

име:	фак. №	стр. 1/14
------	--------	-----------

СОФИЙСКИ УНИВЕРСИТЕТ "СВ. КЛИМЕНТ ОХРИДСКИ"
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

ДЪРЖАВЕН ИЗПИТ
ЗА ПОЛУЧАВАНЕ НА ОКС "БАКАЛАВЪР ПО ИНФОРМАТИКА"
19-20.07.2003 г.

ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)

Оценка от практическите задачи:	
задача	брой точки
Задача 1	
Задача 2	

Задача 1

- 1) **{1 т.}** Даден е следният фрагмент от програма на езика C:

```
int x, y, z;
z = ( x = - y ) ? ( - x / y ) : y ;
```

Кой от следните оператори е еквивалентен на горния оператор за присвояване?

- a) `z = (x = - y) ;`
- b) `z = (- x / y) ;`
- c) `z = y ;`
- d) `z = !! (x = - y);`

- 2) **{1,5 т.}** Дадени са следните три варианта A), B) и C) на оператори switch:

A) `switch (k)`
`{ case 1: y = y + 1; break;`
`case 2: y = y + 2; break;`
`case 3: y = y + 3; break ;`
`};`

B) `switch (k)`
`{ case 3: y = y + 1 ;`
`case 2: y = y + k ; break;`
`case 1: y = y ++ ; break;`
`};`

C) `switch (k)`
`{ case 3: ++ y ;`
`case 2: y = -- y + 3 ; break;`
`case 1: y += 1 ; break;`
`};`

Кое от следните твърдения е вярно?

- a) `A ≡ B , A ≠ C`
- b) `A ≠ B , A ≡ C`
- c) `A ≡ B , A ≡ C`
- d) `A ≠ B , A ≠ C`

- 3) **{1 т.}** Даден е следният оператор на езика C:

```
for ( S=0, i=0; i<3 ; i++ ) S+=i ;
```

Кой от следните оператори е еквивалентен на него?

- a) `for (S=0, i=0; i<3 ; i++ , S+=i) ;`
- b) `for (S=0, i=0; i<3 ; S+=i , i++) ;`
- c) `for (S=0, i=0; i<3, S+=i ; i++) ;`
- d) `for (S=0, i=0; S+=i, i<3 ; i++) ;`

- 4) {2 т.} Годишният календар е разпределен в:

две полугодия $h \in H = \{1, 2\}$,
 всяко от които с по две тримесечия $t \in T = \{1, 2\}$,
 всяко от които с по три месеца $m \in M = \{1, 2, 3\}$.

При тези условия тройката входни данни $(h, t, m) \in H \times T \times M$ посочва един конкретен календарен месец от една невисокосна година.

Допишете съответните оператори в следващата функция на езика C така, че тя да може да изготвя справка за броя на календарните дни в посочения месец.

```
int days (int h, int t, int m)
{ int D [2][2][3] = .....
  return D .....
}
```

- 5) {4 т.} В следната функция на езика C допишете липсващото така, че при изпълнение на обръщението: magic(2); на екрана да бъде изведен текстът:

abracadabra

```
void magic ( int k )
{ char text [3][4] = { ".....",
                      ".....",
                      "....." };
  if ( k ) magic ( k-1 );
  printf ("%s", text [ k ] );
  if ( k ) magic ( k-1 );
}
```

- 6) {2 т.} Функцията concat конкатенира низовете, сочени от s1 и s2. Резултатът се сочи от s1. Попълнете липсващия израз на мястото на многоточието.

```
void concat(char *s1, const char *s2)
{ while (*s1 != '\0')
  ++s1;
  for ( ; ..... ; s1++, s2++);
}
```

- 7) {2 т.} Функцията cmpstr връща резултат 0, ако съществува двойка съответни символи от низовете, сочени от s1 и s2, които са различни, и връща резултат 1 в противния случай. Попълнете липсващия израз на мястото на многоточието.

```
int cmpstr(const char *s1, const char *s2)
{ for ( ; ..... ; s1++, s2++)
  if (*s1 != *s2)
    return 0;
  return 1;
}
```

