

СОФИЙСКИ УНИВЕРСИТЕТ
„СВ. КЛИМЕНТ ОХРИДСКИ“ФАКУЛТЕТ ПО МАТЕМАТИКА
И ИНФОРМАТИКАДЪРЖАВЕН ИЗПИТ
ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР ПО СОФТУЕРНО ИНЖЕНЕРСТВО”

ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)

10.09.2018 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 на един от езиците C или C++.

Професор X забравя постоянно паролата за своя таен сейф, която представлява последователност от цели числа. За да не е нужно да я помни, той скрива подсказки в кабинета си. На всеки ред от библиотеката му са подредени книги, най-много 20 реда книги, с до 30 книги на ред. Книгите на някои от редовете са подредени по азбучен ред на заглавията си, всяко от които е до 100 символа.

Паролата за сейфа на професор X се определя от числата, които задават последователните дължини на думите в заглавията на книгите, разположени точно в средата на редовете, в които книгите са подредени в азбучен ред. Ако на реда има четен брой книги, за паролата се използва книгата, намираща се по-близо до началото на реда. Думите в заглавията на книгите са разделени от точно един интервал. Дължините на думите формират паролата в реда, в който се срещат, от най-горния към най-долния ред на библиотеката.

Библиотеката на X може да се представи като двумерен масив **a** от низове с **m** реда по **n** низа всеки, представящи заглавията на книгите. Да се дефинира функция **revealPassword**, която по подадени **a**, **m** и **n**, извежда на стандартния изход паролата на професор X като последователност от числа, разделени с по един интервал. Да се демонстрира употребата на функцията **revealPassword** в кратка програма.

Пример:

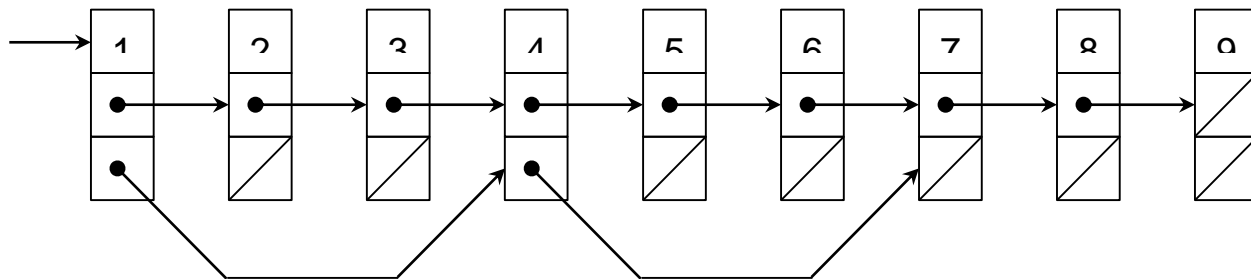
Нека библиотеката на професор X се състои от 3 реда с по 3 книги всеки както следва:

Алгебра	<u>Аналитична геометрия</u>	Математически анализ
Увод в програмирането	Обектно-ориентирано програмиране	Структури от данни и програмиране
Бази от данни	<u>Изкуствен интелект</u>	Функционално програмиране

Тогава паролата на професор X е поредицата от числа 10 9 9 8, получена от дължините на подчертаните думи. Думите “Обектно-ориентирано програмиране” не участват в паролата, защото книгите на втория ред не са подредени по азбучен ред на заглавията си.

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

Нека разгледаме следната структура от данни за бързо търсене, съхраняваща сортирана редица от цели числа:



Нека имаме списък с n елемента и нека с k означим горната цяла част на числото \sqrt{n} . Всеки възел на списъка съдържа цяло число и два указателя. Първият указател винаги сочи следващия елемент (ако има такъв). За възлите, намиращи се на индекс, кратен на k , вторият указател сочи възела, намиращ се на k позиции напред (ако има такъв). В горния пример $n = 9$, $k = 3$.

а) Да се напише функция **readList**, която по подаден път до текстов файл прочита от файла сортирана редица от цели числа и конструира списък от описания по-горе тип, съдържащ прочетените числа. Числата са записани във файла на един ред и са разделени с интервали.

б) Да се напише булева функция **member**, която по даден списък от описания тип и дадено цяло число проверява дали това число се съдържа в списъка. Функцията да реализира ефективен алгоритъм за търсене, който се възползва от особеностите на структурата, за да минимизира броя на обходените елементи.

Пример: ако в списъка от диаграмата по-горе търсим числото 6, при ефективния алгоритъм за търсене ще бъдат последователно обходени следните елементи: 1, 4, 7, 5, 6 и функцията **member** ще върне резултат "истина".

За реализацията на гореописаната структура от данни и за функциите **readList** и **member** не е позволено използване на библиотечни структури от данни и алгоритми, но е позволено използването на стандартните функции за работа с файлове.

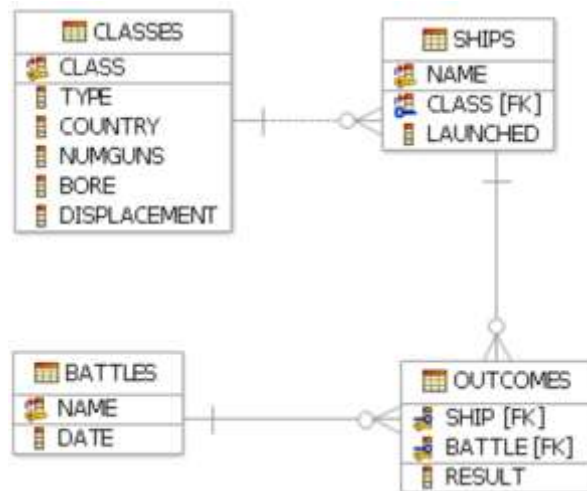
Задача 3. Дадено е следното квадратно уравнение:

$$ax^2 + bx + c = 0$$

Да се опише с псевдо-код намирането на броя на корените на уравнението. Да се конструира модел с граф на данновия поток, от който да се дефинират тестови сценарии за структурно тестване (тестване по метода на бялата кутия). Да се опишат основните стъпки при конструирание на модела.

