

Тип экзамена:

סוג הבחינה:

а. на аттестат зрелости для средних школ

א. בגרות לבתי ספר על-יסודיים

б. на аттестат зрелости для экстернов

ב. בגרות לנבחנים חיצוניים

Время проведения экзамена: лето 2015 года

מועד הבחינה: קיץ תשע"ה, 2015

Номер вопросника: 899205, 603

מספר השאלון: 899205, 603

Перевод на русский язык (5)

תרגום לרוסית (5)

Компьютеры "бет"

2 единицы обучения (дополнение до 5-ти единиц)

מדעי המחשב ב'

2 יחידות לימוד (השלמה ל-5 יח"ל)

Указания экзаменуемым

а. Продолжительность экзамена: три часа.

б. Строение вопросника и ключ к оценке:

в этом вопроснике два раздела.

Раздел первый: в этом разделе четыре вопроса.

Вы должны ответить на два из них.

$$(2 \times 25) = 50 \text{ баллов}$$

Раздел второй: в этом разделе вопросы из четырех различных курсов. Вы должны ответить только

на вопросы курса, который вы изучали, согласно

указаниям этого курса. $(2 \times 25) = 50 \text{ баллов}$

$$\text{Всего} - 100 \text{ баллов}$$

в. Разрешенный вспомогательный материал:

любой вспомогательный материал, за исключением компьютера с возможностью программирования.

г. Особые указания:

1. Все программы, которые вы должны написать на языке программирования в первом разделе, следует писать только на одном языке – Java или C# .

2. На внешней обложке тетради укажите язык программирования, который вы используете: Java или C#.

3. На внешней обложке тетради укажите название курса, который вы изучали, одного из четырех следующих курсов: "Компьютерные системы и ассемблер" [מערכות מחשב ואסמבלי], "Введение в исследование операций" [מבוא לחקר ביצועים], "Вычислительные модели" [מודלים חישוביים], "Объектно-ориентированное программирование" [תכנות מונחה עצמים].

Примечание: в программах, которые вы должны написать, за использование большой буквы вместо маленькой или наоборот баллы сниматься не будут.

כתוב במחברת הבחינה בלבד, בעמודים נפרדים, כל מה שברצונך לכתוב כטייפ (ראשי פרקים, חישובים וכדומה). רשום "טייפ" בראש כל עמוד טייפ. רישום טייפות לשהן על דפים מחוץ למחברת הבחינה עלול לגרום לפסילת הבחינה!

Все черновые записи (тезисы, вычисления и т.п.) делайте только на отдельных страницах экзаменационной тетради. Напишите слово «черновик» в начале каждой страницы, отведенной вами под черновик. Выполнение любых черновых записей на листах, не относящихся к экзаменационной тетради, может привести к тому, что экзамен будет аннулирован!

Желаем успеха!

בהצלחה!

הוראות לנבחן

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח ההערכה:

בשאלון זה שני פרקים.

פרק ראשון – בפרק זה ארבע שאלות,

ומהן יש לענות על שתיים.

$$(25 \times 2) - 50 \text{ נקודות}$$

פרק שני – בפרק זה שאלות בארבעה

מסלולים שונים, ענה על שאלות רק

במסלול שלמדת, לפי ההוראות בקבות

השאלות במסלול זה.

$$(25 \times 2) - 50 \text{ נקודות}$$

$$\text{סה"כ} - 100 \text{ נקודות}$$

ג. חומר עזר מותר בשימוש:

כל חומר עזר, חוץ ממחשב הניתן לתכנות.

ד. הוראות מיוחדות:

1. את כל התכניות שאתה נדרש לכתוב

בשפת מחשב בפרק הראשון כתוב

בשפה אחת בלבד –

Java או C#.

2. רשום על הכריכה החיצונית של

המחברת באיזו שפה אתה כותב –

Java או C#.

3. רשום על הכריכה החיצונית של

המחברת את שם המסלול שלמדת,

המסלול הוא אחד מארבעת המסלולים

האלה: מערכות מחשב ואסמבלי,

מבוא לחקר ביצועים, מודלים

חישוביים, תכנות מונחה עצמים.

הערה: בתכניות שאתה כותב לא יורדו לך

נקודות, אם תכתוב אות גדולה

במקום אות קטנה או להפך.

Вопросы

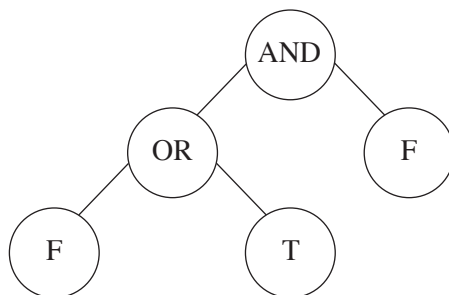
В данном вопроснике два раздела: **раздел первый** и **раздел второй**.
Вы должны ответить на вопросы обоих разделов, согласно инструкциям каждого раздела.

РАЗДЕЛ ПЕРВЫЙ (50 баллов)

Внимание: во всех вопросах, где требуется реализация [מימוש], разрешается пользоваться методами классов Список, Очередь, Стек, Двоичное дерево и Узел [רשימה, תור, מחסנית, עץ בינארי וחוליה], не реализуя их. Если вы используете дополнительные методы, вы должны их реализовать [לממש אותם].

Ответьте на два из вопросов 1-4 (за каждый вопрос – 25 баллов).

