
**bash procA ab bc cd**

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

```
count=1
for i in 6 1 4 2
do
    for each
    do if test $i -gt $#
        then count=`expr $count \* $i`
            echo $count $each >> f1
        else until false
            do echo $*
                break 3
            done
        fi
    done
done
read k1 k2
while cat f1 | grep $k1
do set $k1 $k2 $count
    shift
    echo $1 $2
    grep $2 f1 > f2
    wc -c f2
    exit
    echo END
done
wc -l f1
tail -2l f1
echo FIN
```



Задача 6. Даден е следният недетерминиран краен автомат:



Входната азбука е  $\{0, 1\}$ , множеството от състояния е  $\{A, B, C, D\}$ , началното състояние е A, множеството от крайни състояния е  $\{C, D\}$ , а преходите са илюстрирани на фигурата.

Да се построи детерминиран краен автомат, еквивалентен на дадения.

**Задача 7.** Нека е даден е неориентиран граф  $G = (V, E)$ . Цикъл в  $G$  е всяка последователност  $v_1, e_1, v_2, e_2, \dots, v_k, e_k, v_1$ , където  $v_1, v_2, \dots, v_k \in V$ , а  $e_1, e_2, \dots, e_k \in E$ , като  $e_i = (v_i, v_{i+1})$  за  $1 \leq i \leq k-1$  и  $e_k = (v_k, v_1)$ . Дължината на цикъла е броят на ребрата в него, като в предходната дефиниция дължината на цикъла е  $k$ . Разглеждаме само цикли, в които няма повтаряне на върхове, с изключение на това, че  $v_1$  се среща два пъти — в началото и в края, и няма повтаряне на ребра. При това ограничение най-малката възможна дължина на цикъл е 3. За всеки връх  $u$  в графа, *степента на  $u$*  е броят на съседите му.  $G$  е  $t$ -регулярен тогава и само тогава, когато всички върхове в  $G$  имат степен  $t$ .

Да се докаже, че ако най-малката дължина на цикъл в  $G$  е 4 и  $G$  е  $t$ -регулярен, то  $|V| \geq 2t$ .

Задача 8. Да се пресметне определеният интеграл

$$\int_0^{\pi/2} x \cos^2 x \, dx$$

**ЧЕРНОВА**