

СОФИЙСКИ УНИВЕРСИТЕТ
“СВ. КЛИМЕНТ ОХРИДСКИ”



ФАКУЛТЕТ ПО МАТЕМАТИКА
И ИНФОРМАТИКА

ДЪРЖАВЕН ИЗПИТ

ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР ПО ИНФОРМАЦИОННИ СИСТЕМИ”

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. Графично изображение е представено с матрица от **m** реда и **n** колони. Клетките на матрицата, във всяка от които е записана целочислена стойност от 0 до 255, съответстват на пикселите в графичното изображение (формат *grayscale*).

Всяка клетка в матрицата има до 8 съседа — до 4 по диагонал, до два, разположени хоризонтално, и до два – вертикално.

Област в изображението е непрекъсната последователност от съседни клетки с ненулеви стойности. Черните елементи, представени със стойност 0, се считат за контури на областите. Така, една област се определя от граница от нулеви елементи и границите на матрицата.

А) Дефинирайте функция, която получава като аргумент матрица от посочения вид и извежда на стандартния изход средната яркост на всяка от областите, сортирани в низходящ ред според яркостта. Средна яркост на дадената област се изчислява като средно-аритметично на стойностите на всички клетки (пиксели), образуващи областта. За всяка област изведете координатите на една произволна клетка от нея и средната яркост на областта.

Б) Дефинирайте функция `main`, която илюстрира извикване на функцията, дефинирана към т. А. Не се изисква да се реализира вход на данните.

Задачата да се реализира на езика C/C++. Позволява се използването на функции и класове от стандартната библиотека.

Пример: При входна матрица, **m** = 6, **n** = 6

170	0	0	255	221	0
68	0	17	0	0	68
221	0	238	136	0	255
0	0	85	0	136	238
238	17	0	68	0	255
85	170	0	221	17	0

Програмата ще изведе следните области и средни яркосты, сортирани в низходящ ред според яркостта:
(0, 0) 153; (0, 3) 151.11

Задача 2. Задачата да се реши на езика C++.

А) Разгледайте дадения по-долу фрагмент. Оградете всички твърдения за него, които са верни. Всяко вярно твърдение, което е маркирано, увеличава резултата за задачата. Всяко грешно твърдение, което е маркирано, го намалява.

```
#include <string>

// Представа предмет
class Item {
    // Етикет на предмета
    std::string label;
    virtual int hash() = 0;
};

// Представа молив
class Pencil : protected Item {};

// Представа кутия
class Box : public Item {

    // Сочи към това, което
    // се пази в кутията
    Item* contents;

public:
    // Съхранява something в кутията
    void store(Item* something) {
        contents = something;
    }
};
```

- А) Член-променливата label на класа Item е protected.
- Б) Един обект от тип Pencil може да се съхрани в себе си.
- В) Един обект от тип Box може да се съхрани сам в себе си.
- Г) Един обект от тип Pencil може да се съхрани в две кутии едновременно.
- Д) От класа Box НЕ МОГАТ да се създават обекти, но могат да се създават указатели (pointer) и псевдоними (reference).
- Е) Кодът за Pencil ще предизвика компилационна грешка, защото в тялото на класа не е дефинирано нищо.
- Ж) Един обект от тип Box може да се съхрани в друг обект от тип Box.
- З) Класът Pencil наследява виртуално класа Item.
- И) Всеки обект от тип Pencil има член label, който е protected.
- К) Класът Pencil е абстрактен.
- Л) Обекти от тип Item не могат да се създават, защото в класа не е дефиниран нито един конструктор.
- М) При създаването на обект от тип Box не се задава стойност за contents и тази променлива остава неинициализирана.

Б) Разгледайте дадения по-долу фрагмент. Някои от редовете на main() са номерирани в коментар. За всеки от тях посочете какво ще се изведе на стандартния изход след неговото изпълнение.

```
#include <iostream>

class base {
public:
    base()          { std::cout << "base::base()\n"; }
    ~base()         { std::cout << "base::~~base()\n"; }
    virtual void f() { std::cout << "base::f()\n"; };
};

class derived : public base {
public:
    derived()       { std::cout << "derived::derived()\n"; }
    ~derived()      { std::cout << "derived::~~derived()\n"; }
    virtual void f() { std::cout << "derived::f()\n"; }
};

int main()
{
    base* p = new derived(); //1
    p->f();    //2
    base obj = *p;
    obj.f();  //3
    delete p; //4
}
```

Ред //1 ще изведе:

Ред //2 ще изведе

Ред //3 ще изведе:

Ред //4 ще изведе

Задача 3. Решете задачата на езика *Haskell*.