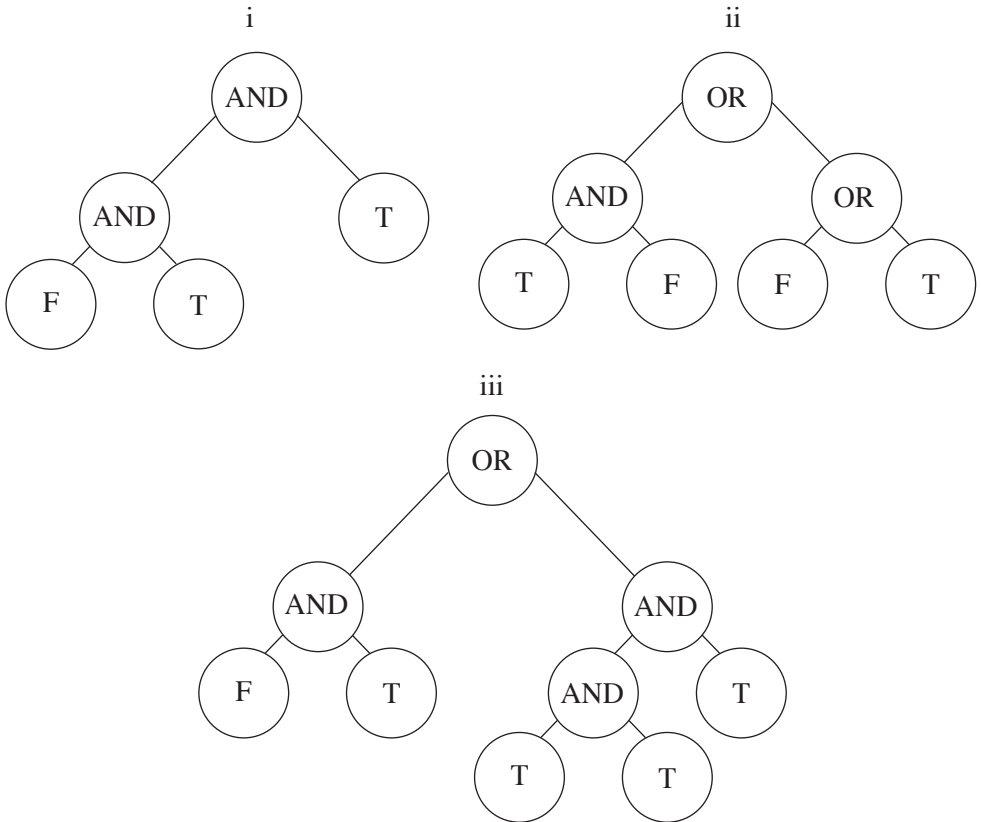
1. **Дерево логического выражения** – это непустое двоичное дерево [עץ בינארי] типа строка [מחרוזת], представляющее логическое выражение. Каждый лист дерева содержит одну из строк "T" или "F". Строка "T" обозначает true, а строка "F" – false.
- Каждая вершина, не являющаяся листом, содержит одну из строк "AND" или "OR".
- Строка "AND" обозначает логический оператор (логическое действие) "и", а строка "OR" – логический оператор "или".
- У каждой вершины, не являющейся листом, есть два сына.
- Чтобы рассчитать логическое выражение, представленное деревом, логическое выражение в вершине, не являющейся листом, применяют к значениям, полученным из левого и правого под-дерева данной вершины.
- Например, приведенное ниже дерево представляет собой логическое выражение $((F \text{ OR } T) \text{ AND } (F))$, значение которого false .



/продолжение на странице 3/

(א) Перед вами 3 двоичных дерева i-iii, каждое из которых является **деревом логического выражения**.

Для каждого из деревьев запишите логическое выражение, представленное деревом, и значение этого выражения.



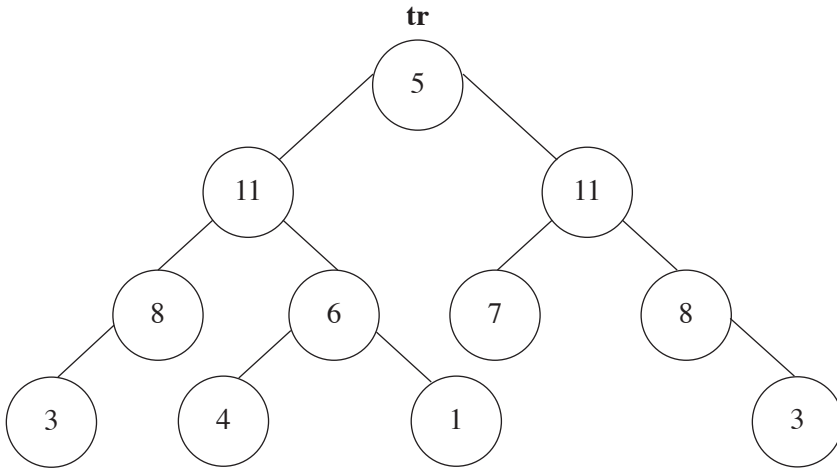
(ב) Напишите на языке Java или C# внешний метод [פעולה חיצונית], который получит **дерево логического выражения** и вернет логическое значение (true или false) выражения, определяемого данным деревом.

/продолжение на странице 4/

2. В этом вопросе два пункта א-ב, между которыми нет связи.

Ответьте на вопросы обоих пунктов.

