

**ДЪРЖАВЕН ИЗПИТ ЗА ПОЛУЧАВАНЕ НА ОКС "БАКАЛАВЪР ПО ИНФОРМАТИКА"**

18 – 19 . 03 . 2006 г.

**ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)**

**Задача 1.**

В следната булева функция броят на променливите е означен с **n**

$$f(\tilde{x}^n) = \overline{x_1 \oplus x_2 \vee x_3 x_4 \dots x_n}, n \geq 3$$

Коя от следните посочени стойности съответства на дължината на Съвършената Дизюнктивна Нормална Форма на тази функция ?

а)  $2^{n-1} + 1$ ;      б)  $2^{n-1} + 2$ ;      в)  $2^{n-1} - 2$ ;      г)  $2^n - n + 1$ .

**Задача 2.**

В следната булева функция броят на променливите е означен с **n**.

Да се определи за кои стойности на  $n \geq 3$  тази функция е Шеферова.

$$f(\tilde{x}^n) = \overline{x_1 x_2} \oplus \overline{x_2 x_3} \oplus \dots \oplus \overline{x_{n-1} x_n} \oplus \overline{x_n x_1}$$

**Задача 3.**

Дадена е следната автоматната граматика :

$\Gamma = \langle \{S, A, B, C, D\}, \{0, 1\}, S, \{S \rightarrow 0A \mid 1B \mid 0, A \rightarrow 0A \mid 1B \mid 0D \mid 0, B \rightarrow 0C \mid 0D \mid 1B \mid 0, C \rightarrow 1B\} \rangle$

**3. 1.**

Да се построи детерминиран краен автомат, който разпознава езика породен от нея.

**3. 2.**

Да се построи минимален детерминиран краен автомат, еквивалентен на горния.

**Задача 4.**

Даден е следният програмен код на езика C,

в който са използвани системни примитиви на ОС UNIX и LINUX :

```
#include <stdio.h>
#define COM "ls -dl"
main ( )
{
    int i = 1, status ;
    if ( fork ( ) ) {
        - - i ; wait (&status) ;
    }
    else {
        ++i ;
        execlp( COM, COM, 0 ) ;
        i += 2 ;
    }
    printf ( "\n i = %d\n", i ) ;
}
```

Какво ще бъде изведено на стандартния изход,  
като резултат от изпълнението на този програмен код.

**Задача 5.**

Даден е следният програмен код на езика C,

в който са използвани системни примитиви на ОС UNIX и LINUX:

```
#include <fcntl.h>
main(int argc, char *argv[])
{
```

```
int fd ;
close ( 1 );
if ( ( fd=open (argv [ 1 ], O_CREAT|O_WRONLY|O_TRUNC, 0666) ) == -1 ) exit ( 1 );
write ( fd, "Hello ", 6 );
write ( 1, "END", 3 );
}
```

Заградете с кръгче всички възможни резултати от изпълнението на този програмен код.

- а) Низът “Hello END” ще бъде изведен на терминала.
- б) Низът “Hello ” ще бъде записан във файл, чието име е подадено като първи аргумент в командния ред, а низът “END” ще бъде изведен на терминала.
- в) Низът “Hello END” ще бъде записан във файл, чието име е подадено като първи аргумент в командния ред.
- г) Процесът ще приключи връщайки код на завършване 1.

### Задача 6.

Какво ще бъде изведено на стандартния изход (терминала) като резултат от изпълнението на следната последователност от команди на интерпретатора **bash** (заградете верния отговор с кръгче).

```
set 2 3 4
var=0
echo $* > fff
str=`cat fff`
for each in $str
do
    if test $each -eq 3
    then var=`expr $var + 1`
    echo $var
fi
done
```

- |      |      |      |      |      |      |
|------|------|------|------|------|------|
| а) 1 | б) 2 | в) 3 | г) 4 | А) 1 | е) 2 |
|      |      |      |      | 2    | 3    |
|      |      |      |      | 3    | 4    |

### Задача 7.

Да се напише командна процедура (последователност от команди) на езика на интерпретатора **bash** за LINUX, която прочита десетично число от стандартния вход (терминала), и в случай, че то е по-малко от броя на онези файлове в текущия каталог (директория), които имат имена съдържащи разширение “out”, прехвърля тези файлове във каталога, чието име е подадено като позиционен параметър в командния ред при стартирането.

