

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



**ФАКУЛТЕТ ПО МАТЕМАТИКА
И ИНФОРМАТИКА**

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

ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)

09.07.2019 г.

Драги абсолвенти:

- Попълнете факултетния си номер в горния десен ъгъл на всички листове.
- Пишете само на предоставените листове, без да ги разкопчавате.
- **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 имате нужда от допълнителен лист, можете да поискате от квесторите.
- На един лист не може да има едновременно и чернова, и белова.
- Черновите трябва да се маркират, като най-отгоре на листа напишете "ЧЕРНОВА".
- Ако решението на една задача не се побира на нейния лист, трябва да поискате нов бял лист от квесторите. Той трябва да се защити с телбод към листа със задачата.
- Всеки от допълнителните листове (белова или чернова) трябва да се надпише най-отгоре с вашия факултетен номер.
- Черновите също се предават и се защитават в края на работата.
- Времето за работа по изпита е 3 часа.

Изпитната комисия ви пожелава успешна работа!

Задача 1. Задачата да се реши на езика C++.

Даден е двумерен масив с размер 6 на 6 от символи — малки и главни латински букви и цифри. Две клетки в него ще наричаме “съседни”, ако имат обща стена (т.е. всяка клетка е съседна с най-много четири други, намиращи се под, над, вляво и вдясно от нея). Път с дължина N ще наричаме редица $a_0, a_1, \dots, a_N - 1$ от клетки, за която:

1. за всяко $0 \leq i < N - 1$ е изпълнено, че a_i и a_{i+1} са съседни;
2. никоя от клетките не се среща повече от веднъж (т.е. няма цикли).

Да се попълнят празните места в кода на дадените по-долу функция `contains` и помощната ѝ функция `walk`. Функцията `contains` получава два аргумента – масив `arr` от дадения тип `char[6][6]` и символен низ `str`. Тя трябва да връща истина тогава и само тогава, когато в `arr` съществува път, чиито клетки образуват точно съдържанието на низа `str` (вижте примера по-долу). За определеност считаме, че функцията трябва да връща истина за празния низ.

Пример: За дадения по-долу двумерен масив `contains` трябва да върне истина, ако ѝ бъдат подадени низовете "abcdefgh", "A123B123C" или "". За улеснение, за да може да ги видите по-лесно, те са маркирани в сиво.

y	u	f	a	b	c
G	o	p	g	B	1
c	b	a	h	3	2
d	k	j	i	2	3
e	f	Q	N	1	C
h	g	h	M	A	r

Кодът на двете функции е даден на следващия лист:



```
bool contains(char arr[6][6], const char* str)
```

```
{
    for (int row = 0; row < ____; _____)
        for (int col = 0; col < ____; _____)
            if (walk(arr, row, col, str))
                return ____;
    return ____;
}

bool walk(char arr[6][6], int row, int col, const char* str)
{
    if (*str == '\\0')
        return ____;

    if (row < 0 || col < 0 || row >= 6 || col >= 6)
        return ____;

    if (arr[row][col] != *str)
        return ____;

    arr[row][col] *= -1;

    bool result =
        walk(arr, row + ___, col, str + 1) ||
        walk(arr, ____, ____, str + 1) ||
        walk(arr, ____, ____, str + 1) ||
        walk(arr, ____, ____, str + 1);

    arr[row][col] ____;

    return result;
}
```

Задача 2. Задачата да се реши на един от езиките C, C++ или Java. В началото на решението си посочете кой език сте избрали.

Разглеждаме кореново дърво, във възлите на което има записани двойки от символ (`char`) и цяло число (`int`). Всеки възел на дървото може да има произволен, краен брой наследници. За удобство разглеждаме функциите *sym* и *val*, дефинирани над множеството от възлите на дървото, така че за всеки възел *v* на дървото, в който е записана двойката $\langle a, b \rangle$, $\text{sym}(v) = a$ и $\text{val}(v) = b$.