(א) Дано двоичное дерево [עץ בינארי] tr.



Перед вами метод wrap на языке Java, использующий метод branch, и метод Wrap на языке C#, использующий метод Branch.

Обратите внимание: метод написан в двух вариантах. Ответьте на вопросы (1)-(2) в соответствии с вариантом, который вы изучали.

Вариант для решающих по текущей программе (Java и C#) на странице 5.

Вариант для решающих по новой программе (Java и C#) на странице 6.

(1) Проследите за выполнением метода wrap на языке Java и метода Wrap на языке C# для данного дерева tr, и запишите возвращаемое значение.

Покажите рекурсивную трассировку [דפוס].

(2) Приведите пример двоичного дерева, для которого метод wrap на языке Java и метод Wrap на языке C# вернет true, и пример двоичного дерева, для которого этот метод вернет false. Каждое из этих деревьев должно содержать ровно 5 вершин.

/продолжение на странице 5/

Для решающих по текущей программе:

Java

```
public static boolean wrap(BinTreeNode<Integer> tr)
{
    return branch(tr.getLeft(), tr.getRight());
}
public static boolean branch(BinTreeNode<Integer> t1, BinTreeNode<Integer> t2)
{
    if ((t1 == null) && (t2 == null))
        return true;
    if (((t1 != null) && (t2 == null)) || ((t1 == null) && (t2 != null)))
        return false;
    return ((t1.getInfo() == t2.getInfo()) && branch(t1.getLeft(), t2.getRight()));
}
```

C#

```
public static bool Wrap(BinTreeNode<int> tr)
{
    return Branch(tr.GetLeft(), tr.GetRight());
}
public static bool Branch(BinTreeNode<int> t1, BinTreeNode<int> t2)
{
    if ((t1 == null) && (t2 == null))
        return true;
    if (((t1 != null) && (t2 == null)) || ((t1 == null) && (t2 != null)))
        return false;
    return ((t1.GetInfo() == t2.GetInfo()) && Branch(t1.GetLeft(), t2.GetRight()));
}
```

Обратите внимание: пункт (з) этого вопроса на странице 7.

/продолжение на странице 6/

Для решающих по новой программе:

Java

```
public static boolean wrap(BinNode< Integer > tr)
{
    return branch(tr.getLeft(), tr.getRight());
}
public static boolean branch(BinNode<Integer> t1, BinNode<Integer> t2)
{
    if ((t1 == null) && (t2 == null))
        return true;
    if (((t1 != null) && (t2 == null)) || ((t1 == null) && (t2 != null)))
        return false;
    return ((t1.getValue() == t2.getValue()) && branch(t1.getLeft(), t2.getRight()));
}
```

C#

```
public static bool Wrap(BinNode<int> tr)
{
    return Branch(tr.GetLeft(), tr.GetRight());
}
public static bool Branch(BinNode<int> t1, BinNode<int> t2)
{
    if ((t1 == null) && (t2 == null))
        return true;
    if (((t1 != null) && (t2 == null)) || ((t1 == null) && (t2 != null)))
        return false;
    return ((t1.GetValue() == t2.GetValue()) && Branch(t1.GetLeft(), t2.GetRight()));
}
```

Обратите внимание: пункт (2) этого вопроса на следующей странице.

/продолжение на странице 7/

(ב) Нет связи с пунктом (א).

Перед вами фрагмент [קטע] программы на языке Java и на языке C# .

Дано, что a – массив [מערך] целого типа длиной n .

```
int i = 0;
while (i < n - 1)
{
    if (a[i] > a[i+1])
    {
        a[i] = a[i] + a[i+1];
        a[i+1] = a[i] - a[i+1];
        a[i] = a[i] - a[i+1];
        i = 0;
    }
    else
        i++;
}
```

(1) Проследите за выполнением данного фрагмента для приведенного ниже массива a длиной 4:

a	5	7	8	12
-----	---	---	---	----

Трассировка должна отображать i и массив в конце каждой итерации.

(2) Какова временная сложность [סיבוכיות זמן הריצה] данного фрагмента для массива длиной n , отсортированного по возрастанию? Обоснуйте.

(3) Проследите за выполнением данного фрагмента для приведенного ниже массива a длиной 4 :

a	12	8	7	5
-----	----	---	---	---

Трассировка должна отображать i и массив в конце каждой итерации.

(4) Какова временная сложность [סיבוכיות זמן הריצה] данного фрагмента для массива длиной n , отсортированного по убыванию? Обоснуйте.

/продолжение на странице 8/

3. Класс Автобусный Маршрут – **BusRoute** описывает маршрут автобуса. Каждый маршрут содержит некоторое число остановок – не менее двух – в определенном порядке. Каждая остановка представлена двумя целыми числами, отражающими ее расположение на плоскости. Каждая остановка присутствует в маршруте один раз.
- Например, маршрут из пяти остановок начинается на остановке (0, 2) и заканчивается на остановке (5, 0) : $(0, 2) \rightarrow (1, 4) \rightarrow (5, 4) \rightarrow (3, 1) \rightarrow (5, 0)$
- Автобус едет от одной остановки к другой, по порядку.
- Длина маршрута – это сумма расстояний между остановками.
- Считайте, что дан класс Остановка – **Station**, атрибутами [תכונה] которого являются два целых числа x и y , отражающими расположение остановки на плоскости. В классе **Station** определены два метода: конструктор `Station(int x, int y)` и метод с заголовком: `double distance(Station other)` – на языке Java `double Distance(Station other)` – на языке C# :
- Метод получает остановку `other` и возвращает расстояние между данной остановкой и остановкой `other`.
- Перед вами интерфейс [פרמףס] класса Автобусный Маршрут – **BusRoute**, написанный на языке Java или на языке C#.