### Задача 8.

Какво ще бъде изведено на екрана при изпълнението на следния програмен код на C ?

```
#include <iostream.h>
int m, d, s ;
for ( m = 1 ; d = 28 , m < 12 ; m += m )
{ s = 2 - 2 * ( m == 2 ) - ( m == 4 + m == 6 + m == 9 + m == 11 ) ;
  switch ( s )
  { case 2 : d += 1 ;
    case 1 : d += 2 ;
  }
  cout << d << ' ' ;
}
```

**Задача 9.**

Какво ще бъде изведено на екрана  
при изпълнението на следния програмен код на C ?

```
#include <iostream.h>
int M ( int a, int b, int *m )
{ b = b - a ;
  a += a + b ;
  b = b < 0 ? - b : b ; *m = ( a - b ) / 2 ;
  return (a + b) / 2 + 1 ;
} ;

void main ( )
{ int a = 2 ;
  int b = 3 ;
  a = M (a, b, &b) ;
  cout << a << b ;
}
```

**Задача 10.**

Какво ще бъде изведено на екрана  
при изпълнението на следния програмен код на C ?

```
#include <iostream.h>
void print (char *p)
{ if (*(p ++)) print (++ p) ;
  cout << * (-- p) ; } ;

void main ( )
{ char str [ ] = "0123456789" ;
  print (str) ; }
```

**Задача 11.**

Какъв ще бъде резултатът от изпълнението на следната програма:

```
#include <iostream.h>
class KLAS
{private:
  int a, b;
public:
  KLAS(int=0, int=1);
  KLAS(const KLAS&);
  KLAS& operator=(const KLAS&);
  ~KLAS();
  void print() const;
};

KLAS::KLAS(int x, int y)
{a = x;
 b = y;
 cout << "Конструктор \n";
}

KLAS::KLAS(const KLAS& r) : a(r.a+3), b(r.b+5)
{cout << "Конструктор за присвояване \n";
}

KLAS& KLAS::operator=(const KLAS& p)
{cout << "Оператор =\n";
 if(this!=&p)
 {a = p.a+10;
  b = p.b+20;
```

Изпишете решението тук:

```

}
return *this;
}
KLAS::~~KLAS()
{cout << "Деструктор()\n";
}
void KLAS::print() const
{cout << a << " " << b << endl;
}

void main()
{KLAS a, b(2), c(3,4), d(c);
a.print(); b.print(); c.print();
KLAS f = d; f.print();
KLAS g(c); g.print();
d = g; d.print();
}

```

### **Задача 12.**

Дефинирана е следната йерархия от класове:

<pre> class base {private: int a1; protected: int a2; public: int a3(); } a;  class der1: public base {private: int a4; protected: int a5; public: int a6(); } b; </pre>	<pre> class der2 : public der1 {private: int a7; protected: int a8; public: int a9(); } c;  class der3 : der2 {private: int a10; protected: int a11; public: int a12(); } d; </pre>
--	---

Определете възможностите за пряк достъп на обектите: a, b, c и d до компонентите на класовете.

Обектът a има пряк достъп до компонентите : .....

Обектът b има пряк достъп до компонентите : .....

Обектът c има пряк достъп до компонентите : .....

Обектът d има пряк достъп до компонентите : .....

### **Задача 13.**

Отстранете грешните обръщения към функции в main() .

Какъв ще бъде резултатът от изпълнението на програмата след отстраняване на грешките?

```

#include <iostream.h>
class B
{public:
virtual void p() const
{cout << "B::p()\n";
}
private:
virtual void s() const
{cout << "B::s() \n";
}
}

```

```

    }
protected:
virtual void t() const
{cout << "B::t( )\n";
}
public:
void usual() const
{cout << "B::usual()\n";
    p();
    s();
    t();
}
};

```

```

class D : public B
{private:
    void p() const
    {cout << "D::p()\n";
    }
public:
    void s() const
    {cout << "D::s()\n";
    }
protected:
    void t() const
    {cout << "D::t()\n";
    }
};

```

```

class F : public D
{protected:
    void s() const
    {cout << "F::s()\n";
    }
private:
    void t() const
    {cout << "F::t()\n";
    }
public:
    void p() const
    {cout << "F::p()\n";
    }
private:
    void usual()
    {cout << "F::usual()\n";
    s();
    t( );
    }
};