- 8) {2 т.} Функцията enqueue добавя елемент в последователна опашка. Попълнете липсващия оператор на мястото на многоточието в следния фрагмент от програма:

```
#define M 100
struct queue
{ int f; // номер на фиктивен елемент преди
  // началото на опашката
  int r; // номер на елемента в края на опашката
  float q_array[M];
};
void enqueue(queue *q, float x)
{ .....
  if (q->r == q->f)
  { cout << "\nПрепълване\n ";
    exit(1); }
  q->q_array[q->r-1] = x;
}
```

- 9) **{2 т.}** Функцията pop изключва елемент от свързан стек. Попълнете липсващите оператори на местата на многоточията в следния фрагмент от програмата:

```
struct stack_el
{ int info;
  stack_el *link;
};
int pop(stack_el **t)
{ stack_el *p;
  int x;
  if (*t == NULL)
    { cout << "\nСтекът е празен\n";
      exit(1);}
  p = *t;
  .....
  x = p->info;
  .....
  return x;
}
```

- 10) **{2 т.}** Функцията enqueue добавя елемент в свързана опашка с последен елемент, който се сочи от г. Попълнете липсващите оператори на местата на многоточията в следния фрагмент от програмата:

```
struct queue_el
{ float info;
  queue_el *link;
};
void enqueue (queue_el *&r, float x)
{ queue_el *p = new queue_el;
  if (p == NULL)
    { cout << "\nНяма свободна памет\n";
      exit(1);}
  p->info = x;
  .....
  r->link = p;
  .....
}
```

- 11) **{2 т.}** Да се открият, обяснят и коригират грешките в следния фрагмент от програмата:

```
#include <iostream.h>
class Time{
public:
  Time()
  { hour = minute = second = 0;}
private:
  int hour;
  int minute;
  int second;
};
void main()
{ Time t;
  t.hour = 7;
  cout << t.minute;
}
```

- 12) **{4 т.}** Напишете какъв резултат извежда следващата програма. Обяснете защо програмата е пример за лош стил на програмиране.

```
#include <iostream.h>
class Time{
public:
    Time()
    { hour = minute = second = 0;}
    int getHour() { return hour;}
    int &badSetHour (int hh)
    { hour = (hh >= 0 && hh < 24) ? hh : 0;
      return hour;
    }
private:
    int hour;
    int minute;
    int second;
};
void main()
{ Time t;
  int &hourRef = t.badSetHour(20);
  cout << t.getHour() << endl;
  hourRef = 30;
  cout << t.getHour() << endl;
  t.badSetHour(12) = 74;
  cout << t.getHour() << endl;
}
```

- 13) **{2 т.}** Да се открие, обясни и коригира грешката в следния фрагмент от дефинирането на клас String и предефиниращата функция на операцията за присвояване.

```
class String{
public:
    ...
    const String &operator=(const String &);
private:
    char* sPtr; // Указател към началото на низа
    int length; // Дължина на низа
};
const String &String::operator=(const String &right)
{ delete []sPtr;
  length = right.length;
  sPtr = new char[length+1];
  assert(sPtr != 0);
  strcpy(sPtr, right.sPtr);
  return *this;
}
```

- 14) **{2 т.}** Да се открият, обяснят и коригират грешките в следния фрагмент от дефинирането на клас Date и предефиниращата функция на операцията ++ в постфиксна форма.

```
class Date{
public:
    ...
    Date &operator++(int);
private:
    int month; // Месец
    int day;   // Ден
    int year;  // Година
    void helpIncrement(); //Обслужваща функция за увеличаване
                          //на датата с единица
};
Date &Date::operator++(int)
{ Date temp = *this;
  helpIncrement();
  return temp;
}
```

