

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



**ФАКУЛТЕТ ПО МАТЕМАТИКА
И ИНФОРМАТИКА**

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

**ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)
21.03.2009 г.**

Време за работа – 3 часа

Драги абсолвенти:

- Попълнете факултетния си номер на всички страници;
- За всяка от задачите, беловата с решението може да е само на листите, на които е изписано условието на съответната задача.

Изпитната комисия ви пожелава успешна работа.

Задача 1. (10т.) Да се построи краен детерминиран автомат, еквивалентен на дадения недетерминиран автомат

$$A = \langle \{q_0, q_1, q_2, q_3, q_4, q_5\}, \{0, 1\}, q_0, \delta, \{q_5\} \rangle$$

с функция на преходите δ , зададена чрез таблицата:

q	0	1
q ₀	{q ₁ , q ₂ }	∅
q ₁	{q ₁ , q ₂ }	{q ₁ }
q ₂	∅	{q ₃ }
q ₃	{q ₄ , q ₅ }	∅
q ₄	{q ₄ , q ₅ }	{q ₄ }
q ₅	∅	∅

Задача 2. (15т.) Да се избере и опише представяне на граф, ребрата на който са надписани със символи. Напишете програма, която прочита от текстов файл информация за графа и я съхранява чрез описаното представяне. На първия ред на файла са записани две числа разделени с празен символ – броя на върховете и броя на ребрата на графа. Следва по един ред за всяко ребро, съставен от две числа и символ, разделени също с празен символ, номерата на върховете съответно в началото и края на реброто и надписът на реброто. За този граф, при подаден начален връх да се намерят всички други върхове, достижими от него, така че пътят до тях не съдържа ребра с повтарящи се надписи.

Задача 3. (10 т.) Да се реализира абстрактен базов клас, представящ обект от файловата система. Този обект се характеризира с име. Класът да предоставя метод за определяне на заеманото от обекта дисково пространство.

- Да се реализира наследник на този клас представящ файл. Файлът трябва да пази информация за размера си.
- Да се реализира наследник на този клас представящ директория. Директорията съхранява масив от обекти от файловата система.
- Напишете кратка програма, която да създаде директория, съдържаща поне една поддиректория и поне един файл и след това да изведе на екрана размера на тази директория.

Задача 4. (8т.) Попълнете в празните полета текстът, който се отпечатва на конзолата в резултат на изпълнението на съответните програмни конструкции.

```
#include <iostream.h>

class Base
{
public:
    Base () {cout << "Base::Base()\n"; f(); g();}
    Base (Base&) {cout << "Base::Base(Base&)\n"; f();g();}

    virtual void f () {cout << "Base::f()\n";g();}
    void g () {cout << "Base::g()\n";}
};

class Derived1 : public Base
{
public:
    Derived1 () {cout << "Derived1::Derived1()\n"; f(); g();}
    Derived1 (Derived1 &) {cout <<
"Derived1::Derived1(Derived1&)\n";f();g();}

    void f () {cout << "Derived1::f()\n";g();}
    virtual void g () {cout << "Derived1::g()\n";}
};

class Derived2 : public Derived1
{
public:
    Derived2 () {cout << "Derived2::Derived2()\n";f();g();}
    Derived2 (Derived2 &) {cout <<
"Derived2::Derived2(Derived2&)\n";f();g();}

    void f () {cout << "Derived2::f()\n";g();}
    void g () {cout << "Derived2::g()\n";}
};

void fCopy (Base obj)
{
    obj.f ();
    obj.g ();
}

void main ()
{
```

```
Base *pBase = new Derived2;
```

a)

```
Derived1 *pDerived1 = new Derived2;
```

б)

```
Derived2 *pDerived2 = new Derived2;
```

в)

```
pBase->f ();
```

г)

```
pDerived1->f ();
```

д)

```
pDerived2->f ();
```

e)

```
fCopy (*pBase);
```

ж)

```
delete pBase, pDerived1, pDerived2;
```

```
}
```

Задача 5. (15 т.) Списъкът $l_1 = (u_0..u_k)$ наричаме *подписък* на списъка $l_2 = (v_0..v_n)$, ако $k \leq n$ и съществува такава i , че $0 \leq i \leq n - k$ и $u_0 = v_i, u_1 = v_{i+1}, \dots, u_k = v_{i+k}$. Пример: списъкът (1 2) е подписък на списъка (0 1 2 3), но не е подписък на списъка (1 0 2).

Да се дефинира функция (*count-sub l1 l2*) на езика Scheme, която намира колко пъти списъкът от числа $l1$ се среща като подписък на списъка от числа $l2$.

Пример: (*count-sub '(1 1) '(1 1 1 2 1 1)*) -> 3

Задача 6. (7т.) Опишете с думи действията, които ще се реализират при изпълнението на зададената командна процедура на езика на bash за Linux

```
if test -d /home/compiled
then :
else
    mkdir /home/compiled
fi
l_files=`ls $HOME/*.c `
proc=`ps | wc -l `
while [ $proc -lt 5 ]
do
    for i in $l_files
    do
        name=$i.exe
        cc -o $name $i &
        break
    done
    mv $i /home/compiled
    proc=`ps | wc -l `
    l_files=`ls $HOME/*.c `
done
```

Задача 7. (5 т.) Текстов файл с име procA съдържа следната последователност от команди на bash

```
br=0
for x in a b c
do for y
do if test $y -lt $#
then br=`expr $br + 1`
echo $br
else continue
echo $y
fi
done
break
done
```

Какъв резултат ще бъде изведен на стандартния изход при стартиране на изпълнение чрез командния ред

```
sh procA 1 2 3
```

- a) 1 б) 1 в) 1 г) 1 д) нито един от посочените
2 2 2 2
3 1 3
 2 3
 1
 2

Задача 8. (8 т.) Какво ще отпечата следващата програма на Пролог?

```
p(X) :- write(X), nl, Y is X + 2, q(Y).
p(X) :- Y is X + 1, write(Y), nl, q(Y).
q(X) :- Y is X + 5, write(Y), nl.
q(X) :- X < -4, p(X).
?- p(-7), write('===='), nl, fail.
```

Задача 9. (12 т.) С методите на резолюцията да се докаже, че множеството $\{\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \varphi_6\}$ от предикатната формула е неизпълнимо, където:

$$\varphi_1 \text{ е } \forall x(p(x) \Rightarrow (q(x) \vee \forall y r(x, y))),$$

$$\varphi_2 \text{ е } \forall x(p(x) \Rightarrow s(x)),$$

$$\varphi_3 \text{ е } \forall x(\forall y(t(y) \Rightarrow r(x, y)) \Rightarrow s(x)),$$

$$\varphi_4 \text{ е } \exists x p(x) \vee \forall x(s(x) \Rightarrow p(x)),$$

$$\varphi_5 \text{ е } \neg \exists x p(x) \& s(x) \Rightarrow p(x),$$

$$\varphi_6 \text{ е } p(c) \vee s(c).$$

(Тук p, q, s и t са едноместни предикатни символи, r е двуместен предикатен символ, а c е индивидуална константа.)