

СОФИЙСКИ УНИВЕРСИТЕТ  
„СВ. КЛИМЕНТ ОХРИДСКИ“ФАКУЛТЕТ ПО МАТЕМАТИКА  
И ИНФОРМАТИКА**ДЪРЖАВЕН ИЗПИТ**  
**ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР ПО СОФТУЕРНО ИНЖЕНЕРСТВО”****ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)**  
**13.07.2018 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 C/C++.

Да се дефинира функция **sortLex**, която получава като аргументи положителното число **n** и масив **a**, съдържащ **n** на брой цели неотрицателни числа, и ги сортира във възходящ ред относно лексикографската наредба (например, 123 е преди 9 по лексикографската наредба). Да се напише кратка програма, в която да се демонстрира използването на функцията **sortLex**.

За реализацията на функцията **sortLex** не е позволено използването на стандартни библиотечни функции.

**Пример:**

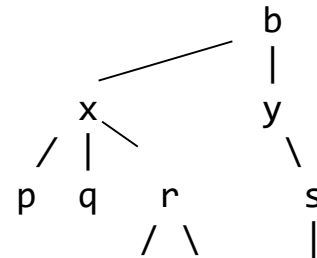
При подаден масив {13,14,7,2018,9,0}, след изпълнение на функцията **sortLex** масивът ще бъде подреден по следния начин: {0,13,14,2018,7,9}.

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

Троично дърво от латински букви ще наричаме следната структура:

- Празното дърво е троично дърво от латински букви;
- Ако  $T_1$ ,  $T_2$  и  $T_3$  са троични дървета от латински букви, а  $x$  е латинска буква, то наредената четворка  $\langle x, T_1, T_2, T_3 \rangle$  също е троично дърво от латински букви.

а) Да се напише функция **readLast**, която по дадено троично дърво от латински букви намира думата, която може да се прочете отляво надясно в последното му ниво. Ниво на троично дърво от латински букви наричаме списък от възли в дървото, които са равноотдалечени от корена.



**Пример:** за дървото вдясно функцията **readLast** трябва да връща думата “cat”.

Представянето на дървото е по Ваш избор. Опишете избраното от Вас представяне.

б) Да се напише функция **serialize**, която по дадени: (1) троично дърво от латински букви и (2) низ, описващ път до файл, записва дървото в текстов файл. Текстовото представяне на троично дърво да бъде следното:

- Текстовото представяне на празното дърво е звездичка (“\*”);
- Текстовото представяне на троичното дърво  $\langle x, T_1, T_2, T_3 \rangle$  е “(x T1 T2 T3)”, където  $T_1$ ,  $T_2$  и  $T_3$  са текстовите представяния съответно на  $T_1$ ,  $T_2$  и  $T_3$ .

**Пример:** Текстовото представяне на дървото горе е:

(b (x (p \* \* \*) (q \* \* \*) (r (c \* \* \*) \* (a \* \* \*))) (y \* \* (s \* (t \* \* \* \*) \*))) \*

За реализацията на функциите **readLast** и **serialize** е позволено използването на стандартните за съответния език библиотечни функции.

---

Задача 3. Оценка на студент за конкретна дисциплина се изчислява в точки при скала от 1 до 100 като се взимат предвид следните компоненти:

- 20% семинар
- 30% курсов проект
- 50% тест

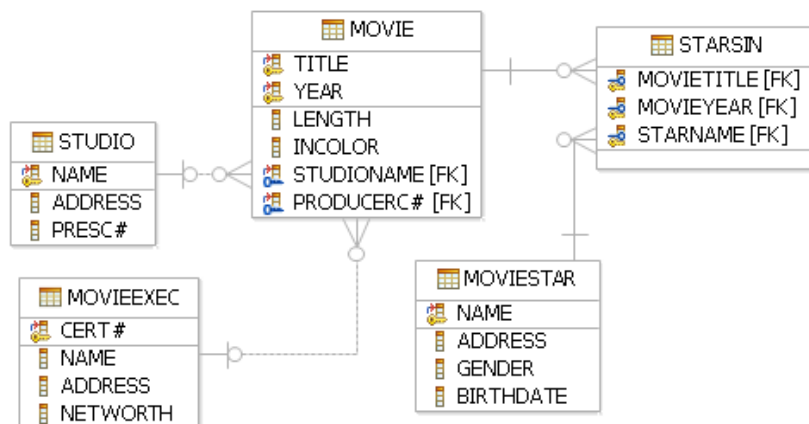
Определянето на крайната оценка въз основа на получените точки е по следната скала:

- $\leq 40$  – Слаб 2
- $> 40$  – Среден 3
- $> 55$  – Добър 4
- $> 70$  – Мн. добър 5
- $> 85$  – Отличен 6

Да се опише с псевдокод изчисляването на крайната оценка, като се обозначат броя на получените точки за всеки от трите компонента на оценката съответно променливите **a**, **b** и **c**. Да се конструира модел с граф на управляващия поток, от който да се дефинират тестови сценарии за структурно тестване (тестване по метода на бялата кутия). Да се опишат основните стъпки при конструиране на модела.

