СОФИЙСКИ УНИВЕРСИТЕТ
“СВ. КЛИМЕНТ ОХРИДСКИ”

ФАКУЛТЕТ ПО МАТЕМАТИКА
И ИНФОРМАТИКА

ДЪРЖАВЕН ИЗПИТ
ЗА ПОЛУЧАВАНЕ НА ОКС “БАКАЛАВЪР ПО ИНФОРМАТИКА”

ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)
13.09.2008 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. (10 т) Да се построи минимален краен детерминиран автомат, еквивалентен на дадения краен детерминиран автомат $A=<\{q_1,q_2,q_3,q_4,q_5,q_6,q_7,q_8,q_9\},\{x,y,z\},q_1,\delta,\{q_1,q_2,q_4,q_7,q_8\}>$ с функция на преходите δ:

q	**x**	y	**z**
q_1	q_3	q_9	q_7
q_2	q_3	q_5	q_4
q_3	q_4	q_6	$\mathbf{q_3}$
q_4	q_9	q_9	q_2
q_5	q_8	q_9	q_9
q_6	q_2	q_6	q_2
q_7	q_5	q_5	q_1
q_8	q_5	q_5	q_1
q_9	q_7	q_5	q_5

Задача 2. (15т.) Реализирайте абстрактен базов клас *множество от числа от тип* `int`, който има метод за проверка на принадлежността към множеството.

1.1. Реализирайте производен клас, който представлява множеството от всички числа, които се делят без остатък на някакво предварително зададено в конструктора число.

1.2. Реализирайте и друг производен клас, който представя множество от числа чрез динамичен масив (който също се задава в конструктора).

1.3. Реализирайте външна за тези класове функция, която по зададен масив от множества и някакво число проверява дали то се съдържа в обединението на множествата от масива.

1.4. Демонстрирайте използването на тази функция в подходяща кратка програма.

Задача 3. (15 т.) Да се реализира подходящо представяне на граф. За така представения граф да се реализира функция, която намира всички върхове, до които има път от подаден връх с определена дължина *в брой ребра.*

Задача 4. (5т.) Зададен е следния фрагмент от програма:

```
int filed, i;
filed = creat ( “exam_txt”, 0777 );
close ( 1 );
i = dup (filed);
write ( i, “TEST\n”, sizeof(“TEST \n” ) );
write ( 1, “TEST \n”, sizeof(“TEST \n” ) );
```

Като резултат от изпълнението на последователността от зададените оператори:

- на терминала ще се изведе два пъти низа “TEST”
- на терминала ще се изведе низа “TEST” и във файла “exam_txt” ще се запише низа “TEST”
- във файла “exam_txt” ще се запише два пъти низа “TEST”

Задача 5. (7т.) Напишете какво ще бъде изведено на стандартния изход като резултат от изпълнението на дадения по-долу фрагмент от команди на bash

```
for  var  in  a1  a2  a3
do
     set  $var
done
shift
listpar=` echo  $* `
if  [  -n  “$listpar“  ]
then
     true
else
     false
fi
echo $?
echo  $listpar
```

Задача 6. (7 т.) Предикатът square от следната програма трябва да проверява дали едно число е точен квадрат:

```
between(A,A,B) :- A =< B.
between(X,A,B) :- A1 is A+1, between(X,A1,B).
square(N) :- between(K,0,N), N is K*K.
```

Открийте грешката и я поправете с промени само в един от трите реда.

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

`Classes`**(class, type, country, numGuns, bore, displacement)**
`Ships`**(name, class, launched)**
`Battles`**(name, date)**
`Outcomes`**(ship, battle, result)**

се съхранява информация за кораби (`Ships`) и тяхното участие в битки (`Battles`) по време на Втората Световна Война. Всеки кораб е построен по определен стереотип, определящ класа на кораба (`Classes`). Обикновено класът носи името на първия построен кораб от този клас.

Таблицата `Classes` **съдържа информация за класовете кораби:**

- **`class` – името на класа, първичен ключ;**
- **`type` – типът (‘`bb`’ за бойни кораби и ‘`bc`’ за бойни крайцери);**
- **`country` – страната (държавата), която строи такива кораби;**
- **`numGuns` – броят на основните оръдия;**
- **`bore` – калибърът им (диаметърът на отвора на оръдето в инчове);**
- **`displacement` – водоизместимостта (тегло, в тонове).**

