



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

**ФАКУЛТЕТ ПО МАТЕМАТИКА
И ИНФОРМАТИКА**

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

**ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)
10.09.2009 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)

А) Да се построи краен недетерминиран автомат, който разпознава езика, определен от регулярния израз R:

$$R = ((00) \cup (11))^*$$

Пример: Разпознава: ϵ , 00111100

Не разпознава: 0100, 0011100

Б) Да се построи минимален детерминиран автомат, еквивалентен на дадения недетерминиран автомат $K = (Q, \Sigma, \delta, s, F)$, където:

$$Q = \{ s, p, q \}$$

$$\Sigma = \{ a, b \}$$

$$F = \{ p \}$$

$$\delta(s, a) = \{ p \}, \delta(p, a) = \{ s, p, q \}, \delta(s, b) = \{ p, q \}, \delta(q, a) = \{ s \}$$

Задача 2. (6 точки) Да се напише отдясно на програмния код какво ще бъде изведено на стандартния изход като резултат от успешното изпълнение на дадения по-долу фрагмент от програма на C

```
main()
{
    .....
    int i, pid, flag = 2, status;

    pid = fork();
    if ( pid ) { -- flag; wait(&status); }
    else for ( i=1; i <= 2; i++ ) ++flag;
    ++flag;
    if ( flag == 3 ) exit ( - i );
    printf ( "flag = %d\n",flag);
    .....
}
```

Задача 3. (6 точки) Даденият по-долу фрагмент от команди на командния интерпретатор bash за LINUX се изпълнява успешно. Да се напише отдясно на командите какво и в кои оператори се извежда на стандартния изход.

```
count = 0
for var in a1 a2 a3
do
    echo $var > fxxx
done
cat fxxx
listpar=`grep a3 fxxx`
echo $listpar > fyuu
until cat fyuu | grep a2 > /dev/null
do
    count = `expr $count + 1`
    echo a2 >> fyuu
done
echo $count
echo $listpar
```

Задача 4. (10 точки) Да се напише програма, която въвежда от клавиатурата масив от 5 символни низа, всеки с максимална дължина 50 символа. Програмата да извежда текста "yes" в случай, че от въведените низове има такъв низ *s*, на който всички останали низове са поднизове и "no" в противен случай.

Забележка: Низът $a_0 a_1 \dots a_k$ е подниз на низа $b_0 b_1 \dots b_m$, ако $k \leq m$ и съществува такова естествено число i , че $0 \leq i \leq m-k$ и $a_0 = b_i, a_1 = b_{i+1}, \dots, a_k = b_{i+k}$. Например, "abc" е подниз на "0abc1", но не е подниз на "a0b1c".

Задача 5. (15 точки) Да се дефинира тип данни, описващ възел на двоично дърво от цели числа. Да се напише булева функция, която проверява дали дадено двоично дърво е балансирано.

Забележка. Едно двоично дърво е балансирано, ако за всеки негов връх v , височините на лявото и дясното поддървета на v се различават най-много с 1.

Задача 6. (10 точки) Да се попълни в празните полета изходът от предшестващите ги програмни конструкции.

<pre>#include <iostream.h> class Base { public: Base() { cout << "Base::Base()\n"; } Base(const Base&) { cout << "Base::copy()\n"; } virtual void f() { cout << "Base::f()\n"; } virtual ~Base() { cout << "Base::~~\n"; } }; class Der : public Base { public: Der() { cout << "Der::Der()\n"; } Der(const Der&) { cout << "Der::Copy()\n"; } void f() { cout << "Der::f()\n"; } ~Der() { cout << "Der::~~\n"; } }; void g1(Base a) { cout << "F:"; a.f(); } void g2(Der a) { cout << "F:"; a.f(); } void g3(Base& a) { cout << "F:"; a.f(); }</pre>	<pre>void main() { cout << "1:\n"; Der d; <div style="border: 1px solid black; height: 100px; width: 100%;"></div> cout << "2:\n"; g1 (d); <div style="border: 1px solid black; height: 100px; width: 100%;"></div> cout << "3:\n"; g2 (d); <div style="border: 1px solid black; height: 100px; width: 100%;"></div> cout << "4:\n"; g3 (d); <div style="border: 1px solid black; height: 100px; width: 100%;"></div> cout << "5:\n"; }</pre>
---	---

Задача 7. (15 точки) Да се реализира абстрактен клас **Sequence**, представящ безкрайна редица от числа и метод за достъп до нейните елементи (по пореден номер). Да се реализират класовете **ArithmeticProgression**, **PeriodicSequence** и **InterleavedSequence**, наследници на класа **Sequence**, които представят съответно аритметична прогресия, периодична редица и комбинация на две редици.

Периодична редица е такава безкрайна редица от числа, която се получава като последователно повторение на крайна редица от числа безкраен брой пъти. Комбинация на две редици $a_1, a_2, \dots, a_n, \dots$ и $b_1, b_2, \dots, b_n, \dots$ е редицата $a_1, b_1, a_2, b_2, \dots, a_n, b_n, \dots$

За класовете да се реализират подходящи конструктори, мутатори (методи за промяна на стойността на полетата) и ако е необходимо – деструктори.