Java

<code>BusRoute(Station first, Station second)</code>	конструктор, получающий две остановки и создающий маршрут из этих двух остановок
<code>void addStation(Station newStation)</code>	метод, получающий остановку и добавляющий ее к концу существующего маршрута. Считайте, что такой остановки нет в данном маршруте
<code>double routeLength()</code>	метод, возвращающий длину данного маршрута, т.е. сумму расстояний от остановки к остановке

C#

<code>BusRoute(Station first, Station second)</code>	конструктор, получающий две остановки и создающий маршрут из этих двух остановок
<code>void AddStation(Station newStation)</code>	метод, получающий остановку и добавляющий ее к концу существующего маршрута. Считайте, что такой остановки нет в данном маршруте
<code>double RouteLength()</code>	метод, возвращающий длину данного маршрута, т.е. сумму расстояний от остановки к остановке

Обратите внимание: пункты этого вопроса на следующей странице

/продолжение на странице 9/

- (א) Напишите на языке Java или на языке C# заголовок [כותרת] класса **BusRoute** и его атрибут/атрибуты. Опишите назначение [תיעוד] каждого атрибута.
- (ב) Реализуйте [ממש] на языке Java или на языке C# конструктор класса [פעולה בונה] **BusRoute**.
- (ג) Реализуйте [ממש] на языке Java или на языке C# метод [פעולה], добавляющий остановку к концу существующего маршрута.
- (ד) Реализуйте [ממש] на языке Java или на языке C# метод, возвращающий длину маршрута.
- (ה) Напишите на языке Java или на языке C# в главном методе класса Program фрагмент программы, который создаст автобусный маршрут из примера в начале вопроса и выведет на экран длину этого маршрута.

Примечание: можно использовать методы класса **Station**, не реализуя их.

4. В фирме "Эффективная работа" существуют следующие категории заданий: сверхсрочные задания, срочные задания, обычные задания. Сверхсрочные задания выполняются первыми, после них выполняются срочные задания, и последними выполняются обычные задания. Фирма пользуется компьютерной системой **Администратор заданий**. Добавление заданий в **Администратор заданий** происходит по следующим правилам:
- новое сверхсрочное задание добавляется перед всеми другими – сверхсрочных, срочных и обычных – заданий, находящихся в данный момент в **Администраторе заданий**
 - новое срочное задание добавляется после всех сверхсрочных заданий и перед срочными и обычными заданиями, находящимися в данный момент в **Администраторе заданий**
 - новое обычное задание добавляется после всех других – сверхсрочных, срочных и обычных – заданий, находящихся в данный момент в **Администраторе заданий**.

Задания удаляются (и переходят к выполнению) из **Администратора заданий** в том порядке, в котором они там расположены.

Дан класс **Задание – Task**, с двумя атрибутами [תכונות]:

content – строка [מחרוזת] с описанием задания

code – целое число, обозначающее срочность:

1 обозначает сверхсрочное задание, 2 обозначает срочное задание,

3 обозначает обычное задание.

Считайте, что для каждого из атрибутов определены методы set и get на языке Java или Set и Get на языке C#.

Вы должны реализовать класс **Администратор заданий** при помощи любого числа стеков [מחסניות] и очередей [תורים], так чтобы временная сложность выполнения [סיבוכיות זמן הריצה] методов добавления в **Администратор заданий** и методов удаления из **Администратора заданий** была $O(1)$.

Обратите внимание: продолжение вопроса на следующей странице.

/продолжение на странице 11/

- (א) Напишите на языке Java или на языке C# заголовок класса **Администратор заданий** и его атрибуты. Напишите документацию для каждого атрибута.
- (ב) Напишите на языке Java или на языке C# конструктор [פעולה בונה] без параметров для класса **Администратор заданий**.
- (ג) Напишите на языке Java или на языке C# в классе **Администратор заданий** метод, получающий **Задание** и добавляющий его в **Администратор заданий** в соответствии с правилами, приведенными в начале вопроса.
- (ד) Напишите на языке Java или на языке C# в классе **Администратор заданий** метод, который удалит (и переведет к выполнению) очередное **задание** и вернет его.
Если в **Администраторе заданий** нет **заданий**, метод вернет null.

РАЗДЕЛ ВТОРОЙ (50 баллов)

В данном разделе вопросы четырех разных курсов:

- Компьютерные системы и ассемблер [מערכות מחשב ואסמבלי], стр. 12-16.
- Введение в исследование операций [מבוא לחקר ביצועים], стр. 17-29.
- Вычислительные модели [מודלים חישוביים], стр. 30-34.
- Объектно-ориентированное программирование [תכנות מונחה עצמים] на языке Java, стр. 35-44; объектно-ориентированное программирование на языке C#, стр. 45-54.

Ответьте только на вопросы того курса, который вы изучали.

Компьютерные системы и Ассемблер [מערכות מחשב ואסמבלי]

Если вы изучали этот курс, ответьте на два вопроса из вопросов 5-8 (за каждый вопрос – 25 баллов).

