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 защипе 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 защипват в края на работата.
- Времето за работа по изпита е 3 часа.

Изпитната комисия ви пожелава успешна работа!

Задача 1. а) Имената на служители в дадена компания и на техните преки ръководители можем да представим с двумерен масив `const char* leaders[][2]`.

Например:

Служител	Ръководител
Иван Иванов	Мария Иванова
Мария Иванова	Иван Драганов
Иван Драганов	Стоян Петров

Казваме, че служителят *A* е ръководител на служителя *B*, ако *A* е пряк ръководител на *B* или е пряк ръководител на ръководител на *B*. Да се дефинира рекурсивна функция:

```
bool is_subordinate (const char* employee,
                     const char* manager,
                     const char* leaders[][2],
                     size_t n),
```

която проверява дали служителят с име `employee` е подчинен на служителя с име `manager` в компанията, описана с масива `leaders` с `n` реда.

б) Да се дефинира функция

`const char* the_big_boss(const char* leaders[][2], size_t n)`,

намираща името на служителя, който се намира най-високо в йерархията на компанията, описана с масива `leaders` с `n` реда.

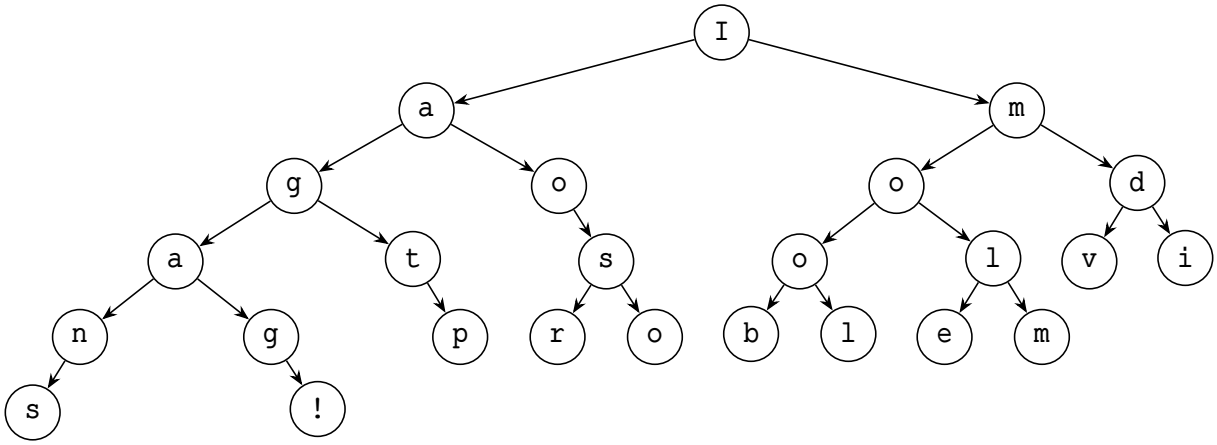
Приемаме, че йерархията от служителите, описана в `leaders` е коректна, т.е. всеки служител има по точно един пряк ръководител, с изключение на точно един служител, който няма пряк ръководител, и няма двама служители такива, че всеки от тях е ръководител на другия.

Да се демонстрира извикването на функциите с кратка програма.

Задача 2. Разглеждаме **двоично дърво** с данни от тип символ и елементи, описвани от следната структура:

```
struct Node {
    char text;
    Node *left, *right;
};
```

Стойностите са малки и главни латински букви, цифри и препинателни знаци. Няма празни символи (интервал, нов ред, табулация) и символи от разширената ASCII таблица (с код по-голям от 127). Пример за такова дърво е:



Да се реализира функция `printText`, която получава като аргумент указател към корен на такова дърво и извежда на стандартния изход текста, записан във върховете, спазвайки следните правила:

- съдържанието на всяко ниво (елементи с еднаква дълбочина) се извежда на отделен ред;
- ако на дадено място в нивото липсва възел, да се изведе интервал;
- дървото се извежда до последния елемент от последното ниво.

За показаното като пример дърво трябва да се изведе следния текст:

```
|I|
|am|
|good|
|at␣solvi|
|ng␣p␣␣roblem␣␣␣␣|
|s␣␣!|
```

Забележка: за яснота на примера в началото и края на всеки ред е добавен символът pipe `|`, който `printText` не трябва да извежда.

Да се демонстрира работата на тази функция в кратка програма.

Забележка: позволено е използването на класовете контейнери от стандартната библиотека STL.

Задача 3. Часовник за изкачване има 2 режима: Time и Altimeter. В режим Time, натискането на бутона за превключване на режим (Mode Switch), кара часовника да влезе в режим Altimeter. При повторно натискане на бутона за превключване на режим, часовникът се връща режим Time. При натискане на бутон за настройки (Set), когато часовникът е в режим Time, то той влиза в режим Set Hrs, при който часовете (Hrs) могат да се увеличават с единица при всяко натискане на бутона за настройки (Set). Ако се натисне бутонът за превключване на режим (Mode Switch), докато часовникът е в режим Set Hrs, то той преминава в режим Set Mins, при който минутите (Mins) могат да се увеличават с единица при всяко натискане на бутона за настройки (Set). Ако се натисне бутонът за превключване на режим (Mode Switch), докато часовникът е в режим Set Mins, то той преминава в режим Time.