Клон в дървото *T* ще наричаме всеки път $\pi = (v_0, v_1, \dots, v_n)$, за който v_0, \dots, v_n са върхове на *T*, v_n е листо на *T*, а v_i е родител на v_{i+1} за всяко $i < n$. За всеки клон $\pi = (v_0, v_1, \dots, v_n)$ на дървото *T* дефинираме съответни “дума” и “стойност” по следния начин:

$\text{word}(\pi) = \text{sym}(v_0)\text{sym}(v_1)\dots\text{sym}(v_n)$,

$$\text{value}(\pi) = \sum_{i=0}^n \text{val}(v_i),$$

т.е. $\text{word}(\pi)$ е думата, която се получава от последователното прочитане на символите, записани във възлите на пътя, а $\text{value}(\pi)$ е сумата на числата, записани в тях.

- а) Да се избере, дефинира и опише подходящо представяне на дърво от описания тип.
б) За така дефинираното представяне да се реализира функцията:

int sumVal ([подходящ тип] *T*, [подходящ тип] *u*, [подходящ тип] *v*),

която по дадено дърво *T* и два негови върха *u* и *v* намира и връща сумата от всички стойности $\text{value}(\pi_u) + \text{value}(\pi_v)$, за които π_u и π_v са клони с начала *u* и *v* (съответно) със свойството $\text{word}(\pi_u) = \text{word}(\pi_v)$. Ако такива клони няма, сумата се счита за 0.

Забележки:

1. В зависимост от избраното представяне, параметърът *T* може да бъде пропуснат.
2. Не е нужно дефиницията на представянето на дървото да бъде пълна, нужно е само да е достатъчна за реализацията на функцията **sumVal**.
3. Позволено е използването на функциите и класовете от стандартната библиотека на избирания от Вас език.

Задача 3. Спецификация на софтуерна система за провеждане на обучения за получаване на допълнителна квалификация за учители изисква попълване на формуляр за заявка. Формулярът включва попълване на 4 полета за избор (ОКС, научно звание, специалност, тип идентификационен номер), като всяко от тях предлага следните опции:

- ОКС: Бакалавър, Магистър, Доктор
- Научно звание: Да, Не
- Специалност: Математика, Информатика, Компютърни науки
- Тип идентификационен номер (ИН): ЕГН, ЛНЧ, Друго

Да се дефинират тестови сценарии, покриващи избора на стойности от полетата за избор чрез прилагане на техниката за тестване по двойки с ортогонален масив. Да се опишат стъпките за прилагане на техниката и да се представят тестовите сценарии в таблица.

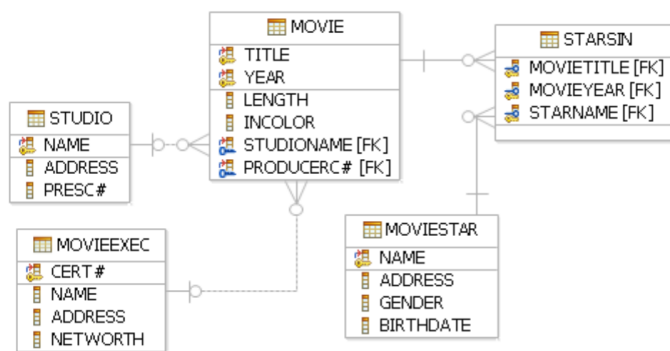
Задача 4. Дадена е базата от данни **Movies**, в която се съхранява информация за филми, филмови студия, които ги произвеждат, продуцентите на филмите, както и актьорите, които участват в тях.

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 чернобял;
- **studioName** – име на студио, външен ключ към **Studio.name**;
- **producerc#** – номер на сертификат на продуцента, външен ключ към **MovieExec.cert#**.

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 **Movie**.

- **movietitle** – заглавие на филма;
- **movieyear** – година на заснемане на филма;
- **starname** – име на филмовата звезда, външен ключ към **MovieStar.name**.



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

- **name** – име, първичен ключ;
- **address** – адрес;
- **gender** – пол, 'M' за мъж (актьор) и 'F' за жена (актриса);
- **birthdate** – рождена дата.

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

- **name** – име, първичен ключ;
- **address** – адрес;
- **presc#** – номер на сертификат на президента на студиото.

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

- **cert#** – номер на сертификат, първичен ключ;
- **name** – име;
- **address** – адрес;
- **networth** – нетни активи.