```

```

void main()
{ B b; D d; F f;
  B* pb = &b;
  B* pd = &d;
  D* pf = &f;
  B* pf1 = &f;

```

```

b.usual();    pb->p();    pb->s();    pb->t();    pb->usual();
d.usual();    pd->p();    pd->s();    pd->t();    pd->usual();
f.usual();    pf->p();    pf->s();    pf->t();    pf->usual();
              pf1->p();    pf1->s();    pf1->t();    pf1->usual();
}

```

**Задача 14.** Следният шаблон на клас реализира линеен свързан списък, представен с две указателни връзки.

```

template <class T>
struct elem_link2          // тройна кутия, носител на елемент на списъка
{
    T inf;
    elem_link2<T> *pred,
                  *succ;
};

template <class T>
class DLList
{
public:
    DLList();                // конструктор
    ~DLList();               // деструктор
    DLList(DLList const&);   // конструктор за присвояване
    DLList& operator=(DLList const &); // оператор =
    void print();            // извежда свързан списък
    void print_reverse();    // извежда свързан списък
    void IterStart(elem_link2<T>* = NULL); // инициализира итератора CurrentS
    void IterEnd(elem_link2<T>* = NULL);   // инициализира итератора CurrentE
    // следващата член-функция връща текущата стойност на итератора CurrentS,
    // след което го премества на следващата позиция
    elem_link2<T>* IterSucc();
    // следващата член-функция връща текущата стойност на итератора CurrentE,
    // след което го премества на следващата позиция
    elem_link2<T>* IterPred();
    void ToEnd(T const & x); // включва x в края на свързан списък
    // изтрива елемента, сочен от указателя p и го записва в x
    void DeleteElem(elem_link2<T> *p, T & x);
private:
    elem_link2<T> *Start,    // указател към началото
                  *End,      // указател към края
                  *CurrentS, // итератор към началото
                  *CurrentE; // итератор към края
    void DeleteList();
    void CopyList(DLList const &);
};

```

**14.1.** Реализирайте член-функциите на каноничното представяне :  
(конструктори, деструктор и оператор =) на шаблона на класа DLList.

**14.2.** Член-функцията **void IterEnd(elem\_link2<T>\* = NULL);**  
инициализира итератора **CurrentE** по следния начин:  
В случай, че явно е указана стойност на указателя – формален параметър на процедурата, итераторът **CurrentE** се инициализира с тази стойност.  
В противен случай итераторът се инициализира със стойността на указателя сочещ края на списъка. Реализирайте процедурата **IterEnd** !

**14.3.** Член-функцията `elem_link2<T>* IterPred( );`  
връща текущата стойност на итератора `CurrentE`,  
след което го премества на следващата позиция. Реализирайте я !.

**14.4.** Свързан списък, съдържащ  $2n$  цели числа  $a_1, a_2, \dots, a_{2n-1}, a_{2n}$ , е представен чрез две връзки.  
Да се напише външна функция, която намира сумата:  $S = a_1 \cdot a_{2n} + a_2 \cdot a_{2n-1} + \dots + a_n \cdot a_{n+1}$

**Задача 15.** Шаблонът на класа `tree` реализира двоично дърво с елементи от тип `T`.

```
template <class T>
struct node
{
    T inf;
    node *Left,
        *Right;
};
template <class T>
class tree
{
public:
    tree(); // конструира празно двоично дърво
    ~tree(); // деструктор
    tree(tree const&); // конструктор за присвояване
    tree& operator=(tree const&); // оператор =
    bool empty() const; // проверява дали двоично дърво е празно
    T RootTree() const; // намира корена на двоично дърво
    tree LeftTree() const; // намира лявото поддърво на двоично дърво
    tree RightTree() const; // намира дясното поддърво на двоично дърво
    // Create3 създава непразно дв. дърво по зададени корен, ляво и дясно поддърво
    void Create3(T const&, tree<T> const&, tree<T> const&);
    void print() const; // извежда двоично дърво
    void Create(); // създава двоично дърво
    int depth() const; // намира дълбочината на двоично дърво
private:
    node<T> *root;
    // ....
};
```

**15.1.** Дефинирайте член-функцията `int depth() const;` на шаблона на класа `tree`.

**15.2.** Едно двоично дърво е по-късо ( $<$ ) от друго двоично дърво,  
ако дълбочината му е по-малка от тази на другото дърво.  
Предефинирайте оператора `<` чрез шаблон на функция-приятел на шаблона на `tree`.

