



**СОФИЙСКИ УНИВЕРСИТЕТ
“СВ. КЛИМЕНТ ОХРИДСКИ”**

**ФАКУЛТЕТ ПО МАТЕМАТИКА
И ИНФОРМАТИКА**

**ДЪРЖАВЕН ИЗПИТ
ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР
ПО КОМПЮТЪРНИ НАУКИ”**

**ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)
15.07.2009 г.**

Време за работа – 3 часа

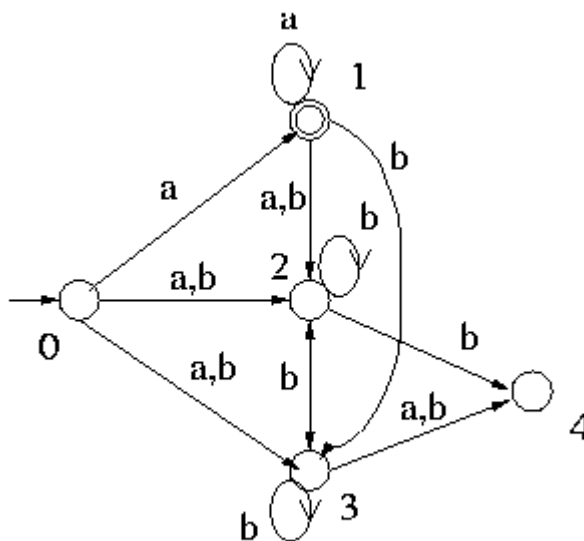
Драги абсолвенти,

Попълнете факултетния си номер на всички страници!

Решението на всяка от задачите се разполага само в мястото от края на условието на тази задача до началото на условието на следващата задача. Могат да се използват и двете страни на листата.

Изпитната комисия ви пожелава успешна работа.

Задача 1. (15 точки) Нека $A = (Q, \Sigma, \delta, s, F)$ е крайният недетерминиран автомат, представен чрез следния граф:



А) Да се напишат стойностите на символите Q, Σ, s, F и δ от дефиницията на A .

Б) Да се дефинира краен детерминиран автомат B , еквивалентен на автомата A . Обосновете отговора си.

Задача 2. (15 точки) В текущия каталог има текстов файл fileA.txt със следното съдържание

```
xxxxxxxxxx
abcdefgh
012345678
```

При успешно изпълнение на файла, получен при успешна компилация на задания по-долу програмен код на C, в който са използвани системни примитиви на ОС UNIX и LINUX:

А) Означете със стрелка края на всяка линия, в която се извежда на стандартния изход (терминала). Всяка стрелка да е свързана с пореден номер, съответстващ на реда на извеждането.

```
#include <stdio.h>
#include <fcntl.h>
main()
{
    int fd, n_byt, i = 0, status ;
    char sline [ 40 ], c ;

    if ( fork( ) )
    {
        wait ( &status);
        execlp ("echo", "echo", "End_of_program",0);
    }
    else{
        if ( ( fd = open ("fileA.txt",O_RDONLY ) ) == -1 )
        { printf ("\n Cannot open \n"); exit (1); }

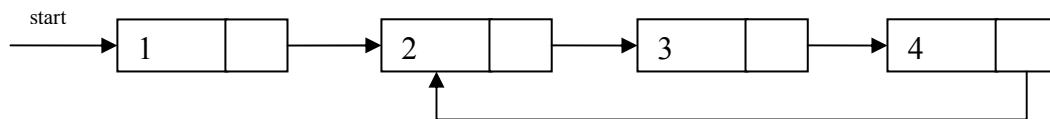
        n_byt = read ( fd , sline, 20 );
        c = sline[ i++];
        if ( c >= 'a' && c <= 'z' )
        {
            while ( sline [ i ++] != '\n' && i < n_byt )
                write ( 1, "$", 1 );
            write ( 1, "\n", 1 );
        }
        while ( i < n_byt ) write (1,&sline[i++],1);
        close ( fd );
        execlp ("wc", "wc", "-l", "fileA.txt",0 );
    }
}
```

Б) Какво се извежда на стандартния изход (терминала)?

Задача 3. (10 точки) Да се напише програма, която въвежда от клавиатурата правоъгълна матрица от реални числа с 5 реда и 10 стълба. Програмата да извежда на екрана текста "yes" в случай, че въведената матрицата има поне два различни линейно зависими реда и "no" в противен случай.

Задача 4. (15 точки) Да се дефинира тип данни, описващ възел в линеен едносвързан списък от цели числа. Да се напише булева функция, която проверява дали в даден едносвързан списък съществува цикъл. Цикъл наричаме такава редица от указатели към съседни възли, в която поне един указател се среща повече от веднъж.

Пример: В списъка на фиг. 1 съществува цикъл, а в списъка на фигура 2 – не съществува.