15) {4 т.} Напишете какъв резултат извежда следващата програма:

```
#include <iostream.h>
class Point{
public:
    Point(float=0.0, float=0.0);
    ~Point();
protected:
    float x, y;
};
Point::Point(float a, float b)
{ x = a;
  y = b;
  cout << "Конструктор на Point: "
    << '[' << x << ", " << y << ']' << endl;
}
Point::~~Point()
{ cout << "Деструктор на Point: "
  << '[' << x << ", " << y << ']' << endl;
}
class Circle:public Point{
public:
    Circle(float r=0.0, float x=0.0, float y=0.0);
    ~Circle();
private:
    float radius;
};
Circle::Circle(float r, float a, float b):Point(a, b)
{ radius = r;
  cout << " Конструктор на Circle: " << radius
    << '[' << a << ", " << b << ']' << endl;
}
Circle::~~Circle ()
{ cout << "Деструктор на Circle: " << radius
  << '[' << x << ", " << y << ']' << endl;
}
void main()
{
    {Point p(1.1, 2.2);}
    Circle circle1(4.5, 7.2, 2.9);
    Circle circle2(10, 5, 5);
}
```

16) {2 т.} Напишете оценката на всеки от следващите изрази на езика Scheme:

```
(cdr (cadr '((a (b)) ((c (d)) e)))) → .....
(cons '(a b) (list 'c '((d) e))) → .....
(append '(a (b c)) (caddr '((a b) c ((d) e)))) → .....
(list '(a b) (list '(c d))) → .....
```

17) {1,5 т.} Попълнете липсващите изрази в дефиницията на функцията f при условие, че тя пресмята сумата на естествените числа a и b, като за целта генерира линеен итеративен процес. В липсващите изрази могат да бъдат използвани функциите add1, която прибавя 1 към числовия си аргумент, и sub1, която извежда 1 от числовия си аргумент.

```
(define (f a b)
  (if (= b 0)
      .....
      (f .....)))
```

18) **{3 т.}** Напишете оценката на всеки от следващите изрази на езика Scheme:

$((\text{lambda } (x) (\text{cons } x (\text{list } 1 \ 2))) \text{'a}) \rightarrow \dots\dots\dots$
 $(\text{append } (\text{map } \text{list } \text{'(a b c)}) (\text{map } \text{list } \text{'(x y)})) \rightarrow \dots\dots\dots$
 $(\text{map } (\text{lambda } (x) (\text{list } 1 \ x)) (\text{list } 1 \ 2 \ 3)) \rightarrow \dots\dots\dots$
 $(\text{map } (\text{lambda } (x) (* \ x \ x)) ((\text{lambda } (y) (\text{append } \text{'(1 2 3) } y)) \text{'(4 5)})) \rightarrow \dots\dots\dots$

19) **{1,5 т.}** Какво е предназначението на следната функция на езика Scheme:

```
(define (f l1 l2)
  (cond ((null? l2) '())
        ((member (car l2) l1) (f l1 (cdr l2)))
        (else (cons (car l2) (f l1 (cdr l2))))))
```

- a) намира сечението $l1 \cap l2$
- b) намира обединението $l1 \cup l2$
- c) намира разликата $l1 \setminus l2$
- d) намира разликата $l2 \setminus l1$

20) **{2 т.}** Попълнете липсващите изрази в дефиницията на функцията filter при условие, че тя връща като резултат списък от тези елементи на списъка l, които са числа и за които функцията pred връща стойност true:

```
(define (filter l pred)
  (cond ((null? l) '())
        ( ..... (cons (car l) (filter (cdr l) pred)))
        (else ..... )))
```

21) **{2 т.}** Да се състави командна процедура на езика на командния интерпретатор shell (bash) с два параметъра, първият от които е символен низ, а вторият - цяло положително число. Ако броят на терминалите, на които работи в сесия потребител с идентификатор, зададен от първия параметър, надвишава стойността на втория параметър, процедурата да изпраща на потребителя подходящо съобщение с указание за прекратяване на сесията.