5. Перед вами фрагмент [קטע] программы на Ассемблере.

```
MOV  BX , 3000H
MOV  CX , 5
AGAIN: MOV  AL , CL
      CMP  AL , 3
      JNZ  NEXT          ;(*)
      MOV  [BX] , CL
      INC  BX
NEXT:  LOOPZ  AGAIN
SHOOV: MOV  AL , CL
      CMP  AL , 3
      JNZ  NEXT1
      MOV  [BX] , CL
      INC  BX
NEXT1: LOOP  SHOOV        ;(**)
      NOP
```

Обратите внимание: пункты этого вопроса на следующей странице

/продолжение на странице 13/

(א) Проследите с помощью таблицы трассировки [טבלת מעקב]

за выполнением данного фрагмента. Таблица трассировки должна
содержать столбец для каждого из регистров [אוגרים] AL , BX , CL .

Ответьте на вопросы пунктов (ב)-(ד). Пункты не связаны друг с другом,
и каждый из них относится к фрагменту программы, приведенному в
начале вопроса.

(ב) Определите, сколько байтов будет записано в памяти, начиная
с ячейки памяти с адресом 3000H, после выполнения данного
фрагмента.

(ג) Из данного фрагмента удалили команду, отмеченную (*).
Определите, что будет записано в памяти, начиная с ячейки памяти
с адресом 3000H, после выполнения измененного фрагмента.

(ד) Ответьте на основании фрагмента программы, приведенного в начале
вопроса и включающего в себя команду, отмеченную (*).

Вместо строки, отмеченной (**) была записана строка
NEXT1: LOOPNE SHOOV

Определите, что будет записано в памяти, начиная с ячейки памяти
с адресом 3000H, после выполнения измененного фрагмента.

6. В данном вопросе два не связанных между собой пункта (א)-(ב). Ответьте на вопросы обоих этих пунктов.

(א) В таблице перед вами записаны команды, составляющие фрагмент [עבר] программы на Ассемблере.

Скопируйте данную таблицу в вашу тетрадь и допишите в каждом из столбцов состояние флагов после выполнения каждой из команд.

Считайте, что перед выполнением данного фрагмента все три флага SF, ZF, CF обнулены, а BL содержит значение -1.

команды	CF	ZF	SF
MOV AL, 3H			
CMP AL, 3H			
CMP AL, 2H			
CMP AL, 5H			
XOR AL, AL			
DEC AL			
MUL BL			

(ב) Нет связи с пунктом (א).

(1) Напишите на Ассемблере процедуру, которая получит в качестве параметра через стек [מחסנית] число от 0 до 255.

Процедура сохранит в регистре [אח] AX количество единиц (1), имеющееся в двоичной (бинарной) форме записи полученного числа.

(2) Дан массив ARR из 100 элементов, каждый длиной в один байт. Данный массив содержит числа. Напишите на Ассемблере фрагмент программы, который сохранит в регистре CX количество элементов массива, в двоичной форме записи которого нечетное количество единиц.

Вы должны использовать процедуру, написанную в подпункте (1).

7. В данном вопросе два не связанных между собой пункта (א)-(ב). Ответьте на вопросы обоих этих пунктов.

(א) Для каждого из приведенных ниже утверждений (1)-(10) определите, верно оно или нет. Если утверждение неверно, поясните почему.

(1) Перед вами фрагмент программы на Ассемблере

PUSH AL

POP AH

Данный фрагмент записывает в AH значение, хранящееся в AL .

(2) При применении метода дополнения до двух возможно при помощи 8 битов записывать числа от -128 до $+127$.

(3) Приведенная ниже команда копирует в BX значение адреса ячейки памяти, хранящегося в AX.

MOV BX , [AX]

(4) Команда CALL меняет SI .

(5) Команда RET увеличивает значение SP .

(6) Команда RET всегда уменьшает значение IP .

(7) Перед вами фрагмент программы на Ассемблере, в котором процедура DONOTHING не выполняет никаких действий и не изменяет значения регистров [אוגרים].

MOV CX , NUMBEROFTIMES

LP: CALL DONOTHING

LOOP LP

Если NUMBEROFTIMES равно нулю, процедура DONOTHING будет вызываться один раз.

(8) Приведенная ниже команда обнуляет BX .

XOR BX , BX

(9) Приведенный ниже фрагмент программы обнуляет DL .

MOV CL , 8

SHR DL , CL

(10) Во фрагменте данных [מקטע הנתונים] определены переменные NUM1 , NUM 2 .

NUM1 DB 100

NUM2 DB -156

Представление этих переменных в памяти идентично.

Обратите внимание: пункт (2) этого вопроса на следующей странице.

/продолжение на странице 16/

(ב) Нет связи с пунктом (א).

Перед вами фрагмент [קטב] программы на Ассемблере.

```
MOV    AL, -1
MOV    DL, 00001111B    ;(*)
MOV    CL, 4
AND    AL, DL
SHR    AL, CL
```

(1) Проследите с помощью таблицы трассировки [טבלת מעקב] за выполнением данного фрагмента и запишите, каким будет значение в регистре AL после выполнения.

Таблица трассировки должна содержать столбец для каждого из регистров: AL, CL, DL.

(2) В строке, отмеченной (*), вместо числа 00001111B записали число 00001111. Будет ли данный фрагмент переведен в машинный код без ошибок? Обоснуйте ваш ответ.