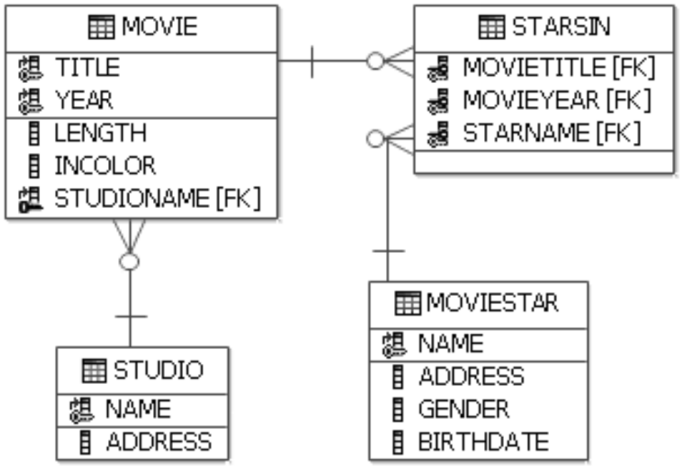
а) Да се конструира диаграма на преходите между състоянията.

б) Да се конструира таблица на преходите между състоянията.

в) Да се дефинират тестови сценарии, които покриват всички преходи, като се приеме, че часовникът е в първоначален режим Time.

г) Да се даде пример за негативен тестове сценарий.

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



Таблицата `Studio` съдържа информация за филмови студиа:

- `name` — име, първичен ключ
- `address` — адрес;

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`;

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 `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`.

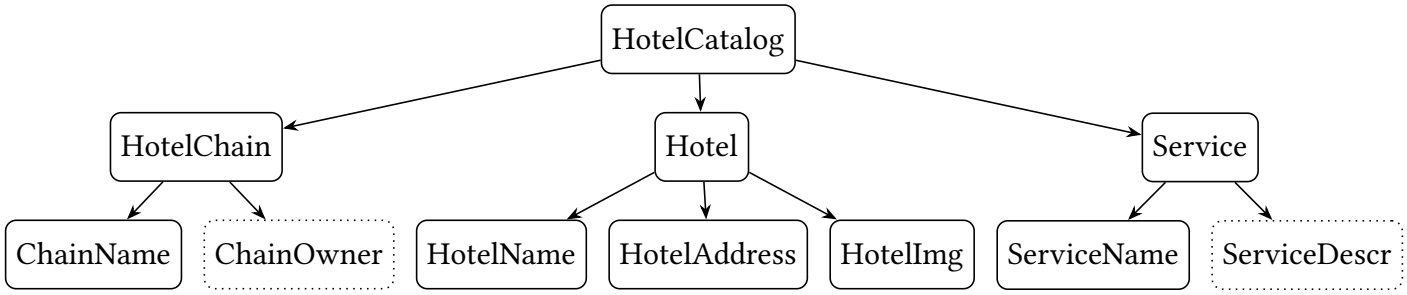
Зад 1. Да се напише заявка, която извежда имената и адресите на всички студиа, които имат поне един цветен и поне един черно-бял филм. Резултатът да се сортира възходящо по адрес.

Зад 2. Да се напише заявка, която за всяко студио с най-много три филма извежда:

- името му;
- адреса;
- средната дължина на филмите на това студио.

Студиа без филми също да се изведат (за средна дължина да се извежда `null` или 0).

Задача 5. За дадената по-долу схема да се създаде DTD документен тип и валиден спрямо този тип XML документ. XML документът трябва да съдържа поне две хотелски вериги, поне два хотела и поне три услуги, като всеки хотел предлага поне една услуга. DTD граматиката и XML документът трябва да отговарят на следните условия:



1. Елементът **HotelCatalog** има задължителен атрибут с име Version;
2. Под-елементите на **HotelCatalog** имат следния ред на подреждане: **HotelChain**, **Hotel**, **Service**;
3. Елементите **HotelChain** и **Service** могат да се срещат нула или много пъти, а елементът **Hotel** – един или много пъти; Под-елементите на всеки един от елементите **HotelChain**, **Hotel** и **Service** са задължителни в посочения ред отляво надясно, с изключение на под-елементите **ChainOwner** и **ServiceDescr**, които са опционални. Всички те са с текстово съдържание с изключение на под-елемента **HotelImg**, който е графично изображение във формат JPG и трябва да се представи като XML ENTITY;
4. Елементът **Service** може да съдържа опционален атрибут с име Included, имащ стойност **YES** или **NO**;
5. Елементът **Hotel** може да съдържа опционален атрибут, реферирящ към една хотелска верига **HotelChain**;
6. Елементът **Hotel** може да съдържа опционален атрибут, реферирящ към една или повече услуги **Service**.

Задача 6. За всеки два езика $L, M \subseteq \{0,1\}^*$ означаваме

$$L^M = \{w \in L^{|u|} \mid u \in M\}.$$

Вярно ли е, че винаги, когато L и M са регулярни, то и L^M е регулярен език? Отговорът да се обоснове.

Задача 7. На всеки опит хвърляме три пъти последователно зар. Дефинираме събитие

A = {точките при някое хвърляне са равни на сумата от точките на другите две хвърляния}.

а) Да се определи вероятността на A при извършване на един опит.

б) Извършваме опити докато събитието A се изпълни. Нека X е броят на хвърлянията на зар, които сме направили при това. Да се намери математическото очакване $\mathrm{E}X$ и дисперсията $\mathrm{D}X$.

в) Колко опита трябва да бъдат направени, така че да е по-вероятно събитието A да се сбъдне поне веднъж, отколкото да не се сбъдне нито веднъж?

Чернова