22) **{2 т.}** Да се състави командна процедура на езика на командния интерпретатор shell (bash) със следното действие: ако като първи параметър на процедурата е зададено име на директория (каталог), процедурата копира всички обикновени файлове с размер, по-малък от 50 000 байта, от този каталог в поддиректория, зададена от втория параметър.

- 23) {3 т.} Да се състави командна процедура на езика на командния интерпретатор shell (bash) с три параметъра, първите два от които са десетични числа, а третият е символен низ. Процедурата периодично да прави проверка, докато броят на стартираните в рамките на текущата сесия процеси стане по-голям от първия параметър и по-малък от втория параметър. В този случай, ако броят на файловете с име, съдържащо разширение ".c", в директорията, която е зададена като трети параметър, е по-малък от 20, процедурата да стартира за всеки от тези файлове компилация във фонов режим.

- 24) {1 т.} Ако съдържанието на директория /usr е

```
a.out
abc
copy.c
copy.out
opit.out
wcount.c
xyz
```

каква стойност ще изведе на стандартния изход следната командната процедура:

```
cd /usr
a=`ls *.out | wc -l`
if [ $a -gt 2 ]
then a=2
else echo $a
fi
echo $a
```

- a) 2
b) 3
c) 1

- 25) {1 т.} Дадена е следната програма:

```
#define NO 0
#define YES 1
#include <stdio.h>
main()
{
int broi= 0, c;
int INLINE = NO;
while ( ( c=getchar() ) != EOF )
{
if ( c == '\n' ) INLINE = NO;
else {
if ( c == '*' )
if ( INLINE == NO ) ++broi ;
INLINE = YES;
}
}
printf ("\n broi= %d \n ", broi);
}
```

Като резултат от изпълнението на програмата на стандартния изход ще бъде изведен:

- a) броят срещания на символа '*' във файла
b) броят на редовете, в които '*' е първи символ
c) броят на редовете, в които '*' е последен символ

- 26) **{1,5 т.}** 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>`. Какъв алгоритъм за търсене на път `FinalPath` от даден начален възел `Start` до целевия възел `Goal` се реализира чрез дефинираното по-долу отношение `search`:
- ```
search(Goal,Goal,Path,[Goal|Path]).
search(Node,Goal,Path,FinalPath) :-
 arc(Node,NewNode),
 not member(NewNode,Path),
 search(NewNode,Goal,[Node|Path],FinalPath).
```
- a) търсене на най-добър път (*best-first search*)  
b) търсене с ограничена дълбочина (*depth-bound search*)  
c) търсене по метода на най-бързото изкачване (*hill climbing*)  
d) търсене в ширина (*breadth-first search*)  
e) търсене в дълбочина (*depth-first search*)
- 27) **{2 т.}** Дадена е задача за удовлетворяване на ограничения (*constraint satisfaction problem*), в която трябва да се намерят стойности на променливите A, B, C, D, E. Всяка от тези променливи има дефиниционна област {1,2,3,4}. Дефинирани са следните ограничения:  $(B \neq 3) \wedge (C \neq 2) \wedge (A \neq B) \wedge (B \neq C) \wedge (C < D) \wedge (A = D) \wedge (E < A) \wedge (E < B) \wedge (E < C) \wedge (E < D) \wedge (B \neq D)$ . Ако A=4, какво е множеството от допустими стойности за C?