8. (א) Напишите на Ассемблере процедуру с именем FINDSECOND, которая получит через стек два параметра: указатель на массив байтов и число элементов в данном массиве.

Процедура FINDSECOND запишет в регистр AX второе по величине значение в массиве. Например для массива

03H	08H	0CH	01H	09H
-----	-----	-----	-----	-----

в регистр AX будет помещено значение 09H.

Считайте, что

- числа в массиве положительные,
- числа в массиве отличны друг от друга,
- массив содержит по меньшей мере два числа.

(ב) Во фрагменте данных [מקטב נתונים] определен массив ARR из 100 элементов, каждый длиной в один байт. Данный массив состоит из чисел.

Напишите на Ассемблере фрагмент [קטב] программы, который выведет на экран [ידפס] второе по величине число из массива, представленное в десятичной системе исчисления.

Считайте, что числа в массиве положительные и отличны друг от друга. Вы должны использовать процедуру FINDSECOND, написанную для пункта (א).

/продолжение на странице 17/

Введение в исследование операций [מבוא לחקר ביצועים]

Если вы изучали этот курс, ответьте на два из вопросов 9-12
(за каждый вопрос – 25 баллов).

9. В данном вопросе шесть не связанных между собой пунктов (א)-(ו).

Ответьте на вопросы всех этих пунктов.

(א) Даны ограничения:

$$6x_1 + 3x_2 \leq 18$$

$$-4x_1 + 2x_2 \leq 4$$

$$x_1 + x_2 \geq 3$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

Начертите в вашей тетради область возможных решений,
определяемую на основе данных ограничений. Обозначьте область
возможных решений при данных ограничениях и запишите на
полученном графике значения x_1 и x_2 для каждой из вершин области
возможных решений.

**Все пункты (ב)-(ו) относятся к приведенной ниже задаче линейного
программирования.**

Задача линейного программирования:

$$\max \{z = 20x_1 + 15x_2\}$$

при следующих ограничениях:

$$2x_1 + x_2 \leq 6$$

$$-2x_1 + x_2 \leq 2$$

$$x_1 + x_2 \leq b$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

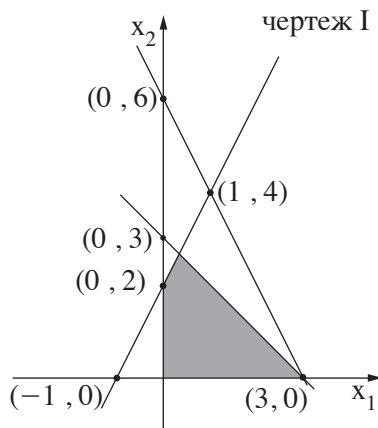
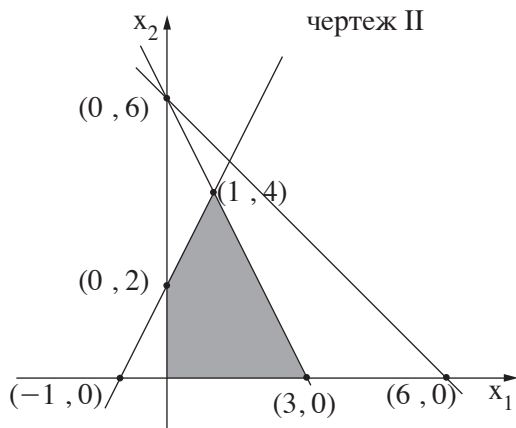
b – параметр.

Для $b = 3$ область возможных решений представлена на чертеже I.

Для $b = 6$ область возможных решений представлена на чертеже II.

Обратите внимание: продолжение вопроса на следующей странице.

/продолжение на странице 18/



- (ב) Для $b = 3$, каково оптимальное решение данной задачи линейного программирования? Обоснуйте свое утверждение.
- (ג) Для $b = 6$, оптимальное решение данной задачи – это $(1, 4)$. Объясните почему.
- (ד) Найдите два значения b (отличные от 6), для которых оптимальное решение данной задачи – это $(1, 4)$. Обоснуйте свой ответ.
- (ה) В первом ограничении задачи, записанной перед пунктом (ב), изменяют символ неравенства на противоположный, получают ограничение: $2x_1 + x_2 \geq 6$.
Для $b = 3$ начертите область возможных решений.
Каково оптимальное решение задачи, полученной в результате данного изменения? Обоснуйте свой ответ.
- (ו) В первых двух ограничениях задачи линейного программирования, данной перед пунктом (ב), изменяют символ неравенства на противоположный.
Для $b = 3$ ограничения после изменения таковы:

$$2x_1 + x_2 \geq 6$$

$$-2x_1 + x_2 \geq 2$$

$$x_1 + x_2 \leq 3$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

Существует ли оптимальное решение у задачи, полученной в результате данного изменения? Если да, найдите его, если нет, объясните, почему такого решения нет.

/продолжение на странице 19/

10. В данном вопросе шесть не связанных между собой пунктов (א)-(ו).

Ответьте на вопросы всех этих пунктов.

(א) В транспортной задаче [בעיית תובלה] три пункта отправления и три пункта назначения. Стоимость доставки единицы из каждого пункта отправления в каждый пункт назначения приведена в таблице.

Пункты отправления	Пункты назначения			Предло- жение
	1	2	3	
1	8	9	6	18
2	4	5	8	a
3	5	10	7	10
Спрос	12	15	15	

Определите, каким должно быть предложение a , чтобы получить таблицу стоимостей и спроса, без фиктивного пункта отправления и без фиктивного пункта назначения.

