

**СОФИЙСКИ УНИВЕРСИТЕТ
„СВ. КЛИМЕНТ ОХРИДСКИ“**



**ФАКУЛТЕТ ПО МАТЕМАТИКА
И ИНФОРМАТИКА**

**ДЪРЖАВЕН ИЗПИТ
ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР ПО СОФТУЕРНО ИНЖЕНЕРСТВО”**

**ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)
15.07.2014 г.**

Време за работа – 3 часа

Драги абсолвенти:

- Попълнете факултетния си номер в горния десен ъгъл на всички страници;
- Пишете само на предоставените листове без да ги разкопчавате;
- За всяка от задачите, беловата с решението може да е само на листите, на които е изписано условието на съответната задача, или на празна страница след условието. При необходимост пренасяте решението на подпечатан нов лист със заглавен текст „Задача N, стр. M, ф.н. F“, където M ($M \geq 1$) е поредния номер допълнителен лист за задача N, а F е вашият факултетен номер.

Изпитната комисия ви пожелава успешна работа!

Задача 1. (10 т.) Даден е недетерминираният краен автомат

$$A = \langle \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}, \{a, b, c\}, q_0, \delta, \{q_4, q_5\} \rangle$$

с функция на преходите δ , определена както следва:

δ	a	b	c
q_0	$\{q_1, q_3, q_5\}$	$\{q_5\}$	\emptyset
q_1	\emptyset	\emptyset	$\{q_2\}$
q_2	\emptyset	$\{q_4\}$	\emptyset
q_3	\emptyset	$\{q_3, q_4\}$	\emptyset
q_4	\emptyset	\emptyset	\emptyset
q_5	\emptyset	\emptyset	$\{q_6\}$
q_6	\emptyset	\emptyset	$\{q_7\}$
q_7	$\{q_7\}$	$\{q_5\}$	\emptyset

Да се построи детерминиран краен автомат A' , еквивалентен на A.

Задача 2. (10 т.) Текстов файл с име `comproc1` съдържа зададената по-долу последователност от команди на `bash` за Linux. Напишете внятно какво ще бъде изведено на стандартния изход след стартиране на файла с команден ред

```
bash comproc1 ab cd ef
```

ако на стандартния вход бъде подадена следната последователност от символи: 1 2

```
count=1
for i in 5 1 4 2
do for j
    do if test $i -ge $#
        then count=`expr $count \* $i`
            echo $count $j >> f1
        else while true
            do echo $*
                break 3
            done
        fi
    done
done
read k1 k2
while cat f1 | grep $k2
do set $k1 $count
    shift
    echo $2
    echo $1 $i
    exit
done
echo FIN
```

Задача 3. (10 т.) *Задачата да се реши на езика C, C++ или Java. В началото на вашето решение посочете кой език сте избрали.*

Да се напише програма, която създава **едномерен масив** А с 27 елемента и след това:

- a) Генерира по случаен начин стойности в интервала (0, 4.9) за елементите на така зададения масив А;
- b) Разпечатва данните от масива в 5 КОЛОНИ, по диапазони, както следва: [0,1), [1,2), [2,3), [3,4), [4,5) Етикетите на колоните да бъдат съответно „Под 1”, „Под 2”, „Под 3“ и т.н. – както е показано в примера по-долу. При извеждането данните да бъдат подравнени и форматирувани по следния начин: всяка колона да бъде с ширина 8 символа, числата в колоните да бъдат с 4 знака след запетайката и изведени с дясно подравняване, а етикетите – с ляво подравняване.

Примерен резултат от програмата*:

Под 1	Под 2	Под 3	Под 4	Под 5
0,004	1,028	2,068	3,272	4,376
0,156	1,412	2,656	3,724	4,688
0,22	1,432	2,96	3,828	4,716
0,272	1,436			4,748
0,288	1,544			4,82
0,396	1,768			4,888
0,416	1,992			
0,46				

* Забележка: НЕ Е необходимо резултатите във всяка от колоните да бъдат сортирани.

Задача 4. (10 т.) *Задачата да се реши на езика C++ или Java. В началото на вашето решение посочете кой език сте избрали.*

А) Да се дефинира структура `ChessPosition`, описваща коректна позиция на фигура върху шахматна дъска (координатите на позицията са от 'А' до 'Н' по едното измерение и от 1 до 8 по другото).