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
(rule 1) if has_eggs(X) and has_flour(X) and
 has_water(X) and has_yeast(X) then can_make_dough(X).
(rule 2) if has_vegetables(X) then can_make_salad(X).
(rule 3) if has_vegetables(X) and has_cheese(X) and (has_ham(X) or
 has_peperoni(X)) then can_make_filling(X).
(rule 4) if can_make_dough(X) and can_make_filling(X) then
 can_make_pizza_calzone(X).
(rule 5) if has_eggs(X) and has_ham(X) then can_make_ham_and_eggs(X).
% Факти (работна памет)
(fact 1) fact(has_eggs('Ути Бъчваров')).
(fact 2) fact(has_ham('Ути Бъчваров')).
(fact 3) fact(has_flour('Ути Бъчваров')).
(fact 4) fact(has_vegetables('Ути Бъчваров')).
(fact 5) fact(has_yeast('Ути Бъчваров')).
(fact 6) fact(has_cheese('Ути Бъчваров')).
(fact 7) fact(has_water('Ути Бъчваров')).
```

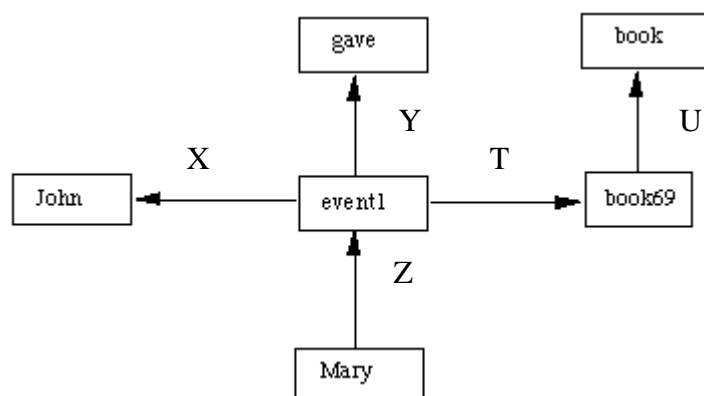
- 28) **{3 т.}** Предполага се, че интерпретаторът на правилата извършва прав извод (извод-напред, управляван от данните), като на всяка стъпка от работния си цикъл записва заключението на избраното за изпълнение правило като нов факт в края на работната памет. Какво ще съдържа работната памет след завършване 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?



- 29) {3 т.} Предполага се, че интерпретаторът на правилата извършва обратен извод (извод-назад, управляван от целите), осъществяван от отношението `is_true(Goal,List)`, като на всяка стъпка от работния си цикъл записва в списъка `List` кое правило или факт е било използвано, за да се достигне до необходимото заключение (текущата цел). Опишете номерата на правилата и фактите, които ще се съдържат в `List` (в реда на обхождането им) след изпълнението на целта:  
`?-is_true(can_make_pizza_calzone('Ути Бъчваров'),List).`

- 30) {2 т.} На фигурата е дадена семантична мрежа, която представя изречението *John gave Mary the book*. Определете връзките между възлите и попълнете таблицата (запишете знак "X" в подходящите клетки).

|                 | X | Y | Z | T | U |
|-----------------|---|---|---|---|---|
| <b>agent</b>    |   |   |   |   |   |
| <b>instance</b> |   |   |   |   |   |
| <b>instance</b> |   |   |   |   |   |
| <b>object</b>   |   |   |   |   |   |
| <b>receiver</b> |   |   |   |   |   |



- 31) {2 т.} Да се определи за кои стойности на  $n \geq 2$  е Шеферова функцията

$$f(x_1, x_2, \dots, x_n) = x_1 x_2 \dots x_n \oplus (x_1 \rightarrow x_2)(x_2 \rightarrow x_3) \dots (x_{n-1} \rightarrow x_n)(x_n \rightarrow x_1)$$

- 32) {2 т.} Да се определи дължината на СвДНФ на функцията

$$f(\tilde{x}^n) = (x_1 \vee x_2 \vee \dots \vee x_n) \rightarrow ((\dots (x_1 \sim x_2) \sim x_3) \sim \dots) \sim x_n$$

- 33) {3 т.} Проверете пълно ли е в  $F_2$  множеството от зададените по-долу булеви функции и ако е пълно, отделете в него всички бази:

$$\{0, x \oplus y, (x \rightarrow y) \downarrow (y \rightarrow z), (x \mid (xy)) \rightarrow \overline{z}\}$$

- 34) {3 т.} Нека  $L$  е език над азбуката  $A = \{a, b\}$  от думи, чиито последен символ се съдържа поне още веднъж в думата, но не на първо място. Да се построи ДКА, разпознаващ езика  $L$ , и автоматна граматика, която го поражда.