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

а) Да се ограда буквата на заявката, която извежда име на студио и броя на филмите му, за тези студия с по-малко от два филма. Студията, които нямат нито един филм, **НЕ** трябва да присъстват в резултата.

A) SELECT S.NAME, COUNT(M.TITLE) as CNT
FROM STUDIO S JOIN MOVIE M
ON S.NAME = M.STUDIONAME
GROUP BY S.NAME
HAVING CNT < 2;

B) SELECT S.NAME, COUNT(M.TITLE) as CNT
FROM STUDIO S LEFT JOIN MOVIE M
ON S.NAME = M.STUDIONAME
WHERE M.TITLE IS NULL
GROUP BY S.NAME
HAVING COUNT(M.TITLE) < 2;

C) SELECT S.NAME, COUNT(M.TITLE) as CNT
FROM STUDIO S JOIN MOVIE M
ON S.NAME = M.STUDIONAME
GROUP BY S.NAME
HAVING COUNT(M.TITLE) < 2;

D) SELECT S.NAME, COUNT(M.TITLE) as CNT
FROM STUDIO S JOIN MOVIE M
ON S.NAME = M.STUDIONAME
WHERE COUNT(M.TITLE) < 2
GROUP BY S.NAME;

б) Да се напише заявка, която да изведе имената на всички продуценти с минимален нетен актив.

Задача 5. Всеки от процесите **P**, **Q** и **R** изпълнява поредица от три инструкции:

<u>process P</u>	<u>process Q</u>	<u>process R</u>
p_1	q_1	r_1
p_2	q_2	r_2
p_3	q_3	r_3

Да се осигури чрез семафори синхронизация на **P**, **Q** и **R**, така че да са изпълнени следните изисквания:

- инструкция p_1 да се изпълни преди q_2 и r_2;
- инструкция r_2 да се изпълни преди p_3.

Забележка: Решение с 2 семафора ще бъде оценено с пълен брой точки.

Упътвания:

- Семафорът е обект за синхронизация, локалните му данни са брояч `cnt` и списък на приспаните процеси `L`. Конструкторът му `init(n)` присвоява начална стойност на брояча (`cnt=n`), списъкът се инициализира да бъде празен.

Семафорът има два метода – `wait()` и `signal()`.

Методът `wait()` намалява с единица брояча `cnt` и ако стойността на брояча стане отрицателна, добавя в списъка `L` информация за текущия процес и го спира временно (процесът бива приспан, блокиран).

Методът `signal()` увеличава `cnt` и ако стойността на брояча преди увеличението е отрицателна, изважда процес от списъка `L` и го събужда.

- Да се приеме, че инициализацията на семафорите се извършва от процес, който поражда процесите **P**, **Q** и **R** преди тяхното стартиране.

Задача 6. Софтуерна система организира корабоплаването по река Дунав, като следи и управлява движението на пътнически и товарни кораби.

- R1. Системата изисква всички плавателните съдове, които тя обслужва, да се регистрират в нея, като регистрацията налага електронно банково заплащане на такса.
- R2. Чрез системата отговорните институции на Дунавските страни съгласуват разписанията на регистрираните пътнически кораби и съгласуваните разписания се предоставят на съответните кораби.
- R3. Чрез системата отговорните институции на Дунавските страни могат да променят разписанието на пътническите кораби два пъти в годината.
- R4. Софтуерната система управлява регистрираните кораби, като следи чрез сателит и записва местоположението, а при необходимост и налага корекция в движението им.
- R5. Чрез системата регистрираните кораби могат да получат информация, а при желание от тях и картина, от местоположението на други регистрирани кораби.
- R6. Системата ежедневно информира за състоянието на водата, като за целта събира информация от сензорни системи за нивото и замърсяването на водата в отделните участъци на реката, съхранява тази информация, и я изпраща до регистрираните кораби, а в случай на измервания извън нормите – и до съответните институции на Дунавските страни.

Да се състави диаграма на потребителските случаи (Use Case) по дадените изисквания. Да се отбележат връзките между потребителските случаи и да се обоснове тяхното използване в диаграмата.

Задача 7. Да се пресметне интегралът:

$$\int_1^8 \frac{dx}{\sqrt[3]{x} + x}$$

Чернова