Таблицата `Ships` **съдържа информация за корабите:**

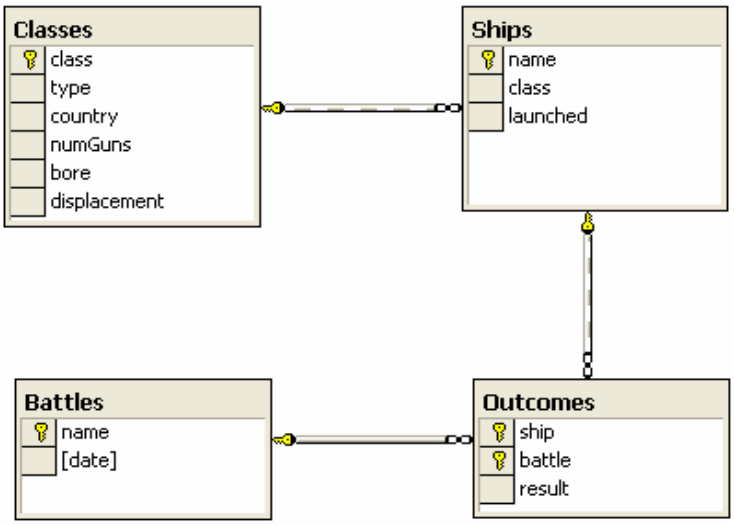
- **`name` – името на кораб, първичен ключ;**
- **`class` – името на неговия клас;**
- **`launched` – годината, в която корабът е пуснат на вода.**

Таблицата `Battles` **съхранява информация за битките:**

- **`name` – името на битката, първичен ключ;**
- **`date` – датата на провеждане.**

Таблицата `Outcomes` **съдържа информация за резултатата от участието на даден кораб в дадена битка (колоните `ship` и `battle` заедно формират първичния ключ):**

- **`ship` – името на кораба;**
- **`battle` – името на битката;**
- **`result` – резултатът (потънал – ‘`sunk`’, повреден – ‘`damaged`’, победил – ‘`ok`’).**



Задача 7.1 Посочете заявката, която извежда за всеки клас годината на най-рано и най-късно пуснатия на вода кораб:

a)
```
select c.class, min(ss.launched), max(ss.launched)
from classes c
join ships ss on ss.class = c.class;
```

b)
```
select c.class, min(ss.launched), max(ss.launched)
from classes c
join ships ss on ss.class = c.class
group by c.class;
```

c)
```
select c.class, min(ss.launched), max(ss.launched)
from classes c
join ships ss on ss.class = c.class
group by c.class
having min(ss.launched) and max(ss.launched);
```

d)
```
select c.class, min(ss.launched)
from classes c
join ships ss on ss.class = c.class
union all
select c.class, max(ss.launched)
from classes c
join ships ss on ss.class = c.class;
```

Задача 7.2 Посочете заявката, която извежда държавата/държавите с най-много класове:

a)
```
select c.country
from classes c
where not exists
    ( select *
      from classes c1
      where c1.country != c.country and
            count(c1.class) > count(c.class)
     );
```

b)
```
select c.country
from classes c
group by c.country
having max(count(c.class));
```

c)
```
select c.country
from classes c
where count(*) = ( select max(count(c1.class)
                     from classes c1
                     group by c1.country
                   )
group by c.country;
```

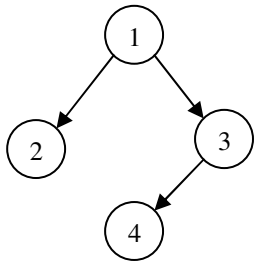
d)
```
select c.country
from classes c
group by c.country
having count(*)>= all ( select count(*)
                        from classes c1
                        group by c1.country
                      );
```

Задача 8. (14 т.) Нека е дадено следното представяне на двоично дърво с произволни стойности по върховете:

- празният списък *()* е празно дърво
- ако t_1 и t_2 са две двоични дървета, то списакът с три елемента $(x\ t_1\ t_2)$ е двоично дърво със стойност на корена x, ляво поддърво t_1 и дясно поддърво t_2.

Да се дефинира функция *(leaves t)*, намираща списък от стойностите по листата на дървото t, представено по писания начин.

Пример: следното двоично дърво



Се представя чрез списъка *t = (1 (2 () ()) (3 (4 () ()) ()))*. За него *(leaves t) = (2 4)*.

Задача 9. (10 т.) 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 () {}
      Base (Base&) {cout << "Base::Base(Base&)\n";}

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

      ~Base () {cout << "Base::~Base()\n";}
};

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

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

      virtual ~Derived1 () {cout << "Derived1::~Derived1()\n";}
};

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

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

      ~Derived2 () {cout << "Derived2::~Derived2()\n";}
};


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

void main ()
{

      Base *aBase  = new Derived2 [2];
      Derived1 *aDerived1 = new Derived2 [2];
      Derived2 *aDerived2 = new Derived2 [2];

      aBase->f ();
      aBase->g ();
```

а)

```
aDerived1->f ();
aDerived1->g ();
```

Б)

```
aDerived2->f ();
aDerived2->g ()
```

в)

```
fCopy (*aBase);
```

Г)

```
delete []aDerived1;
```

д)

```
delete []aDerived2;
```

е)

```
delete []aBase;
```

ж)

```
}
```