35) {1 т.} Да се построи граматика, която поражда всички булеви думи, съдържащи нечетен брой единици.

36) {2 т.} Коя от следните формули е предикатна тавтология?

- a)  $(\forall X p(X) \Rightarrow \forall X q(X)) \Rightarrow \forall X (p(X) \Rightarrow q(X))$
- b)  $\forall X (p(X) \Rightarrow q(X)) \Rightarrow (\forall X p(X) \Rightarrow \forall X q(X))$
- c)  $\exists X (p(X) \Rightarrow q(X)) \Rightarrow (\exists X p(X) \Rightarrow \exists X q(X))$
- d)  $\exists X (p(X) \& q(X)) \Rightarrow (\exists X p(X) \Rightarrow \forall X q(X))$

37) {2 т.} Нека  $\Gamma$  е следното множество от формули:

$\Gamma = \{\exists X_1 p(X_1), \neg p(X_1), \neg p(X_2), \dots, \neg p(X_n), \dots\}$ , където  $x_1, x_2, \dots$  са различни променливи.

Кое от следните твърдения е вярно:

- a)  $\Gamma$  е изпълнимо и няма модел
- b)  $\Gamma$  е изпълнимо и има модел
- c)  $\Gamma$  не е изпълнимо и няма модел
- d)  $\Gamma$  не е изпълнимо и има модел

38) {3 т.} Нека е дадена следната програма на езика Prolog:

$s(X, [A|L], [A|Y], Z) :- A \leq X, s(X, L, Y, Z).$

$s(X, [A|L], Y, [A|Z]) :- X < A, s(X, L, Y, Z).$

$s(\_, [], [], []).$

$q([H|T], S, X) :- s(H, T, A, B), q(A, S, [H|Y]), q(B, Y, X).$

$q([], X, X).$

При цел  $?- q([2,3,1], S, []).$ , кой от следните отговори ще изведе интерпретаторът на Prolog?

- a)  $S = [2, 1, 3]$
- b)  $S = [1, 2, 3]$
- c)  $S = [3, 2, 1]$
- d)  $S = [1, 3, 2]$

39) {1 т.} Нека е дадена следната програма на езика Prolog:

$p(X, Z) :- q(X, Y), p(Y, Z).$

$p(X, X).$

$q(a, b).$

Нека целта, зададена на интерпретатора, е  $?- p(X, b).$

Кое от следните твърдения е вярно:

- a) При преудовлетворяване интерпретаторът на Prolog ще изведе последователно  $X = a$  и  $X = b$
- b) Интерпретаторът на Prolog ще изведе отговор "no"
- c) Интерпретаторът на Prolog ще зацikli
- d) Интерпретаторът на Prolog ще изведе само  $X = a$

40) {1 т.} Нека  $P$  е следната програма:

$p(c).$

$p(f(f(X))) :- p(X).$

Нека  $M = \{p(f^{2^n}(c)) : n \in \mathbb{N}\} \cup \{p(f(c))\}$  и  $M_p$  е най-малкият ербранов модел на  $P$ .

Кое от следните твърдения е вярно:

- a)  $M = M_p$
- b)  $M$  е модел на  $P$
- c)  $M \subseteq M_p$
- d)  $\alpha \in M$  точно тогава, когато  $P \models \alpha$

41) {1 т.} Даден е операторът  $\Gamma: \Phi_1 \rightarrow \Phi_1$

$$\Gamma(f)(x) \cong \begin{cases} x/3, & \text{ако } x \text{ се дели на } 3 \\ f(f(x+1)), & \text{в противен случай} \end{cases}$$

Ако  $f_\Gamma$  е най-малката неподвижна точка на  $\Gamma$ ,  $x > 0$  и  $f_\Gamma(x) \cong y$ , то

- a)  $y < x$
- b)  $y = x/3$
- c)  $y = x+3$
- d)  $y > x$

42) {2 т.} Нека R е следната рекурсивна програма:

R: F(X), where

F(X) = if X=1 then 1

else if X = 0 (2) then  $F(\frac{X}{2})$

else F(X-1)+1

Нека !  $D_V(R)(x)$  и  $D_V(R)(x) \cong y$ .

Кое от следните твърдения е вярно:

- a) y = броя на нулите в двоичния запис на x
- b) y = броя на единиците в двоичния запис на x