**15.3.** Дефинирайте член-функция на шаблона на класа `tree`,  
която извежда на екрана листата на двоично дърво.

**15.4.** Дефинирайте външна функция, която да може да прилага дадена целочислена функция  
към всеки от върховете на дадено двоично дърво с цели върхове.

**Задача 16.** Дефинирайте примерна функция на езика Scheme,  
която генерира логаритмичен итеративен процес:

Обяснете каква задача се решава с така дефинираната функция:

(т.е. оценена от интерпретатора на Scheme) следната поредица от дефиниции:  
(define (f p l)

```

(cond ((null? l) l)
      ((p (car l)) (cons (car l) (f p (cdr l))))
      (else (f p (cdr l))))
(define (g l n)
  (f (lambda (x) (< (length x) n)) l))
(define l '((a b c) (a b d e) (a c d e f) (a d)))

```

Каква е оценката на израза `(map (lambda (x) (cons 'g x)) (g l 4))` :

### **Задача 18.**

Кой метод за сортиране се реализира от функцията `sort` със следната дефиниция:

```

(define (sort pred lst)
  (if (null? lst) lst
      (append
       (sort pred (filter (lambda (el) (pred el (car lst))) (cdr lst)))
       (list (car lst))
       (sort pred (filter (lambda (el) (not (pred el (car lst)))) (cdr lst))))))
(define (filter f lst)
  (cond ((null? lst) '())
        ((f (car lst)) (cons (car lst) (filter f (cdr lst))))
        (else (filter f (cdr lst)))))

```

- a) метод на пряката селекция
- b) сортиране чрез вмъкване
- c) бързо сортиране
- d) пирамидално сортиране

### **Задача 19.**

Дайте пример за неявно (индиректно) конструиране на безкраен поток със средствата на Scheme:

Обяснете какъв вид имат елементите на така генерирания поток:

### **Задача 20.**

Базата от правила и работната памет (базата от факти) на една система, основана на правила, са описани със средствата на езика Prolog както следва:

```

% Дефиниции на използваните оператори
:- op(900,fx,if).
:- op(890,xfx,then).
:- op(880,xfy,or).
:- op(870,xfy,and).

% Правила
if has_feathers(X) then bird(X).
if has_gills(X) then fish(X).
if has_hair(X) or gives_milk(X) then mammal(X).
if bird(X) and has_long_neck(X) then swan(X).
if mammal(X) and has_long_neck(X) then giraffe(X).
if mammal(X) and can_speak(X) then human(X).
if bird(X) and can_speak(X) then parrot(X).
if can_speak(X) then intelligent(X).

% Факти (работна памет)
fact(has_hair(gogo)).
fact(has_feathers(bobo)).
fact(has_feathers(pepo)).
fact(has_gills(fifi)).
fact(can_speak(bobo)).
fact(can_speak(gogo)).
fact(has_long_neck(pepo)).

```



Предполага се, че интерпретаторът на правилата извършва прав извод (извод, управляван от данните), като на всяка стъпка от работния си цикъл записва заключението на избраното за изпълнение правило като нов факт в края на работната памет.

В конфликтното множество се включват само правила, при чието изпълнение се записват нови факти в работната памет.

Какво ще бъде съдържанието на работната памет след завършване на работата на интерпретатора на правилата, ако на всяка стъпка от работния си цикъл той избира и изпълнява това правило от конфликтното множество, чието условие се удовлетворява от най-скоро записан в работната памет факт ?

### Задача 21.

Дадени са правилата R1 и R2 и фактите P1, P2, P3, P4 и P5.

Какви ще бъдат факторите за сигурност (Certainty Factors – CF) на C1 и C2 след изпълнението на правилата ?