А) Двоично дърво ще представяме чрез следния тип:

```
data Tree a = EmptyTree | Node {  
    value :: a,  
    left  :: Tree a,  
    right :: Tree a  
} deriving (Show,Read)
```

Довършете функцията `treeWords`:

- Опишете типа на функцията така, че тя да получава дърво от символи и да връща списък от символни низове.
- Функцията да връща списък от всички думи, които могат да се образуват по път от корена до някое от листата на дървото. Ако дървото е празно, да се връща празният списък.

```
treeWords :: _____  
  
treeWords _____ = []  
  
treeWords (Node v EmptyTree EmptyTree) = [[_____]]  
  
treeWords (Node v l r) = map (v:) (_____ ++ _____)  
  
    where wl = _____  
          wr = _____
```

Б) Довършете кода на функцията *quickSort* така, че тя да сортира списък по метода на бързото сортиране (*quicksort*). Опишете типа на функцията по такъв начин, че тя да може да работи със списък от произволен тип елементи, стига те да са от класа *Ord*.

```
quickSort :: _____  
  
quickSort [] = _____  
  
quickSort (x:xs) = _____ ++ _____ ++ _____  
  
    where  
        lesser = filter _____  
        greater = filter _____
```

Задача 4. Дадена е базата от данни **PC**. В нея се съхранява информация за три вида продукти – персонални компютри, лаптопи и принтери.

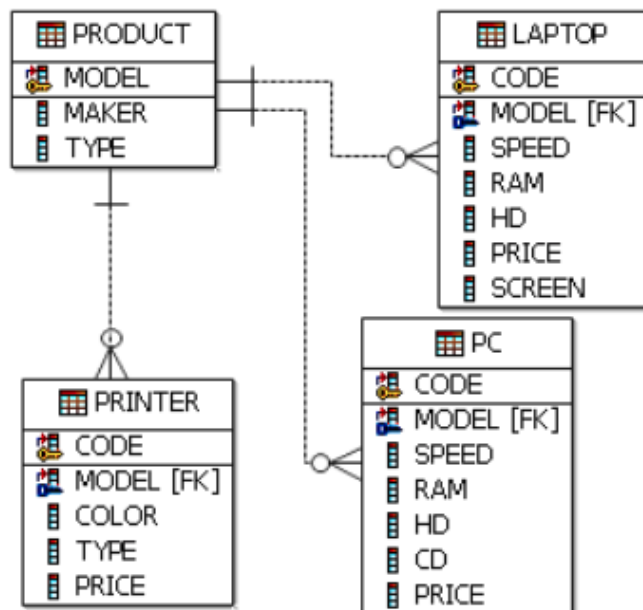
Таблицата **Product** съдържа базова информация за всеки продукт:

- model – модел на продукта, първичен ключ
- maker – производител на продукта
- type – един от следните типове: 'PC', 'Laptop' или 'Printer'

Таблицата **PC** съдържа специфична информация за компютрите:

- code – уникален идентификатор на дадена компютърна конфигурация, първичен ключ
- model – модел на компютъра, външен ключ към **Product.model**. Може да имаме няколко различни компютърни конфигурации от един и същ модел, но с различни параметри
- speed – тактова честота на процесора в MHz
- ram – количество RAM памет в MB
- hd – размер на твърдия диск в GB
- cd – скорост на CD устройството
- price – цена на компютъра

Таблицата **Laptop** съдържа специфична информация за лаптопите. Атрибутите са аналогични на тези на **PC**, но липсва атрибутът **CD** и има атрибут за размера на екрана.



Таблицата **Printer** съдържа информация за принтерите:

- code, model, price – аналогични на едноименните атрибути в **PC**
- color – 'y' за цветен принтер, 'n' за черно-бял
- type – тип на принтера: 'Laser', 'Jet', 'Matrix'

Задача 1: Посочете заявката, която извежда кодовете и цените на всички лаптопи, чийто екран е с диагонал между 13 и 15 инча включително и за които съществува поне един персонален компютър със същото количество RAM памет:

а) `SELECT code, price FROM Laptop
WHERE screen BETWEEN 13 AND 15
AND ram IN (SELECT ram
FROM PC);`

в) `SELECT code, price FROM Laptop
WHERE screen BETWEEN 13 AND 15
HAVING COUNT(SELECT ram
FROM PC) >= 1;`

б) `SELECT code, price FROM Laptop
WHERE screen >= 13
AND screen <= 15
AND ram = (SELECT ram
FROM PC);`

г) `SELECT code, price FROM Laptop
WHERE EXISTS (SELECT *
FROM PC
WHERE ram = L.ram)`