- c) y = броя на нулите и единиците в двоичния запис на x
- d)  $y = [\frac{x}{2}] + 1$

43) {2 т.} Нека  $\Gamma: \Phi_n \rightarrow B_{\perp}$  и  $B_{\perp} = \langle \{tt, ff, \perp\}, \mu, \perp \rangle$  е област на Скот с плоската наредба  $\mu$  и

$$\Gamma(\varphi) = \begin{cases} tt, \text{ ако } \varphi \text{ е тотална} \\ \perp, \text{ в противния случай} \end{cases}$$

Кое от следните твърдения е вярно:

- a)  $\Gamma$  е монотонен и не е непрекъснат
- b)  $\Gamma$  е монотонен и непрекъснат
- c)  $\Gamma$  не е монотонен и е непрекъснат
- d)  $\Gamma$  не е монотонен и не е непрекъснат

44) {3 т.} Нека R е следната рекурсивна програма:

R  $\left\{ \begin{array}{l} F1(X, X, 0), \text{ where} \\ F1(X, Y, Z) = \text{if } X=0 \text{ then } Z \\ \quad \text{else } F1(X-1, Y, Z+F2(X, Y)) \\ F2(X, Y) = \text{if } Y=0 \text{ then } 0 \\ \quad \text{else } X+F2(X, Y-1) \end{array} \right.$

Ако  $D_V(R)(x) \cong y$ , то

- a)  $y = \frac{x(x+1)}{2}$
- b)  $y = \frac{(x+1)^2}{2}$
- c)  $y = \frac{x^2(x+1)}{2}$
- d)  $y = \frac{x(x+1)^2}{2}$

45) {3 т.} Дадена е следната рекурсивна програма:

R  $\left\{ \begin{array}{l} F1(X), \text{ where} \\ F1(X) = \text{if } X>50 \text{ then } X-17 \\ \quad \text{else } F1(F2(X+18)) \\ F2(X) = \text{if } X>50 \text{ then } X-17 \\ \quad \text{else } F2(F1(X+18)) \end{array} \right.$

Ако  $x \leq 51$  и  $D_V(R)(x) \cong y$ , то кое от следните твърдения е вярно:

- a) y = 50
- b) y = 51
- c) y = 34
- d) y = 31

**Задача 2**

Фирма предлага на пазара продукти, доставяни от различни доставчици. За всеки продукт се поддържа следната информация: уникален код, тип на продукта, колко бройки са в наличност в склада (stock\_quantity), цена за единична бройка, отстъпка. Всеки продукт се доставя от един доставчик. Всеки доставчик може да доставя нито един, един или повече продукти. За доставчиците се съхранява информация за лице за контакти, код на областта, телефон, щат. За доставките се съхранява следната информация: уникален код, дата на доставка, кой продукт от кой доставчик и в какво количество е доставен.

Базата **delivery** съдържа три таблици:



В таблица product има следните данни:

| productID | type      | stock_quantity | price  | discount |
|-----------|-----------|----------------|--------|----------|
| A102      | food      | 25             | 100.50 | 0.05     |
| A104      | food      | 8              | 53.75  | 0.1      |
| A250      | food      | 40             | 75.50  | 0.05     |
| B349      | drink     | 80             | 34.00  | 0        |
| B401      | drink     | 53             | 25.00  | 0        |
| B567      | drink     | 8              | 15.00  | 0.1      |
| B709      | drink     | 6              | 65.00  | 0.07     |
| C444      | beverage  | 84             | 8.00   | 0.1      |
| C458      | beverage  | 28             | 20.00  | 0.05     |
| D476      | vegetable | 65             | 27.50  | 0.15     |
| D897      | vegetable | 45             | 89.50  | 0.12     |
| D900      | vegetable | 38             | 8.00   | 0.08     |

