

СОФИЙСКИ УНИВЕРСИТЕТ
“СВ. КЛИМЕНТ ОХРИДСКИ”



ФАКУЛТЕТ ПО МАТЕМАТИКА
И ИНФОРМАТИКА

ДЪРЖАВЕН ИЗПИТ

ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР ПО КОМПЮТЪРНИ НАУКИ”

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 яркости, сортирани в низходящ ред според яркостта:
(0, 0) 153; (0, 3) 151.11

Задача 2. Разредена матрица представлява матрица с голяма размерност, за която малък брой от елементите са ненулеви(*).

- А) Опишете с думи подходяща структура от данни, в която да съхранявате такава матрица.
- Б) Напишете функция, която приема като аргументи две разредени матрици от реални числа, представени по описания от вас начин, и връща като резултат тяхната сума, отново представена по този начин.
- В) Демонстрирайте тази функция в кратка програма.

(*)За матрица с голяма размерност приемаме, например, матрица с размерност $10^6 \times 10^6$ и брой на ненулевите елементи от порядъка на 10^4 .

За задачката се предполага използването на езика C или C++. Разрешава се използването на стандартните за езика библиотеки, като се изисква да се посочи кой език и съответно коя негова версия е използвана.

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.

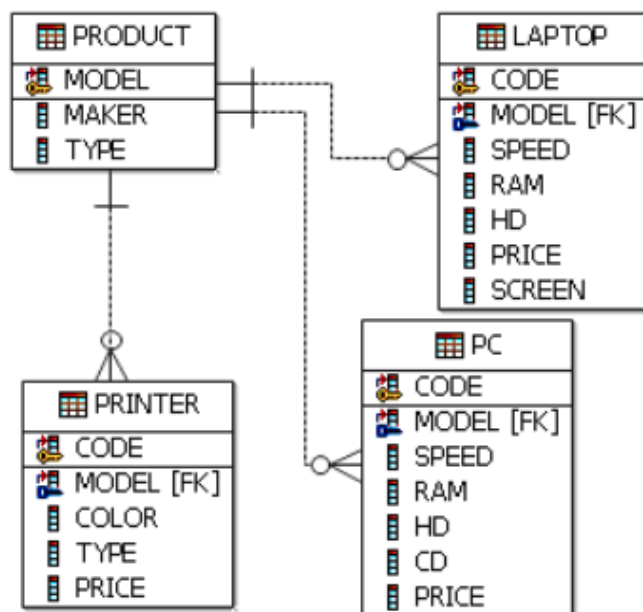
Таблицата **Product** съдържа базова информация за всеки продукт:

- 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. Даден е масив $A[1, 2, \dots, n]$ от цели числа. Предложете алгоритъм със сложност по време $O(n)$, който намира наредена двойка от индекси (i, j) , такива че $1 \leq i \leq j \leq n$ и сумата

$$\sum_{k=i}^j A[k]$$

е максимална. Не е необходимо да давате псевдокод, но описанието на алгоритъма трябва да е абсолютно ясно и недвусмислено. Накратко обосновайте коректността му и сложността му по време.

*Упътване: има решение, изградено по схемата **Динамично програмиране**. Може да започнете да разсъждавате така: всеки максимален като сума на елементите подмасив има някакъв най-десен елемент.*

Задача 6. Говорим за обикновени, неориентирани графи без примки. Нека $G = (V, E)$ е граф. G се нарича *самодуален*, ако е изоморфен на своето допълнение \overline{G} .

Нека G е самодуален. Докажете или опровергайте, че G е свързан.

Задача 7. В равнината е въведена декартова координатна система Oxy . Правите $g : y - 1 = 0$ и $h : 2x - y + 5 = 0$ са страни на един триъгълник, чиято трета страна минава през точката $U(2, 0)$. Известно е още, че една от медианите на този триъгълник лежи върху правата $m : x - 2y + 4 = 0$. Да се намерят координатите на върховете на триъгълника, както и неговото лице.

Чернова