

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



**ФАКУЛТЕТ ПО МАТЕМАТИКА  
И ИНФОРМАТИКА**

**ДЪРЖАВЕН ИЗПИТ  
ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР ПО ИНФОРМАТИКА”**

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

**Време за работа – 3 часа**

*Драги абсолвенти:*

- Пишете само на предоставените ви листове без да ги разкопчавате;
- Попълнете факултетния си номер горе вдясно на всички страници;
- Решението на всяка от задачите се разполага само в пространството 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.** (9т.) Да се напише програма на C++, която въвежда от клавиатурата правоъгълна матрица от реални числа с 5 реда и 10 колони . Програмата да отпечата текста "yes" в случай, че въведената матрицата има поне два реда, които са линейно зависими и "no" в противен случай. Казваме, че редовете  $a_0, a_1, \dots, a_9$  и  $b_0, b_1, \dots, b_9$  са линейно зависими, ако съществува такова число  $k$ , че  $a_0=k.b_0, a_1=k.b_1, \dots, a_9=k.b_9$ .



**Задача 2.** (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.

```
class A
{ public:
    A()
    { cout << "A::A()\n"; g(); 
    }
    void operator=(const A&)
    { cout << "A::=\n";
    }
    void f()
    { cout << "A::f()\n";
      g(); 
    }
    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";
    }
};
```

```
void main()
```

```
{ cout << "1:\n";
```

```
  B* bp = new B;
```

```
  cout << "2:\n";
```

```
  A* bp_a = bp;
```

```
  bp_a->f(); 
```

```
  cout << "3:\n";
```

```
  bp_a->g(); 
```

```
  cout << "4:\n";
```

```
  A a;
```

```
  a = *bp;
```

```
  cout << "5:\n";
```

```
  B b;
```

```
  b = *bp;
```

```
  cout << "6:\n";
```

```
  delete bp_a; 
```

```
  cout << "7:\n";
```

```
};
```



**Задача 3.** (12т.) Компресирано представяне на даден списък от стойности наричаме такъв списък от точкови двойки, за който неколнократното повторение на един елемент е заменено от точкова двойка, за която първият елемент показва повторената стойност, а вторият – броя на повторенията. Например компресираното представяне на

**(2 2 2 Stan 7 7 2)** е списъка **((2.3) (Stan.1) (7.2) (2.1))**.

Реализирайте двойка функции на Scheme за намиране по зададен списък на компресираното му представяне, и обратно.



**Задача 4.** (15т.) 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).
```

**Задача 4.1.** Напишете дефиниция на предикат **is\_prime(N)**, който по дадено естествено число **N>0** проверява дали **N** е просто число.

Можете да използвате предиката **p(N, L)**.

Пример: `?- is_prime(1). No.`

`?- is_prime(7). Yes.`

`?- is_prime(8). No.`

**Задача 4.2.** Напишете дефиниция на предикат **q(N, X)**, който по дадено естествено число **N>1** връща в **X** най-големия прост делител на **N**.

Можете да използвате предикатите **p(N, L)** и **is\_prime(N)** от Задача 4.1.

Пример: `?- q(6, X). X=3.`

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





**Задача 5.** (15т.) Дадена е булевата функция  $f(x, y, z) = ((x \oplus y) \rightarrow \bar{z}) \mid ((x \wedge \bar{y}) \equiv (\bar{y} \vee \bar{z}))$ .

x	y	z	f(x, y, z)

- a) Попълнете таблицата на функцията  $f(x, y, z)$
- b) Напишете Съвършената ДНФ на функцията  $f(x, y, z)$
- c) Намерете полинома на Жегалкин на  $f(x, y, z)$
- d) Изследвайте функцията  $f(x, y, z)$  за принадлежност към всяко от множествата  $T_0, T_1, S, M, L$ .



**Задача 6.** (15 т.) 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:

