

СОФИЙСКИ УНИВЕРСИТЕТ
“СВ. КЛИМЕНТ ОХРИДСКИ”



ФАКУЛТЕТ ПО МАТЕМАТИКА
И ИНФОРМАТИКА

ДЪРЖАВЕН ИЗПИТ

ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР ПО СОФТУЕРНО ИНЖЕНЕРСТВО”

ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)

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/C++.

Да се попълнят празните места в кода на функциите така, че те да отговарят на описанието си.

```
// А) find използва алгоритъма за двоично търсене (binary search), за да
// провери дали value се съдържа в масива arr, състоящ се от size елемента.
// Функцията връща true ако това е така и false в противен случай.

bool find(int value, int* arr, size_t size)
{
    if (size == 0) return _____;

    size_t mid = size / 2;
    if (value == arr[mid]) return _____;

    if (value < arr[mid])
        return find(_____, _____, _____);
    else
        return find(_____, arr + _____, _____);
}

// Б) fold_left изпълнява ляво свиване (left fold) върху масива arr, съдържащ size елемента,
// прилагайки операцията op. Началната стойност е nil.
// Функцията връща стойността op(...op(op(nil, a[0]), a[1]), ..., a[size-1]).

template <typename ReturnType, typename InputType, typename OpType>
ReturnType fold_left(InputType* arr, size_t size, OpType op, ReturnType nil)
{
    _____ result = _____;

    for (size_t i = 0; i < _____; ++i)
        result = op(_____, _____);

    return result;
}

int op(char Digit, int Result)
{
    return (_____ * 10) + (_____ - '0');
}

// Преобразува символен низ от десетични цифри до величина от тип int
int str_to_int(const char * str)
{
    return (str == nullptr) ? 0 : fold_left(str, _____, op, _____);
}
```

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

Дърво с етикети ще наричаме кореново дърво $T = (V, E, r)$ с множество от върхове V , множество от ребра E и корен r , за което са дефинирани две допълнителни функции:

$$value : E \rightarrow \{n \in \mathbb{N} \mid 0 \leq n < 2^{32}\} \quad \text{и} \quad label : E \rightarrow \{a, b, \dots, z\}.$$

Клон в T ще наричаме път $\pi = (v_0, v_1, \dots, v_n)$, за който v_n е листо на T и v_i е родител на v_{i+1} за всяко $i < n$.

За всеки клон $\pi = (v_0, v_1, \dots, v_n)$ дефинираме:

$$val(\pi) = \sum_{i=0}^{n-1} value(\langle v_i, v_{i+1} \rangle) \quad \text{и} \quad word(\pi) = label(\langle v_0, v_1 \rangle) label(\langle v_1, v_2 \rangle) \dots label(\langle v_{n-1}, v_n \rangle).$$

А) Да се избере, дефинира и опише подходящо представяне на дърво от описания вид.

Б) За така дефинираното представяне да се реализира функцията:

CommonBranches(<labeled tree> T, <vertex> u, <vertex> v, <integer> k),

която по дадено дърво с етикети $T = (V, E, r)$, два негови върха u и v и естествено число k извежда на стандартния изход всички думи w , за които има клони π_u и π_v с начало u и съответно v , за които $word(\pi_u) = word(\pi_v) = w$ и $val(\pi_u) + val(\pi_v) = k$.

Забележки:

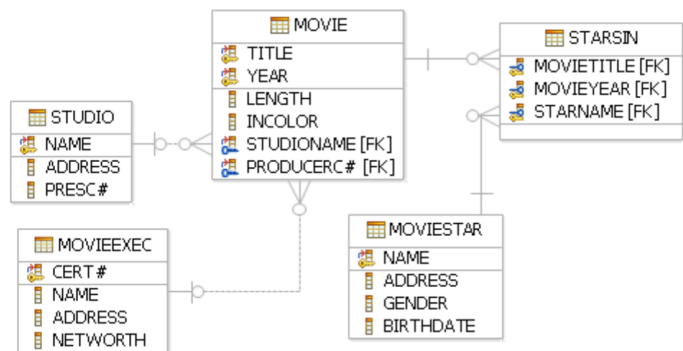
- Отделните думи да са разделени със символа за нов ред. Една дума може да се извежда повече от веднъж.
- Функционалности на структурата дърво с етикети, които нямат отношение към задачата и не се използват във функцията CommonBranches, няма да бъдат оценявани.

Задача 3. Софтуерна система за изчисляване на възнаграждения на банкови служители предоставя следната функционалност:

1. Външен експерт, работещ повече от 40 часа на седмица, получава заплащане на час, като първите 40 часа се заплащат регулярно, а следващите над 40 се заплащат двойно.
2. Външен експерт, работещ по-малко от 40 часа на седмица, получава заплащане на час като изработените часове се заплащат регулярно и се създава протокол за неизпълнени задъжения.
3. Служител на трудов договор, работещ по-малко от 40 часа на седмица, получава заплата и се създава протокол за неизпълнени задължения.
4. Служител на трудов договор, работещ повече от 40 часа на седмица, получава заплата.