Фиг. 1



Фиг. 2

Задача 5. (10 точки) Да се попълни в празните полета изходът от предшестващите ги програмни конструкции. Да се обозначи типът на свързването с адреси на методи – (д)инамично или (с)татично.

<pre> class A { public: A() { cout << "A::A()\n"; g(); <input type="checkbox"/> } void operator=(const A&) { cout << "A::=\n"; } void f() { cout << "A::f()\n"; g(); <input type="checkbox"/> } virtual void g () { cout << "A::g()\n"; } virtual ~A() { cout << "A::~~\n"; } }; class B : public A { public: B() { cout << "B::B()\n"; } void f() { cout << "B::f()\n"; } void g() { cout << "B::g()\n"; } ~B() { cout << "B::~~\n"; } }; </pre>	<pre> void main() { cout << "1:\n"; B* bp = new B; <input type="text"/> cout << "2:\n"; A* bp_a = bp; bp_a->f(); <input type="checkbox"/> <input type="text"/> cout << "3:\n"; bp_a->g(); <input type="checkbox"/> <input type="text"/> cout << "4:\n"; A a; a = *bp; <input type="text"/> cout << "5:\n"; B b; b = *bp; <input type="text"/> cout << "6:\n"; delete bp_a; <input type="checkbox"/> <input type="text"/> cout << "7:\n"; <input type="text"/> }; </pre>
--	---

Задача 6. (15 точки) Да се реализира абстрактен клас **Function**, представящ целочислена функция на една променлива. Да се реализират негови наследници **LinearFunction**, **CharacteristicFunction** и **Superposition**, представящи съответно линейна функция, характеристична функция на крайно множество от цели числа и суперпозиция на две функции, представени чрез обекти от тип **Function**.

Линейната функция от вида $f(x) = ax + b$ се задава чрез коефициентите a и b . Характеристичната функция на дадено множество е такава функция, която за елементите на множеството приема стойност 1, а за всички останали стойности на аргумента – 0. Задава се чрез самото множество, представено чрез динамичен масив. Суперпозиция на две функции f и g е такава функция $h = f.g$, за която $h(x) = f(g(x))$ за всяко x и се задава се чрез (указатели към) два обекта от клас **Function**, представящи f и g .

За класовете да се реализират подходящи конструктори, метод за изчисляване на стойността на функцията и ако е необходимо – деструктори.

Задача 7. (12 точки) Компресирано представяне на даден списък от стойности наричаме такъв списък от точкови двойки, за който неколкото последователно срещане на един елемент е заменено от точкова двойка, първият елемент на която показва повторената стойност, а вторият – броя на повторенията. Например, некомпесираният списък **(2 2 2 Stan 7 7 2)** след компресия приема вида **((2 . 3) (Stan . 1) (7 . 2) (2 . 1))**. Да се реализират на езика Scheme двойка функции, които преобразуват некомпесиран списък в компресиран и обратно.

Задача 8. (15 точки) Дадена е следната дефиниция на предикат $p(N, L)$, който по дадено естествено число $N > 0$ връща в L списък от всички делители на числото N .

```
p(N, L):- help(N, 1, [], L).
```

```
help(N, N, R, [N|R]):- !.
```

```
help(N, C, R, L):- C < N, N mod C == 0, !, C1 is C+1,  
                  help(N, C1, [C|R], L).
```

```
help(N, C, R, L):- C < N, C1 is C+1, help(N, C1, R, L).
```

А) Да се дефинира предикат $is_prime(N)$, който по дадено естествено число $N > 0$ проверява дали N е просто число. Може да се използва предикатът $p(N, L)$.

Примери: `?-is_prime(1).` No.

`?-is_prime(7).` Yes.

`?-is_prime(8).` No.

Б) Да се дефинира предикат $q(N, X)$, който по дадено естествено число $N > 1$ връща в X най-големия прост делител на N . Може да се използват предикатите $p(N, L)$ и $is_prime(N)$ от подточка А.

Примери: `?-q(6, X).` X=3.

`?-q(7, X).` X=7.

Задача 9. (10 точки) В базата от данни със схема:

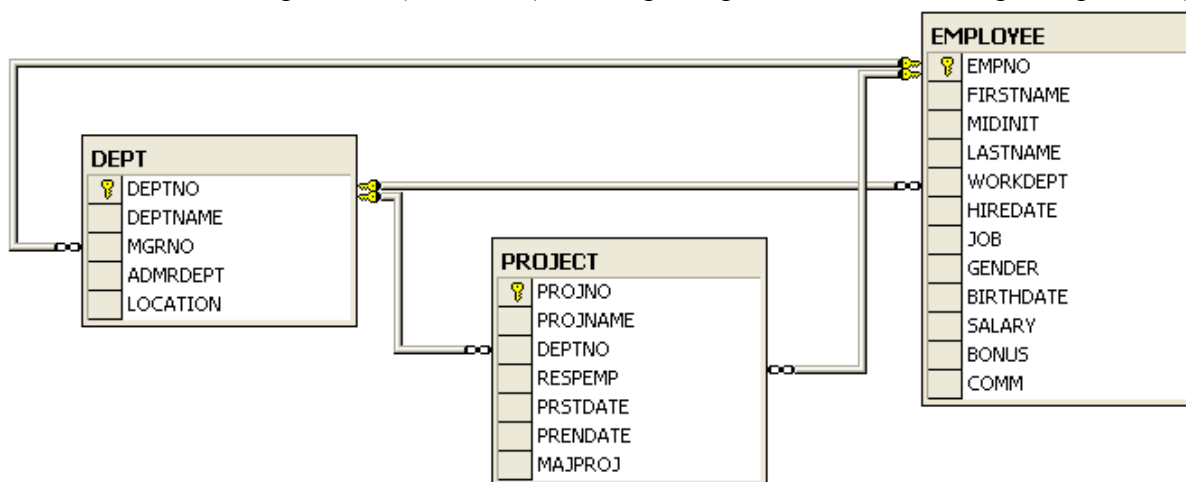
```
DEPT (DEPTNO, DEPTNAME, MGRNO, ADMRDEPT, LOCATION);
```

```
EMPLOYEE (EMPNO, FIRSTNAME, MIDNAME, LASTNAME, WORKDEPT, HIREDATE, JOB, GENDER, BIRTHDATE, SALARY, BONUS, COMM);
```

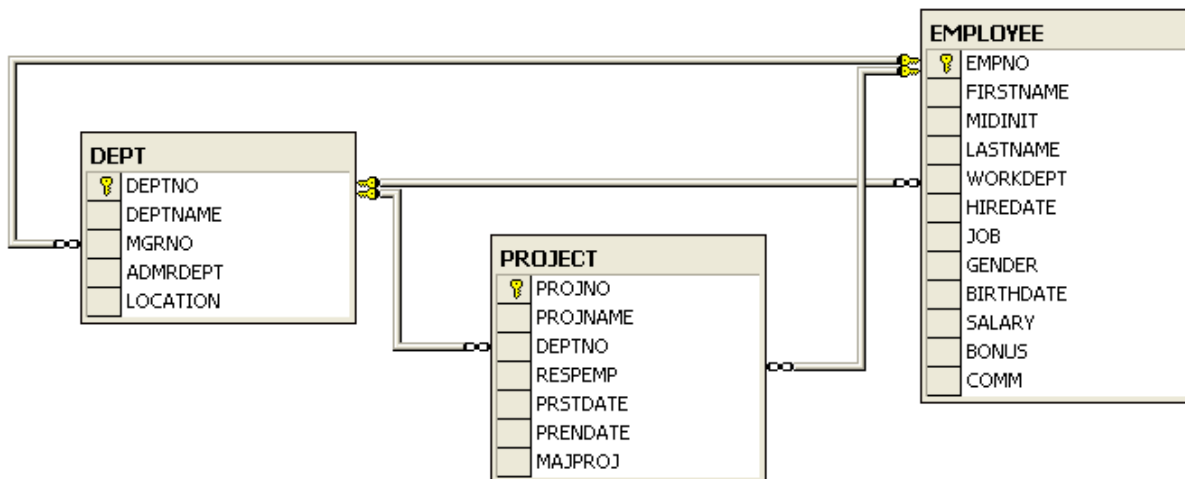
```
PROJECT (PROJNO, PROJNAME, DEPTNO, RESPEMP, PRSTDATE, PRENDATE, MAJPROJ);
```

се съхранява информация за департаменти, работници и проекти.

- Таблицата DEPT съдържа информация за номер и име на департамент (DEPTNO и DEPTNAME) и за номер на работник (MGRNO) – менажер за този департамент;
- Таблицата EMPLOYEE съдържа информация за номер на работник (EMPNO), лични данни (FIRSTNAME, MIDINIT, LASTNAME), длъжност (JOB), заплата (SALARY) и номера на департамента (WORKDEPT), към който работи;
- Таблицата PROJECT съдържа информация за номер (PROJNO) и име на проект (PROJNAME), начална и крайна дата на проекта (PRSTDATE, PRENDATE), департамента, към който е проектът (DEPTNO) и номера на работника – менажер на проекта (RESPEMP).



А) Да се напише заявка, която извежда името на департамента, броя на проектите и сумарната заплата на менажерите на проекти от този департамент.



Б) Като се използва заявката от подточка А, да се напише заявка, която извежда името на департамента с най-висока сумарна заплата на менажерите на проекти от този департамент.

Забележка: Ако има повече департаменти със сумарна заплата равна на най-високата сумарна заплата, тези департаменти също да се изведат.