Да се дефинира абстрактен клас (или интерфейс) `ChessPiece`, описващ шахматна фигура със следните операции:

- `ChessPosition getPosition()` – Връща позицията на фигурата на дъската;
- [подходящ тип] `allowedMoves()` – Връща списък (колекция) с всички възможни позиции, до които дадена фигура може да достигне с един ход;
- [булев тип] `captures(ChessPosition pos)` – Проверява дали фигурата “владее” позицията `pos`, подадена като параметър, т.е. дали позицията е в списъка с възможните ходове на фигурата. Булевият тип да бъде булевият тип в езика, който сте избрали (напр. `bool`, ако пишете на C++).

Б) Да се дефинират класовете `Rook` и `Knight` – наследници на `ChessPiece`, описващи съответно шахматните фигури топ и кон.

В) „Стабилна конфигурация“ наричаме такава подредба на фигурите върху дъската, при която никоя фигура да не е върху позволен ход на друга фигура (т.е. никои две фигури да не се „бият“). Да се дефинира функцията `allMoves ([подходящ тип] pieces[, ...])`, която за списъка (колекцията) `pieces`, съдържащ произволен брой разнородни шахматни фигури, отпечатва на конзолата всеки възможен ход на фигура от `pieces` такъв, че след изпълнението му списъкът с фигури да описва стабилна конфигурация. Информацията за ходовете да съдържа типа на фигурата, старата позиция и новата позиция, например:

`Rook A1 -> B1`

`Knight B3 -> A5`

Забележка: Реализирайте всички конструктори и други операции, които смятате, че са необходими на съответните класове.

Задача 5. (10 т.) Да се проектира софтуерната архитектура на система за онлайн редактиране на документи, според следните изисквания:

- Документите се запазват в отворен формат, който позволява лесното им четене от други редактори.
- Системата да може да чете формати за документи от други популярни редактори.
- Улесняване на работата на потребителите чрез въвеждане на операции по undo, redo и автоматичен запис през определен интервал от време.
- Системата да поддържа едновременна работа на до 5 потребителя едновременно, без това да води до загуба на информация
- Документите да имат различни нива за достъп: скрит, достъпен само за четене, достъпен само за четене с покана, пълни права само за поканени, пълни права за всички (публичен документ)

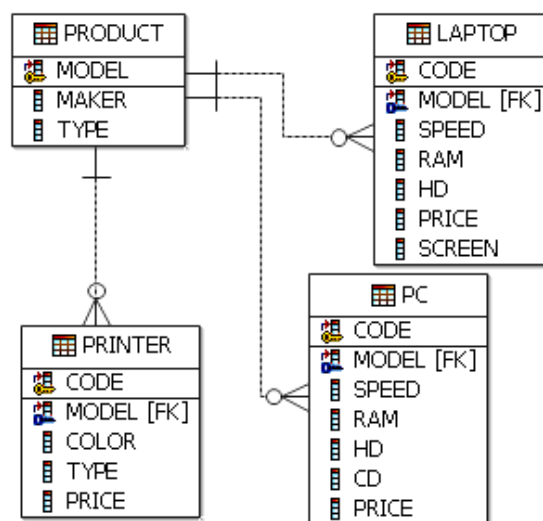
Задача 6. (10 т.) Дадена е базата от данни РС. В нея се съхранява информация за три вида продукти – настолни компютри, лаптопи и принтери.

Таблицата **Product** съдържа базова информация за всеки продукт:

- *model* – модел на продукта, първичен ключ;
- *maker* – производител на продукта;
- *type* – един от следните типове: 'PC', 'Laptop' или 'Printer'.

Таблицата **PC** съдържа специфична информация за настолните компютри:

- *code* – уникален идентификатор на дадена компютърна конфигурация, първичен ключ;
- *model* – модел на настолния компютър, външен ключ към Product.model. Може да имаме няколко различни компютърни конфигурации от един и същ модел, но с различни параметри;
- *speed* – тактова честота на процесора в MHz;
- *ram* – количество RAM памет в MB;
- *hd* – размер на твърдия диск в GB;
- *cd* – скорост на CD устройството;
- *price* – цена на настолния компютър.



Таблицата **Laptop** съдържа специфична информация за лаптопите. Атрибутите са аналогични на тези на PC, но липсва атрибутът CD и има атрибут за размера на екрана.

Таблицата **Printer** съдържа информация за принтерите:

- *code*, *model*, *price* – аналогични на едноименните атрибути в PC;
- *color* – 'y' за цветен принтер, 'n' за черно-бял;
- *type* – тип на принтера – 'Laser', 'Jet', 'Matrix'.

За така описаната база данни решете следните задачи:

1. Оградете буквата на заявката, която извежда всички производители на настолни компютри, които произвеждат и лаптопи.