В таблица vendor има следните данни:

| vendorID | contactperson | areacode | phone    | state |
|----------|---------------|----------|----------|-------|
| 21222    | Smitson       | 972      | 666-8888 | TX    |
| 21232    | Andersen      | 817      | 222-7777 | TX    |
| 22000    | Antonio       | 304      | 888-9999 | FL    |
| 23002    | Settenen      | 302      | 111-2222 | TX    |
| 25489    | Boyce         | 212      | 333-4444 | FL    |

В таблица deliveries има следните данни:

| deliveryID | dateofdelivery | productID | vendorID | quantity |
|------------|----------------|-----------|----------|----------|
| 120        | 9/6/2003       | A250      | 21222    | 20       |
| 121        | 10/6/2003      | A104      | 25489    | 10       |
| 122        | 10/6/2003      | B709      | 25489    | 50       |
| 123        | 11/6/2003      | C444      | 21232    | 14       |
| 124        | 11/6/2003      | D897      | 25489    | 60       |

- 1) **{1 т.}** За базата от данни delivery кой от следните оператори ще изведе кодовете на продукти, доставяни от доставчик с код 25489 или 21222?
- a) `SELECT productID, vendorID  
FROM deliveries  
WHERE vendorID IN (25489, 21222);`
  - b) `SELECT productID, vendorID  
FROM deliveries  
WHERE vendorID IS (25489, 21222);`
  - c) `SELECT productID, vendorID  
FROM deliveries  
WHERE vendorID ARE (25489, 21222);`
  - d) `SELECT productID, vendorID  
FROM deliveries  
WHERE vendorID = (25489, 21222);`
- 2) **{1 т.}** За базата от данни delivery кой от следните оператори ще вмъкне нов запис за продукт с код A103?
- a) `UPDATE product  
SET stock_quantity = 23  
WHERE productID = 'A103';`
  - b) `INSERT INTO product  
VALUES ('A103', 'food', 23, 100.5, 0.05);`
  - c) `DELETE  
FROM product  
WHERE productID = 'A103';`
  - d) `SELECT stock_quantity  
FROM product  
WHERE productID = 'A103';`
- 3) **{1 т.}** За базата от данни delivery кой от следните оператори ще изведе броя на наличните продукти?
- a) `SELECT SUM(*)  
FROM product;`
  - b) `SELECT MAX(*)  
FROM product;`
  - c) `SELECT COUNT(*)  
FROM product;`
  - d) `SELECT AVG(*)  
FROM product;`
- 4) **{1 т.}** За базата от данни delivery кой от следните оператори ще изведе кодовете на продукти, започващи с B и завършващи с 9?
- a) `SELECT productID  
FROM product  
WHERE productID LIKE 'B9';`
  - b) `SELECT productID  
FROM product  
WHERE productID = 'B#9';`
  - c) `SELECT productID  
FROM product  
WHERE productID IS 'B%9';`
  - d) `SELECT productID  
FROM product  
WHERE productID LIKE 'B%9';`

5) {1 т.} За базата от данни delivery кой от следните оператори ще изведе средната цена на продуктите от тип beverage?

- a) 

```
SELECT SUM(price)
FROM product
GROUP BY product.type;
```
- b) 

```
SELECT MIN(price)
FROM product
GROUP BY product.type
WHERE type='beverage';
```
- c) 

```
SELECT AVG(price)
FROM product
GROUP BY product.type
HAVING type='beverage';
```
- d) 

```
SELECT COUNT(price)
FROM product
GROUP BY product.type
HAVING type='beverage';
```