Да се дефинира таблица за взимане на решения, въз основа на която могат да се генерират тестови сценарии. Таблицата трябва да показва условията, следствията от тях и правилата, от които се генерират тестовите сценарии.

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 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 MovieExec съдържа информация за продуцентите на филми.

- cert# — номер на сертификата, първичен ключ
- name — име
- address — адрес
- networth — нетни активи

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 сертификата на продуцента, външен ключ към MovieExec.cert#.

Таблицата Studio съдържа информация за филмови студия:

- name — име, първичен ключ
- address — адрес;
- presc# — номер на сертификата на президента на студиото.

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.

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

Зад 1. Да се огради буквата на заявката, която извежда за всеки продуцент името му и броя на филмите му по години. Продуценти, които нямат нито един филм, НЕ трябва да присъстват в резултатното множество.

A)

```
SELECT ME.NAME, M.YEAR, COUNT(*) AS CNT
FROM MOVIEEXEC ME LEFT JOIN MOVIE M
ON ME.CERT# = M.PRODUCERC#
WHERE M.TITLE IS NULL
GROUP BY ME.CERT#, ME.NAME, M.YEAR;
```

Б)

```
SELECT ME.NAME, M.YEAR, COUNT(*) AS CNT
FROM MOVIEEXEC ME
JOIN MOVIE M
ON ME.CERT# = M.PRODUCERC#
GROUP BY ME.CERT#, ME.NAME, M.YEAR;
```

В)

```
SELECT ME.NAME, M.YEAR, COUNT(*) AS CNT
FROM MOVIEEXEC ME, MOVIE M
GROUP BY ME.CERT#, ME.NAME, M.YEAR
WHERE ME.CERT# = M.PRODUCERC#;
```

Г)

```
SELECT ME.NAME, M.YEAR, COUNT(*) AS CNT
FROM MOVIEEXEC ME
JOIN MOVIE M
ON ME.CERT# = M.PRODUCERC#
ORDER BY ME.CERT#, ME.NAME, M.YEAR;
```

Зад 2. Да се напише заявка, която да изведе името на най-младата звезда (полът е без значение).

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

process P	process Q	process R
p_1	q_1	r_1
p_2	q_2	r_2
p_3	q_3	r_3

Чрез семафори да се осигури синхронизация на P, Q и R, така че да се изпълнят заедно следните изисквания:

- (а) Инструкцията p_1 да се изпълни преди q_2 и r_2.
- (б) Ако q_2 се изпълни преди r_2, то и q_3 да се изпълни преди r_2.
- (в) Ако r_2 се изпълни преди q_2, то и r_3 да се изпълни преди q_2.

Упътвания:

1. 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 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. Ако от *L* се вади най-рано приспания процес, наричаме семафора силен. Всяка друга стратегия на събуждане реализира слаб семафор.

2. Да се приеме, че инициализацията на семафорите се прави от процес, който поражда процесите, обсъждани в условието на задачата, преди тяхното стартиране.

Задача 6. Да се направи декомпозиция на модулите от архитектурата на софтуерна система за споделяне на файлове според дадените по-долу изисквания. Да се обоснове защо така проектираната архитектура удовлетворява изискванията.

R1. Системата трябва да поддържа следните групи потребители:

- (а) Регистриран потребител
- (б) Супер-потребител
- (в) Гост

R2. Към потребителите се причисляват и системните администратори, които имат само офлайн достъп до системата и имат пълен достъп до всички данни и настройки на системата.

R3. Заявките за качване и сваляне на файлове от супер-потребителите са с по-висок приоритет от тези на регистрираните потребители, а тези на гостите – с най-нисък приоритет от всички. Необходимо е да може да се гарантира времето за достъп до системата на всяка група потребители, според нивото им.

R4. Всички качени данни от потребителите трябва да са защитени както от загуба, така и от неоторизиран достъп. Необходимо е също всеки потребител да може да споделя качените от него файлове с различни права за достъп.

R5. Достъпът до системата се осъществява чрез мобилно приложение или уеб-браузър.

R6. При наличие на проблем (например вследствие на натоварване или злонамерена атака), да се предвиди възможност за уведомяване на администраторите в рамките на 30 секунди.

R7. Да се предвиди възможност за интегриране на системата с други популярни платформи за споделяне на съдържание (YouTube, Spotify, социални мрежи).

R8. Системата трябва да е налична в режим на работа 24/7, като се допуска до 6 часа месечно да няма достъп поради поддръжка.

Задача 7. Да се пресметне интегралът $\int_0^2 \ln(x^2 + 4) dx$.

Чернова