A) <pre>select distinct maker from product where product.type = 'PC' and maker in (select maker from product join laptop on product.model = laptop.model);</pre>	Б) <pre>select maker from product p1 cross join product p2 where p1.maker = p2.maker and p1.type = 'PC' and p2.type = 'Laptop' group by maker;</pre>
В) <pre>select distinct maker from product where type = 'PC' and type = 'Laptop';</pre>	Г) <pre>select maker from product where type = 'PC' union select maker from product where type = 'Laptop';</pre>

2. Оградете буквата на заявката, която извежда кодовете, моделите и размерите на екраните на всички лаптопи, чиито производители имат не повече от три модела принтери (евентуално 0).

A) <pre>select code, model, screen from laptop, product where maker is having count(printer.model) <= 3;</pre>	Б) <pre>select code, l.model, screen from product p left join laptop l on p.model = l.model having count(select * from product where maker = p.maker and type = 'Printer') <= 3;</pre>
В) <pre>select code, l.model, screen from laptop l inner join product p on l.model = p.model where maker not in (select maker from product where type = 'Printer' group by maker having count(*) > 3);</pre>	Г) <pre>select l.code, l.model, l.screen from laptop l join product p on l.model = p.model where maker in (select maker from product where type = 'Printer' group by maker having count(*) <= 3);</pre>
Д) <pre>select l.code, l.model, l.screen from product p left join laptop l on p.model = l.model left join printer on p.model = printer.model group by l.code having count(distinct printer.code) <= 3;</pre>	

Задача 7. (10 т.) Дадена е информационна система, която съхранява информация за Световното първенство по футбол през 2014 г. Базата от данни се състои от следните таблици:

Групи (Groups)

- Име на група (gname) - низ, точно един символ (например А, В и т.н.), първичен ключ
- Първо място в групата (fplace) – низ не по-голям от 20 символа, името на държавата (отбора) спечелила първо място в групата, външен ключ към колоната Държава от таблицата Отбори
- Второ място в групата (splace) – низ не по-голям от 20 символа, името на държавата (отбора) спечелила второ място в групата, външен ключ към колоната Държава от таблицата Отбори

Отбори (Teams)

- Държава (country) – низ не по-голям от 20 символа, първичен ключ
- Име на група (gname) - низ, точно един символ, външен ключ към колоната Име на група от таблицата Групи
- Брой участия на световни първенства (num_ws), цяло **положително** число

Играчи (Players)

- Име на играч (pname) – низ не по-голям от 50 символа, първичен ключ
- Отбор на играча (team) - низ не по-голям от 20 символа, външен ключ към колоната Държава на таблицата Отбори
- Номер на фланелка (num_player), цяло **положително** число **от 1 до 22**
- Брой отбелязани голове на това световно първенство (num_goals), цяло **положително** число

Мачове (Matches)

- Номер на мач (mnumber) – цяло **положително** число, първичен ключ
- Вид на фаза (tstage) - от коя фаза е мача (групова фаза, елиминационна фаза), низ не по-голям от 8 символа
- Фаза от директната елиминация (stage) – низ точно 3 символа, например осминафинали ($\frac{1}{8}$), четвъртфинали ($\frac{1}{4}$), полуфинали ($\frac{1}{2}$) и т.н. Може да бъде NULL ако мача е от групов фаза.
- Отбор домакин (team_host) - низ не по-голям от 20 символа, външен ключ към колоната Държава на таблицата Отбори
- Отбор гост (team_guest) - низ не по-голям от 20 символа, външен ключ към колоната Държава на таблицата Отбори
- Резултат от мача (result) – низ, точно 1 символ (например 1 при победа на първия отбор, 2 при победа на втория отбор)
- Голова разлика (scores) – низ точно 5 символа, например 5:3, 10:10 и т.н.
- Дата на провеждане на мача (date) – дата
- Час на провеждане на мача (time) – час

Като използвате SQL създайте така описаните таблици и ограничения.

Задача 8. (10 т.) Да се направи информационен модел на електронен бележник с адреси и телефони на хора, който да замести хартиения бележник. Да се аргументира изборът за всеки обект от модела, структурата на модела и броя на появяванията на обектите. Бележникът съдържа: собствено, бащино и фамилно име на роднини, колеги и познати, адрес, съдържащ пощенски код, името на града, име и номер на улица/булевард, дата на раждане за обаждаме за поздрав за рожден ден, телефони (ако има) – домашни и служебни. Това съдържание е записано във XML документ, в който има информация за нула, един или повече души. Да се състави описание на документния му тип с използване на DTD или XML Schema. Да се създадат примерен XML документ, в който има данни за поне двама души, и съответен DTD или XML Schema документ.