R1: IF (P1 and (P2 or P3)) or P4 or P5 THEN C1 : 0.6

R2: IF (P1 and P2) or P3 or C1 THEN C2 : 0.5

CF(P1)=0.1   CF(P2)=0.4   CF(P3)=0.2   CF(P4)=0.5   CF(P5)=0.8

### Задача 22.

Фреймовете в дадена фреймова система имат следната структура:

```
(<frame-name> (<slot-1> (<facet-11> <value-11>
                        (<facet-12> <value-12>
                          ... )
                        (<slot-2> (<facet-21> <value-21>
                        (<facet-22> <value-22>
                          ... ) ... )
```

В базата от знания на системата са дефинирани следните фреймове:

```
(student (favourite_subject (value literature))
         (favourite_sport (value football)))
(student_at_FMI (favourite_subject (default mathematics))
               (favourite_sport (default chess))
               (isa (value student)))
(peter (isa (value student_at_FMI))
       (age (value 20))
       (favourite_sport (value backgammon)))
```

Какъв ще бъде резултатът от Z-търсенето в посочената база от фреймове за слота favourite\_subject на фрейма peter:

Какъв ще бъде резултатът от N-търсенето в посочената база от фреймове за слота favourite\_subject на фрейма peter:

### Задача 23.

Даден е ориентиран граф, представен чрез поредица от факти на Prolog от вида arc(<Node1>,<Node2>), всеки от които означава, че в графа съществува дъга с начало <Node1> и край <Node2>:

```
arc(s,a).   arc(s,b).   arc(s,c).arc(a,d).   arc(b,a).   arc(b,e).
arc(b,f).   arc(c,f).   arc(d,g).   arc(e,g).   arc(f,g).
```

Дадена е също така поредица от факти на Prolog от вида h(<Node>,<Cost>), дефиниращи евристичната функция, с помощта на която се пресмята приближена стойност <Cost> на разстоянието от върха <Node> до върха "g":

$h(a,1).$	$h(b,2).$	$h(c,3).$	$h(d,3).$
$h(e,4).$	$h(f,2).$	$h(g,0).$	$h(s,5).$

Ако задачата е да се намери път от върха “s” до върха “g” по метода Best-first search, какво ще бъде полученото решение:

#### Задача 24.

За кои от изброените двойки списъци  $L1$  и  $L2$ , при цел  $?$  –  $L1 = L2$ , системата PROLOG ще отговори с YES и за кои с No?  
Обосновете отговорите си.

- |                     |   |                  |
|---------------------|---|------------------|
| a) $L1 = [X   Y]$   | и | $L2 = [a, b]$    |
| б) $L1 = [X   [Y]]$ | и | $L2 = [a, b]$    |
| в) $L1 = [X, X]$    | и | $L2 = [a   [a]]$ |
| г) $L1 = [X   X]$   | и | $L2 = [a, a]$    |

#### Задача 25.

Нека  $F1$  е множеството на едноместните частични функции, дефинирани в множеството на естествените числа  $N$ .

Кои от изброените по-долу оператори  $\Gamma_i$ , дефинирани в множеството  $F1$ , имат най-малка неподвижна точка  $f_{\Gamma_i}$ , която НЕ Е тотална функция?

Обосновете отговорите си (посочете поне едно  $x$ , за което  $\neg f_{\Gamma_i}(x)$ ).

- |  |
|--|
| a) $\Gamma_1(f)(x) \cong \text{if } x = 0 \text{ then } 0 \text{ else } f(x+1)$                              |
| б) $\Gamma_2(f)(x) \cong \text{if } x \equiv 0 \pmod{2} \text{ then } x \setminus 2 \text{ else } f(x+1)$    |
| в) $\Gamma_3(f)(x) \cong \text{if } x \equiv 0 \pmod{2} \text{ then } x \setminus 2 \text{ else } f(f(x+1))$ |
| г) $\Gamma_4(f)(x) \cong \text{if } x = 0 \text{ then } 0 \text{ else } f(x-1)$                              |

#### Задача 26.

Нека  $R$  е следната рекурсивна програма над множеството  $Z$  на целите числа:

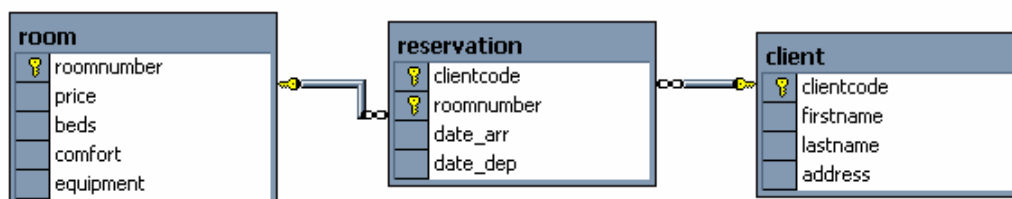
$R : F(X, Y)$  where  
 $F(X, Y) = \text{if } X \leq 0 \text{ then } G(Y) \text{ else } F(X-1, Y+1)$   
 $G(X) = \text{if } X = 0 \text{ then } 0 \text{ else } G(X-1) + 2$ .