```
#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 );
        }
    }
}
```

**Задача 6.1)** Подчертайте операторите, които извеждат на стандартния изход (терминала)

**Задача 6.2)** Напишете вдясно какво се извежда на стандартния изход (терминала)



**Задача 7.** (10т.) В базата от данни със схема:

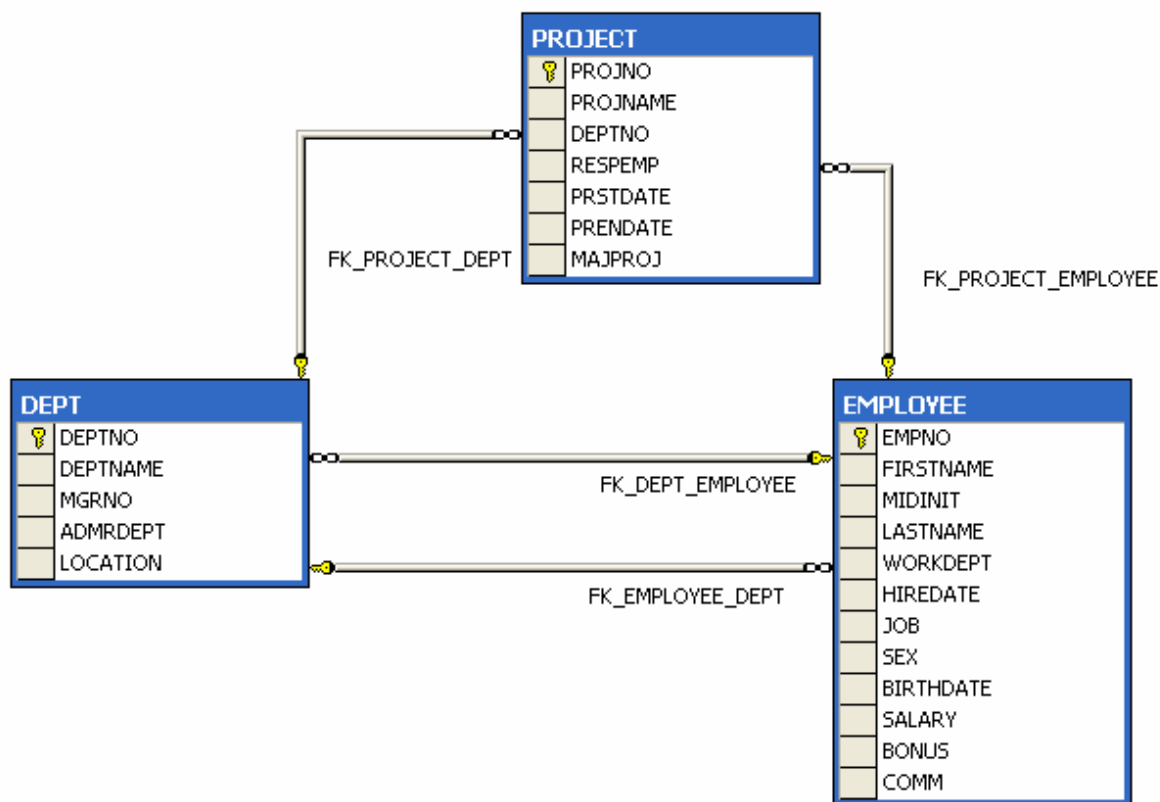
```
DEPT (DEPTNO, DEPTNAME, MGRNO, ADMRDEPT, LOCATION);
```

```
EMPLOYEE (EMPNO, FIRSTNAME, MIDNAME, LASTNAME, WORKDEPT, HIREDATE, JOB, SEX, 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 и за номер на работник (MGRNO) – менажер за този департамент.
- Таблицата EMPLOYEE съдържа информация за номер на работник, лични данни, длъжност (JOB), заплата (SALARY) и номера на департамента, към който работи (WORKDEPT).
- Таблицата PROJECT съдържа информация за номер и име на проект, начална и крайна дата на проекта, департамента, към който е проекта (DEPTNO), и номера на работника – менажер на проекта (RESPEMP).



**Задача 7. 1.** Напишете заявка, която извежда името на департамента, броя на проектите и сумарната заплата на менажерите на проекти от този департамент. (4 точки)

**Задача 7. 2.** Като използвате заявката от зад. 7. 1, напишете 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, те също да се изведат. (6 точки)



**Задача 8.** (15т.) Разполагаме с координатите на  $N$  града, което ни позволява да изчислим разстоянието по въздух между тях.

Разполагаме и с карта за съществуващите пътища от град до град с дължините им.

Да се състави функция на един от езиките C++, C# или Java за намиране на оптимален път по картата от град А до град В.

Търси се алгоритъм по-бърз от алгоритъма на Дийкстра в средния случай.

(Упътване: използвайте алгоритъма A\*)

Вход:

- $N$  – брой градове (номерирани с числата от 1 до  $N$ )
- $N$  двойки с координатите на градовете от вида  $(X, Y)$  в ортогонална координатна система
- $M$  – брой директни пътища от град до град
- $M$  тройки от вида (Град1, Град2, Разстояние)
- Номер на Стартов град
- Номер на Финален град

Изход:

- Оптималният път за придвижване от Стартовия град до Финалния град