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 клас, първичен ключ;
- **type** – тип ('bb' за бойни кораби, 'bc' за бойни крайцери);
- **country** – държава, която строи такива кораби;
- **numguns** – брой 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 клас, външен ключ към *Classes.class*;
- **launched** – година, в която корабът е пуснат на вода, може да приема стойност null.

Таблицата *Battles* съхранява информация за битките:

- **name** – име на битка, 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 кораб, външен ключ към *Ships.name*;
- **battle** – име на битка, външен ключ към *Battles.name*;
- **result** – резултат (потънал – 'sunk', повреден – 'damaged', победил – 'ok').

Забележка за всички таблици: За всички атрибути, за които не е указано, че могат да приемат стойност null, да се счита, че съществува not null ограничение.

1. Да се напише заявка, която извежда име на клас, годината на първата битка, в която кораб на този клас е участвал, годината на последната битка, в която кораб на този клас е участвал, и броя на всички различни битки, в които кораби на този клас са участвали, само за тези класове, започващи с буквата N. Ако за даден клас няма кораб, който да е участвал в битка, за съответните години да се върне стойност null.
2. Да се напише заявка, която да изведе имената на тези битки, за които броят на корабите от тип 'bb', участвали в тази битка, е по-голям от броя на корабите от тип 'bc', участвали в същата битка. Битки, в които не е участвал нито един кораб, да не се извеждат в резултата.

Задача 5. Семафорът е обект за синхронизация, чиито локални 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.

Разглеждаме псевдокод на два процеса, `P` и `Q`, които се изпълняват паралелно:

Process P:

`p_1`

`p_2`

Process Q:

`q_1`

`q_2`

`p_1`, `p_2` и съответно `q_1`, `q_2` са инструкции, които двата процеса изпълняват последователно.

Да се модифицира псевдокодът на `P` и `Q` така, че да се осигури чрез семафори синхронизация инструкция `p_1` да се изпълни преди `q_2`, а `q_1` да се изпълни преди `p_2`. Приемете, че инициализацията на семафорите се прави от родителски процес по следния начин:

Parent process:

`semaphore t1,t2`

`t1.init(0)`

`t2.init(0)`

След инициализиране на семафорите, родителският процес поражда паралелните процеси `P` и `Q`.

Решението дайте под формата на модифицирания псевдокод на `P` и `Q`, като позволените оператори за работа със семафори са според даденото описание на семафор в началото на задачата. Приемете, че `P` и `Q` имат достъп до обектите `t1` и `t2`.

Process P:

Process Q:

Задача 6. Да се направи декомпозиция на модулите от архитектурата на софтуерна система за управление на роботизирана мрежа, извършваща доставки в складово помещение с особено големи размери, според дадените по-долу изисквания. Да се обоснове защо така проектираната архитектура удовлетворява изискванията.

1. Системата е съставена от централен сървър и полеви части, които се изпълняват на движещите се работи.
2. Централният сървър е свързан със съществуваща онлайн система за електронна търговия, от която се правят заявките за доставяне на стоки от склада.
3. Системата следи и управлява движението на работи, които пренасят стоки от стелажите до приемо-предавателния пункт в склада.
4. Всеки робот се движи като следи специална маркировка по пода на склада. Движението на робота се управлява от вградения в него софтуер.
5. Централният сървър трябва да задава начална и крайна точка за движението на робота, като самият робот избира оптималния маршрут.
6. Броят на движещите се в даден момент работи се определя от количеството и големината на заявките в склада.
7. Системата трябва да следи за възникване на критични ситуации (например претоварване или излизане на робот от строя, сблъсък на работи и т. н.). Ако възникне такъв случай, да се вземат съответните мерки (известяване на оператор), като това да стане максимално бързо.
8. Системата да не позволява достъп през интернет до централния сървър, което може да доведе до смущения в движението на роботите в склада.
9. Системата да е 100% надеждна, тъй като евентуален срив може да доведе до спиране на доставките до приемо-предавателния пункт.
10. Да не се допуска вмешателство в движението на роботите през интернет.

Задача 7. Софтуерна система цели да подпомогне земеделците при отглеждането на различни растителни култури. Системата предоставя полезна информация за конкретни култури при наличието на специфични условия като климат, почви, води. Дадени са следните изисквания:

- R1. Нерегистриран потребител (гост) може да разглежда само обща информация за услугите, предоставени от системата.
- R2. Потребител може да се регистрира в самата система или като използва Facebook акаунт.
- R3. Регистриран потребител може да търси определени растения по име.
- R4. Регистриран потребител може също така и да използва разширено търсене по специфични условия като климат, почви, води и др.
- R5. Регистриран потребител може да добавя и редактира информация за отглеждането на дадена култура, като може да добави и съответната локация чрез Google Maps.
- R6. Преди да се публикува информацията за отглеждането на дадена култура, тя се преглежда и одобрява от експерт в областта, който е регистриран в системата.
- R7. Регистриран потребител може да поиска платена консултация от експерт, като заплащането може да стане по банков път, чрез банкова карта или чрез системата ePay.

Да се състави диаграма на потребителските случаи (Use Case Diagram) по така зададените изисквания. Да се обяснят връзките между потребителските случаи и да се обоснове тяхното конкретно използване в диаграмата.

Задача 8. Пресметнете интеграла $\int_0^1 x^2 \operatorname{arctg} x \, dx$.

Чернова