Обратите внимание: продолжение вопроса на следующей странице.

- (ב) В приведенной таблице дана часть возможного базового решения транспортной задачи: $x_{12} = 1$, $x_{11} = 9$.

Пункты отправления	Пункты назначения			Предло- жение
	1	2	3	
1	10 9	15 1	17	10
2	10	18	14	12
3	15	20	18	9
Спрос	9	12	10	

Перепишите эту таблицу в тетрадь и дополните в ней возможное базовое решение в соответствии с методом северо-западного угла.

Обратите внимание: продолжение вопроса на следующей странице.

- (ג) В приведенной ниже таблице дана часть возможного базового решения транспортной задачи и даны соответствующие этому решению значения $u_1, u_2, u_3, v_1, v_2, v_3$.

Пункты отправления	Пункты назначения			Предло- жение	u_i
	1	2	3		
1	2 20	5	7	20	2
2	0	8 10	4	10	0
3	10	0	8 10	15	-8
Спрос	20	15	10		
v_j	0	8	16		

Перепишите эту таблицу в тетрадь и дополните решение до конца, учитывая значения $u_1, u_2, u_3, v_1, v_2, v_3$, так чтобы получилось возможное базовое решение.

Обратите внимание: продолжение вопроса на следующей странице.

- (7) В приведенной ниже таблице дано возможное базовое решение транспортной задачи, а также дано значение u_1 .

Пункты отправления	Пункты назначения			Предло- жение	u_i
	1	2	3		
1	6	7	9	18	0
		15	3		
2	2	0	6	10	
	10	0			
3	7	12	10	10	
			10		
Спрос	10	15	13		
v_j					

- (1) Перепишите эту таблицу в тетрадь и дополните в ней значения u_2, u_3, v_1, v_2, v_3 .
- (2) Объясните, почему данное решение не является оптимальным решением.

Обратите внимание: продолжение вопроса на следующей странице.

(ה) В приведенной ниже таблице дано возможное решение транспортной задачи и даны соответствующие этому решению значения

$u_1, u_2, u_3, v_1, v_2, v_3$.

Пункты отправления	Пункты назначения			Предло- жение	u_i
	1	2	3		
1	10 20	25	30	20	10
2	10 30	22	14 20	50	10
3	16	20 40	20 20	60	16
Спрос	50	40	40		
v_j	0	4	4		

Выберите верное высказывание из приведенных ниже высказываний 1-4.

Перепишите его в тетрадь и обоснуйте свой выбор.

1. Данное решение не является возможным базовым решением.
2. Данное решение является возможным базовым, но не является оптимальным решением.
3. Данное решение является единственным оптимальным решением.
4. Данное решение является оптимальным решением, но не является единственным оптимальным решением.

Обратите внимание: продолжение вопроса на следующей странице.

(י) В приведенной ниже таблице дана транспортная задача.

Данные задачи, включая стоимость доставки единицы из каждого пункта отправления в каждый пункт назначения, приведены в таблице.

Пункты отправления	Пункты назначения			Предложение
	1	2	3	
1	22	30	32	20
2	30	18	24	20
3	15	10	8	30
4	25	15	22	30
Спрос	30	40	25	

Дано условие, согласно которому запрещено оставлять продукцию в пункте отправления 1.

Данная задача не является "уравновешенной" [למאזן], поэтому ученик предложил добавить фиктивный пункт назначения 4, который получит пять продуктов из излишнего предложения. Стоимость доставки единицы из каждого пункта отправления в каждый пункт назначения приведена в таблице. Значения стоимостей фиктивного пункта назначения 4 приведены в таблице как параметры x , y , z , w .

Пункты отправления	Пункты назначения				Предложение
	1	2	3	4	
1	22	30	32	x	20
2	30	18	24	y	20
3	15	10	8	z	30
4	25	15	22	w	30
Спрос	30	40	25	5	

Обратите внимание: продолжение вопроса на следующей странице.

/продолжение на странице 25/

Каким должно быть значение каждого из параметров x, y, z, w , чтобы можно было найти возможное базовое решение данной транспортной задачи?

Выберите правильный ответ из четырех приведенных ниже ответов, перепишите его в тетрадь и обоснуйте свой выбор.

(M обозначает очень большое число).

1. $w = 0$, $z = 0$, $y = 0$, $x = 0$
2. $w = M$, $z = M$, $y = M$, $x = M$
3. $w = 0$, $z = 0$, $y = 0$, $x = M$
4. $w = M$, $z = M$, $y = M$, $x = 0$

11. В вопросе два не связанных между собой пункта, (א) и (ב). Ответьте на вопросы обоих пунктов.

(א) $G = (V, E)$ – **ориентированный граф** [גרף מכוון], представленный матрицей смежностей [מטריצת סמיכויות]:

	a	b	c	d	e
a	0	0	0	0	1
b	0	0	0	1	0
c	1	0	0	0	0
d	0	0	0	0	1
e	0	0	1	0	0

- (1) Начертите граф G , представленный данной матрицей.
- (2) Найдите компонент/компоненты сильной связности [רכיב/רכיבי הקשירות החזקה] (Strong Connected Components – [רק"חים]) в данном графе.
Для каждого найденного компонента запишите множество его вершин.
- (3) Каково **минимальное число** ребер, которые необходимо добавить в данный граф, чтобы он содержал единственный компонент сильной связности? Какое ребро (или какие ребра) требуется добавить?
- (4) Сколько компонент сильной связности есть в транспонированном [מוחלף] графе $G^T = (V, E^T)$ данного графа G и каковы они?
Внимание: каждое ориентированное ребро (a, b) в графе G превращается в транспонированном графе G^T в ребро (b, a) .