Задача 2: За всеки производител да се изведе името и броят 15-инчови лаптопи. Ако даден производител няма никакви лаптопи или има, но нито един от тях не е 15-инчов, срещу името му да се изведе числото 0:

а) `SELECT maker, COUNT(screen = 15)
FROM Product
NATURAL JOIN Laptop
GROUP BY maker;`

в) `SELECT maker, COUNT(code) FROM Laptop
RIGHT JOIN Product
ON Laptop.model = Product.model
AND screen = 15
GROUP BY maker;`

б) `SELECT maker, COUNT(code) AS laptops
FROM Product
LEFT JOIN Laptop ON Product.model = Laptop.model
HAVING screen = 15
GROUP BY maker;`

г) `SELECT maker, COUNT(DISTINCT code)
FROM Product
LEFT OUTER JOIN Laptop
ON Product.model = Laptop.model
GROUP BY maker;
HAVING screen = 15;`

Задача 5. Информационна система съхранява информация за компания, която предлага автомобили под наем. Съхранява се следната информация за клиентите на компанията:

- телефонен номер на клиент — низ от 12 символа, уникален за всеки клиент
- име на клиент — низ до 50 символа
- e-mail — низ до 30 символа.

Съхранява се информация и за офисите на компанията, в които клиентите могат да наемат автомобил:

- код на офиса — низ от 5 символа, уникален за всеки офис
- име на офиса — низ до 20 символа
- адрес на офиса — низ до 50 символа.

В офисите работят служители на компанията. За тях се пази информация за:

- служебен номер на служител — низ от 5 символа, уникален за всеки служител
- име — низ от 30 символа
- телефонен номер — низ от 12 символа.

В един офис могат да работят много служители, но всеки служител работи само в един офис. Компанията предлага автомобили под наем. За автомобили се пази информация за:

- регистрационен номер на автомобила — низ от 8 символа, уникален за всеки автомобил
- марка — низ до 15 символа
- модел — низ до 15 символа
- цена под наем — число с плаваща запетая, по-голямо от нула.

В един офис може да има много автомобили под наем, но един автомобил се намира в и предлага от един офис. Един клиент може да наема във времето много автомобили и един автомобил може да бъде наеман във времето от много клиенти. В системата се пази и информация за:

- датата, на която клиентът е наел автопмобила
- датата, до която трябва да го върне
- внесена от клиента предплата (може да бъде и NULL, ако няма такава).

Задание:

- А) Да се направи E/R модел на БД, която съхранява гореописаната информация. Начертайте E/R диаграма на модела.
 - Б) Преобразувайте E/R диаграмата към релационни схеми. Премахнете излишествата, където това е възможно.
 - В) Напишете DDL код, съответстващ на релационните схеми. Реализирайте всички описани ограничения.
-

Задача 6. Разглеждаме информационната система за даване на автомобили под наем, представена в **задача 5** (E/R моделиране).

- Клиент може да резервира автомобил чрез сайта на компанията, по телефона или на място в офис.
- Резервират само регистрирани в системата клиенти.
- При резервиране се предплаща минимум половината от наема, а при получаване на автомобила се предплаща остатък от наема (ако има такъв) и се предоставя гаранционен депозит.
- Пълният размер на депозита се освобождава при връщане на автомобила в изискуемото състояние.
- До 72 часа преди началото на наемния период, клиентът може да откаже или промени резервацията, без начисляване на такси.
- При промяна на избрания автомобил, свързани с него услуги или наемния период, се преизчислява наема.
- При неявяване за получаване на автомобила, без клиентът да е отказал резервацията, предплатеният наем се задържа.
- След началото на наемния период, клиентът може да удължи срока за ползване на автомобила до 48 часа преди изтичането на предварително договорения срок.
- Удължаването може да се извърши след потвърждение, че графикът на автомобила го позволява и след извършено пълно плащане на цената за удължаването.
- Всички плащания се извършват с банкова карта.

Задание:

- А) Подгответе диаграма на потребителските случаи в термините на UML стандарта. На диаграмата да се представят минимум: актьори; потребителските случаи, които се асоциират с клиента (същите могат да са асоциирани и с други актьори); отношения между потребителските случаи.
- Б) Опишете в пълен формат потребителски случай за резервиране на автомобил. В описанието да се включат минимум: актьори; предусловия; следусловия (резултати); основен сценарий; отношения с други потребителски случаи (ако е приложимо); алтернативен сценарий; какви проверки за коректност се правят при обработката; специфични за потребителския случай допълнителни изисквания.

Ако при изпълнението на заданието са направени допускания във връзка с представения сценарий, те трябва да бъдат описани явно.

Задача 7. Намерете определения интеграл

$$\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} e^{|x|} (\sin x + \cos x) dx.$$

Чернова