**Задача 4.** Дадена е базата от данни **Movies**, в която се съхранява информация за филми, филмови студиа, 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 **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**.

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

- **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**.

1. Да се напише заявка, която да изведе име на студио, годината на първия филм за това студио, годината на последния филм за това студио и броя на всички филми за това студио, само за тези студиа започващи с буквата M.
2. Да се напише заявка, която да изведе името на актрисата, участвала в най-много филми и броя на филмите, в които е участвала.

Задача 5. Файлът **f1** се намира в текущата директория и има следното съдържание:

**1 3 5 2 4 6**

Текстов файл с име **procA** съдържа зададената по-долу последователност от команди на bash за Linux. Напишете вдясно какво ще бъде изведено на стандартния изход и какво ще е съдържанието на файла **f2** след стартиране на **procA** с команден ред

**bash procA ab cd ef gh**

ако на стандартния вход бъде подадена последователността от символи **8 3**

```
var=3
for i
do for j in `cat f1`
do if test $j -lt $#
then echo $i $j
var=`expr $var \* $j`
echo $var $i > f2
else var=`expr $var + $j`
echo $var $j >> f2
fi
done
break
done
read s1 s2
until grep $s1 f2
do set $s2 $var $#
wc -c f2
shift
echo $1 $2
exit
cat f2
done
set `head -21 f2`
echo $*
shift
echo $2
```

Задача 6. Да се направи декомпозиция на модулите от архитектурата на софтуерна система за организиране на онлайн залагания на спортни събития, според дадените по-долу изисквания.

Обосновете защо така проектираната архитектура удовлетворява изискванията.

1. Клиенти на системата са различни организатори на залагания, които представляват юридически лица.
2. Залагания могат да се правят само от регистрирани потребители.
3. Нерегистрирани потребители могат да преглеждат списъка с налични събития за залагане и съответните предложения за коефициенти на залозите към тях.
4. Коефициентите на залозите могат да се променят за секунди, като се изчисляват по математическа зависимост, която се определя количеството и честотата на направените залози до момента, както и от прогнозни резултати.
5. Прогнозните резултати се взимат от външна услуга, която се достъпва по специфичен протокол.
6. Залаганията и печалбите не са в реални пари, а в т. нар. кредити, с които всеки потребител разполага в сметката си. Кредитите могат да се зареждат по различни начини, които се определят индивидуално за всеки клиент.
7. Потребителите могат да правят неограничен брой залагания, но само в рамките на текущо наличните им кредити.
8. Системата е непрекъснато свързана с онлайн регистър на данъчната служба в съответната държава, където е регистриран клиентът.
9. Системата трябва да е достъпна 24/7, тъй като потребителите могат да се включват от различни часови зони.
10. Системата трябва да е защитена от външна намеса.

---

Задача 7. Онлайн система подпомага идентифицирането на различни болести и вредители по растенията чрез разпознаване на изображение (снимка, направена от потребител) и предлага съответната информация за справяне с тях. Дадени са следните изисквания:

**R1.** Нерегистриран потребител (гост) може да търси информация за различни вредители и болести по растенията по зададено име на растението, тип на растението, вид вредител или болест.

**R2.** Потребител може да се регистрира в самата система или да използва Facebook регистрация.

**R3.** Регистриран потребител — експерт по растенията, може да качва информация в системата за вредители и болести по растенията, която задължително включва име и вид на растението, описание на вредителя или болестта и мерки за справяне. Може да се добави и снимков материал.

**R4.** Регистриран потребител може да качи в системата снимка на вредител или болест по дадено растение.

**R5.** При качване на снимка системата съответно локализира потребителя чрез Google Maps, разпознава изображението на снимката и показва информация за съответния вредител или болест.

**R6.** При качване на снимка в системата, тя автоматично се изпраща към експерт, който при преценка може да сигнализира съответните органи.

**R7.** Регистриран потребител може да поиска платена консултация с експерт, като може да заплати услугата чрез PayPal или с банкова карта.

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



---

Задача 8. Пресметнете интеграла  $\int_0^{\frac{\pi}{2}} \frac{dx}{2 + \sin x}$ .

13.07.2018

СУ-ФМИ

Държавен изпит за ОКС  
*Бакалавър*

**Софтуерно  
инженерство**

ф.н. \_\_\_\_\_

лист 10/10

---

**Чернова**