Задача 8. (15 точки) Компресирано представяне на безкраен поток от числа наричаме такъв друг поток, за който неколкото последователно срещане на числото 0 е сменено от точкова двойка с първи елемент 0 и втори – броят на повторенията на 0 . Например, ако началото на един безкраен поток е $(0\ 0\ 0\ 1\ 0\ 0\ 7\ 3\ \dots)$, компресираното му представяне е безкраен поток с начало $((0\ .\ 3)\ 1\ (0\ .\ 2)\ 7\ 3\ \dots)$. Да се реализират на езика Scheme двойка функции, които преобразуват некомпресиран безкраен поток в компресиран и обратно.

Задача 9. (15 точки) Да означим с φ_1 , φ_2 и φ_3 следните предикатни формули:

$$\varphi_1: \forall X \forall Y \forall Z (p(X, Y) \& p(Y, Z) \Rightarrow p(X, Z))$$

$$\varphi_2: \exists X \exists Y (p(X, Y) \& p(Y, X))$$

$$\varphi_3: \forall X \neg p(X, X).$$

А) Да се покаже, че множеството $\{\varphi_1, \varphi_2\}$ е изпълнимо (т.е. има структура, в която са верни едновременно φ_1 и φ_2).

Б) С метода на резолюцията да се докаже, че множеството $\{\varphi_1, \varphi_2, \varphi_3\}$ вече не е изпълнимо.

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

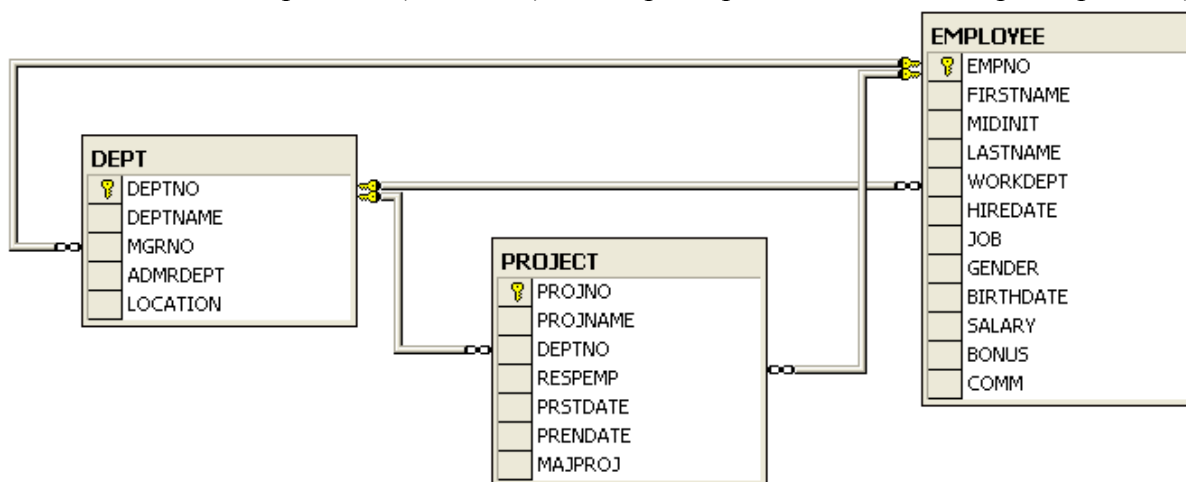
```
DEPT (DEPTNO, DEPTNAME, MGRNO, ADMRDEPT, LOCATION);
```

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

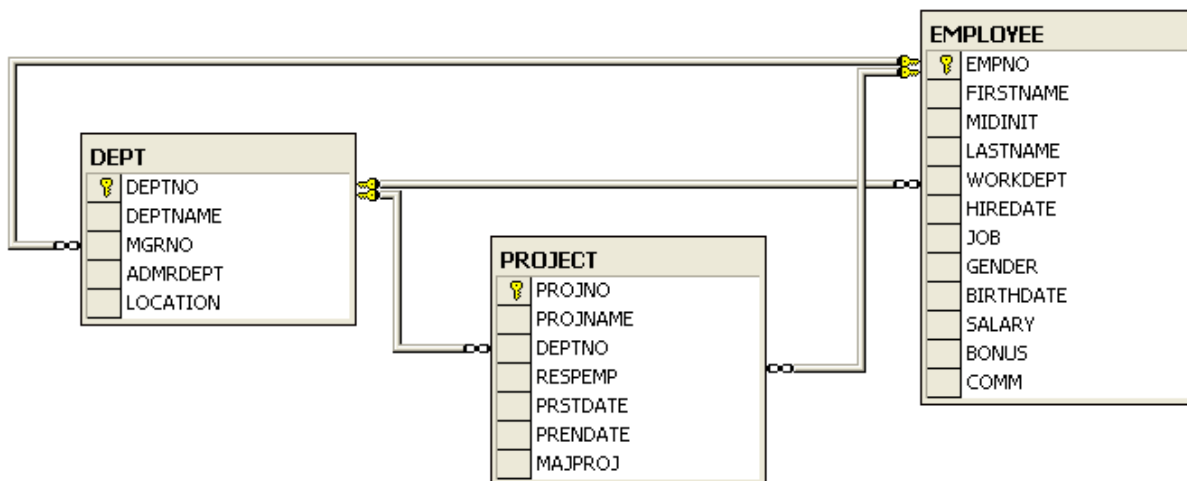
```
PROJECT (PROJNO, PROJNAME, DEPTNO, RESPEMP, PRSTDATE, PRENDATE, MAJPROJ);
```

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).



A) 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-много проекти.

Забележка: Ако има повече департаменти с брой на проектите, равен на максималния брой на проекти, тези департаменти също да се изведат.