Кои от следните условия са верни за  $D_V(R)$  (денотационната семантика на  $R$  с предаване на параметрите по стойност):

- |   |
|---|
| a) $\forall x ( (x \in Z \ \& \ !D_V(R)(x)) \Rightarrow D_V(R)(x) \cong 2x )$                 |
| б) $\forall x ( (x \in Z \ \& \ x \leq 0 \ \& \ !D_V(R)(x)) \Rightarrow D_V(R)(x) \cong 2x )$ |
| в) $\forall x ( (x \in Z \ \& \ !D_V(R)(x)) \Rightarrow D_V(R)(x) \cong 4x )$                 |
| г) $\forall x ( x \in Z \Rightarrow (!D_V(R)(x) \ \& \ D_V(R)(x) \cong 4x) )$                 |

#### Задача 27.

Базата от данни за един **хотел** притежава схема:



В таблицата **room** се съхранява информация за номер на стая (първичен ключ), цена на стая, брой легла в стая, тип комфорт в стая, допълнително оборудване.

roomnumber	price	beds	comfort	equipment
10	80	1	WC	NO
11	80	1	WC	NO
12	100	2	SHOWER	NO
13	120	1	BATH	NO
14	140	2	BATH	TV
15	180	3	BATH	TV
20	80	1	WC	NO
21	80	1	WC	NO
22	100	2	SHOWER	NO
23	120	1	BATH	NO
24	140	2	BATH	TV
25	180	3	BATH	TV
30	90	2	WC	NO
31	90	2	WC	NO
32	100	2	SHOWER	NO
33	120	1	BATH	NO
34	140	2	BATH	TV
35	180	3	BATH	TV
40	90	2	WC	NO
41	90	2	WC	NO
42	100	2	SHOWER	NO
43	120	1	BATH	NO
44	140	2	BATH	TV
45	180	3	BATH	TV

В таблица **client** се съдържа код на клиента (първичен ключ), име, фамилия, адрес.

clientcode	firstname	lastname	address
1001	Ivan	Petrov	Sofia
1002	Peter	Stoyanov	Plovdiv
1003	Ivanka	Sergeeva	Varna
1004	Kiril	Ivanov	Yambol
1005	Maria	Petrova	Pernik
1006	Vasilka	Stoilova	Vratza
1007	Mariana	Ivanova	Varna
1008	Ana-Maria	Petrova	Sofia

В таблица **reservation** се съдържа код на клиента, номер на стая, дата на пристигане, дата на заминаване.

Трите колони clientcode, roomnumber, date\_arr формират първичен ключ.

Clientcode е външен ключ към таблица client.

Roomnumber е външен ключ към таблица room.

clientcode	roomnumber	date_arr	date_dep
1001	11	11.11.2005 г.	13.11.2005 г.
1002	21	24.12.2005 г.	27.12.2005 г.
1003	31	23.12.2005 г.	27.12.2005 г.
1004	41	28.12.2005 г.	31.12.2005 г.
1006	32	03.12.2005 г.	04.12.2005 г.
1006	32	10.12.2005 г.	11.12.2005 г.
1006	32	17.12.2005 г.	18.12.2005 г.
1006	32	24.12.2005 г.	25.12.2005 г.
1007	45	01.1.2006 г.	<NULL>
1008	43	01.2.2006 г.	06.2.2006 г.

**27.1.** При горните общи предположения напишете заявка за извеждане на капацитета на хотела ( т.е. общият брой на леглата в хотела).

**27.2.** При горните общи предположения напишете заявка за извеждане на средната цена за стаи според типа на техния комфорт (WC, SHOWER, BATH), но само за онези, за които средната цена е по-голяма от 110.

**7.3.** При горните общи предположения напишете оператор за извеждане на списък (по име и фамилия) на онези клиенти, които са резервирани стая преди 25-12-2005г. (date\_arr)