СОФИЙСКИ УНИВЕРСИТЕТ „СВ. КЛИМЕНТ ОХРИДСКИ“

ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

ДЪРЖАВЕН ИЗПИТ

ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР ПО СОФТУЕРНО ИНЖЕНЕРСТВО”

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.

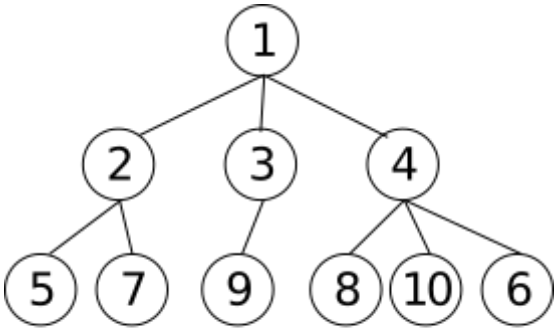
Задача 2. *Задачата да се реши с използване на език за процедурно или обектно-ориентирано програмиране (C, C++ или Java).*

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:



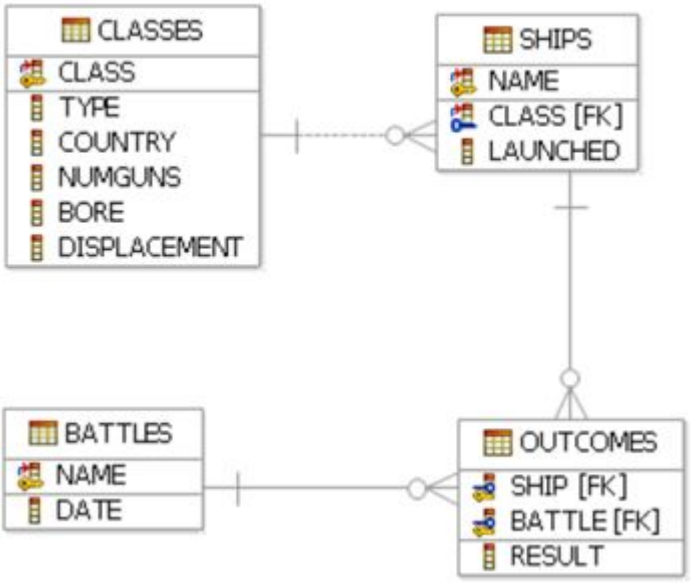
Задача 3. Върху спестовните сметки в банка се начисляват следните лихвени проценти:

- 0 – 100 лв. 3%
- 100 – 1000 лв. 5 %
- Над 1000 лв. 7 %

Счита се, че числените стойности са с точност до втория знак след десетичната запетая.

1) Да се определят валидните класове на еквивалентност.
2) Да се дефинират поне два невалидни класове на еквивалентност.
3) Да се определят граничните стойности за тестване.
4) Да се дефинират тестови сценарии, които да покриват класовете на еквивалентност и граничните стойности.

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.

А)
```
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;
```

2. Да се посочи заявката, която извежда имената на битките, които са по-мащабни (с кораби от повече държави) от битката в Коралово море (Coral Sea).

А)
```
SELECT DISTINCT battle
FROM outcomes o1
WHERE
(SELECT COUNT(DISTINCT country)
 FROM outcomes o,ships s,classes c
 WHERE o.ship=s.name AND s.class=c.class AND battle=o1.battle)
 >
(SELECT count(DISTINCT country)
 FROM outcomes o,ships s,classes c
 WHERE o.ship=s.name AND s.class=c.class AND battle='Coral Sea')
```

Б)
```
SELECT DISTINCT battle
FROM outcomes, (SELECT count(DISTINCT country) as NumCountries
                  FROM outcomes o,ships s,classes c
                  WHERE o.ship=s.name AND s.class=c.class
                    AND battle='Coral Sea') as t
HAVING (SELECT count(DISTINCT country)
        FROM outcomes o1,ships s,classes c
        WHERE o1.ship=s.name AND s.class=c.class AND battle=o1.battle)
         >
       (SELECT NumCountries FROM t);
```

В)
```
SELECT DISTINCT battle
FROM outcomes o, ships s, classes c
Where o.ship=s.name AND s.class=c.class
GROUP BY battle
HAVING COUNT(DISTINCT country)>(SELECT COUNT(country)
                         FROM outcomes o, classes c, ships s
                         WHERE ship=name AND s.class=c.class
                           AND battle='Coral Sea');
```

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');
```

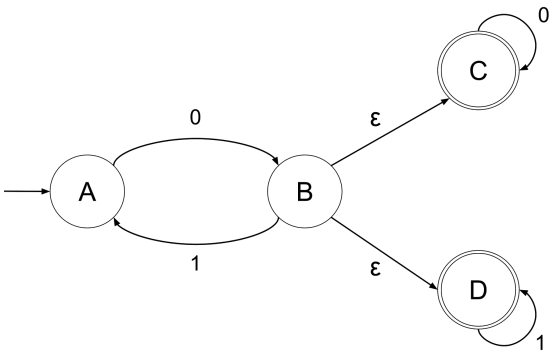
Задача 5. Текстов файл с име **procA** съдържа зададената по-долу последователност от команди на bash за Linux. Да се напише вдясно какво ще бъде изведено на стандартния изход и какво ще бъде съдържанието на файловете **f1** и **f2** след стартиране на командната процедура със следния команден ред:

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 състоянията 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. Софтуерна система „The Best Driver“ позволява на млади шофьори чрез игра да упражняват техниката си на шофиране в реални ситуации чрез симулация на уличната обстановка в голям град като София. По време на симулация системата разиграва реални ситуации, като потребителят трябва да реагира адекватно на пътната обстановка, имайки предвид правилата за движение, пътните знаци, светофарите, уличната маркировка. Системата има следните изисквания:

1. Нерегистриран потребител може да търси маршрути за симулация и да разглежда информация за тях.
2. Системата изисква регистрация за играчи, като това е възможно и чрез Facebook акаунт.
3. Регистриран потребител-администратор трябва да може да създава маршрут за симулация, като предоставя информация, задава ниво на трудност и задължително карта на маршрута, като за целта системата се свързва с платформата OpenStreetMap.
4. Регистриран потребител може да търси маршрут за симулация в нивото, на което се намира.
5. Регистриран потребител може да избере маршрут за симулация, който да изиграе, като това включва и заплащане на такса за избрания маршрут.
6. Системата поддържа следните начини за плащане: с карта или чрез системата ePay.
7. Системата добавя точки към профила на потребителя в зависимост от представянето му в дадена симулация.
8. При достигане на определен брой точки системата класира потребителя за следващо ниво.

Да се състави диаграма на потребителските случаи (Use Case) по така зададените изисквания.

Задача 8. Да се пресметне определеният интеграл

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

ЧЕРНОВА