Обратите внимание: пункт (ב) этого вопроса на следующей странице.

(ב) Нет связи с пунктом (א).

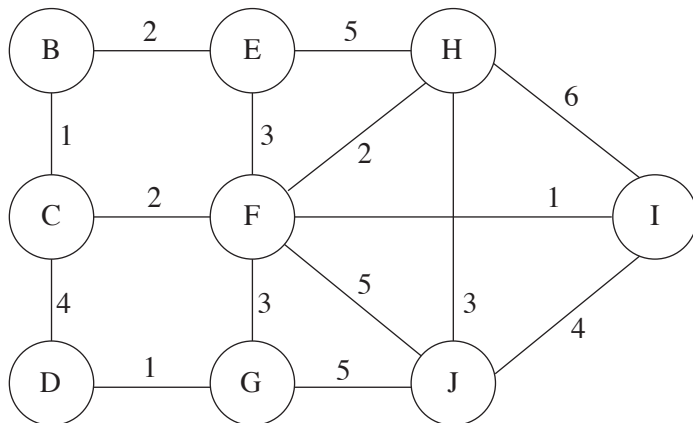
$G = (V, E)$ – **неориентированный граф** [גרף לא מכוון], представленный списком смежностей [סמיכויות]:

a	→	b	→	c	→			
b	→	a	→	c	→	d	→	
c	→	e	→	a	→	b	→	
d	→	b	→					
e	→	c	→					

- (1) Начертите граф G , представленный данным списком.
- (2) Сколько компонент связности (Connected Components) [רכיבי קשירות] есть в данном графе и каковы они?
- (3) Примените алгоритм сканирования в глубину (DFS) для данного графа G , начиная с вершины a .
Начертите в вашей тетради **только** полученное покрывающее дерево [העץ הפורש] (DFS) / покрывающий лес [היער הפורש] (DFS).
Основывайтесь на представлении, данном в списке смежностей.
- (4) Примените алгоритм сканирования в ширину (BFS) для данного графа G , начиная с вершины a .
Начертите в вашей тетради **только** полученное покрывающее дерево [העץ הפורש] (BFS) / покрывающий лес [היער הפורש] (BFS).
Основывайтесь на представлении, данном в списке смежностей.

12. В данном вопросе три не связанных между собой пункта (א)-(ג).
Ответьте на вопросы всех пунктов.

(א) Дана сеть $G = (V, E)$:



- (1) Найдите все самые короткие маршруты от вершины C до вершины J в данной сети. Схематически изобразите каждый из этих маршрутов.
- (2) Найдите минимальное покрывающее дерево [העץ המכסה] в соответствии с алгоритмом Крускала. Схематически изобразите это дерево.

(ב) Дан **связный неориентированный** граф [קשר לא מכוון] $G=(V, E)$, а также функция веса $w: E \rightarrow \mathbb{R}^+$.

Определим новую функцию веса $cl: E \rightarrow \mathbb{R}^+$ следующим образом: для каждого ребра e в E , $cl(e) = w(e) + a$, где a – некоторое число больше нуля.

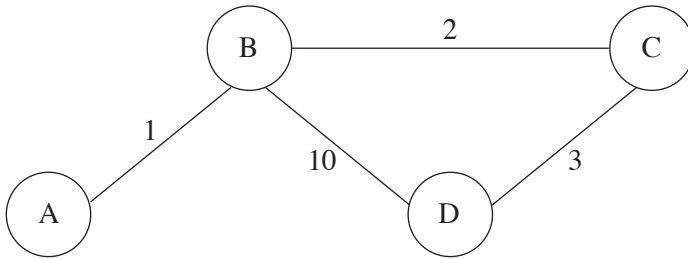
Ученик утверждал, что если T – это минимальное покрывающее дерево [עץ מינימלי] графа G в соответствии с алгоритмом Крускала при использовании функции веса w , то T – это минимальное покрывающее дерево графа G в соответствии с алгоритмом Крускала также и при использовании функции веса cl .

Прав ли этот ученик? Если да, обоснуйте почему. Если нет, приведите пример, который опровергнет слова этого ученика.

Обратите внимание: пункт (ג) этого вопроса на следующей странице.

/продолжение на странице 29/

- (ג) Дан связный неориентированный граф [קשיר לא מכוון] $G=(V, E)$,
а также функция веса $w:E \rightarrow \mathbb{R}^+$.



Для данного графа G , дерево самых коротких маршрутов от вершины A к остальным вершинам графа таково: $A \rightarrow B \rightarrow C \rightarrow D$.
Определим новую функцию веса $c_2:E \rightarrow \mathbb{R}^+$ следующим образом:
для каждого ребра e в E , $c_2(e) = w(e) + a$, где a – некоторое число больше нуля.

- (1) Приведите пример значения a , для которого дерево самых коротких маршрутов от вершины A к остальным вершинам графа не изменится.
- (2) Приведите пример значения a , для которого дерево самых коротких маршрутов от вершины A к остальным вершинам графа изменится.

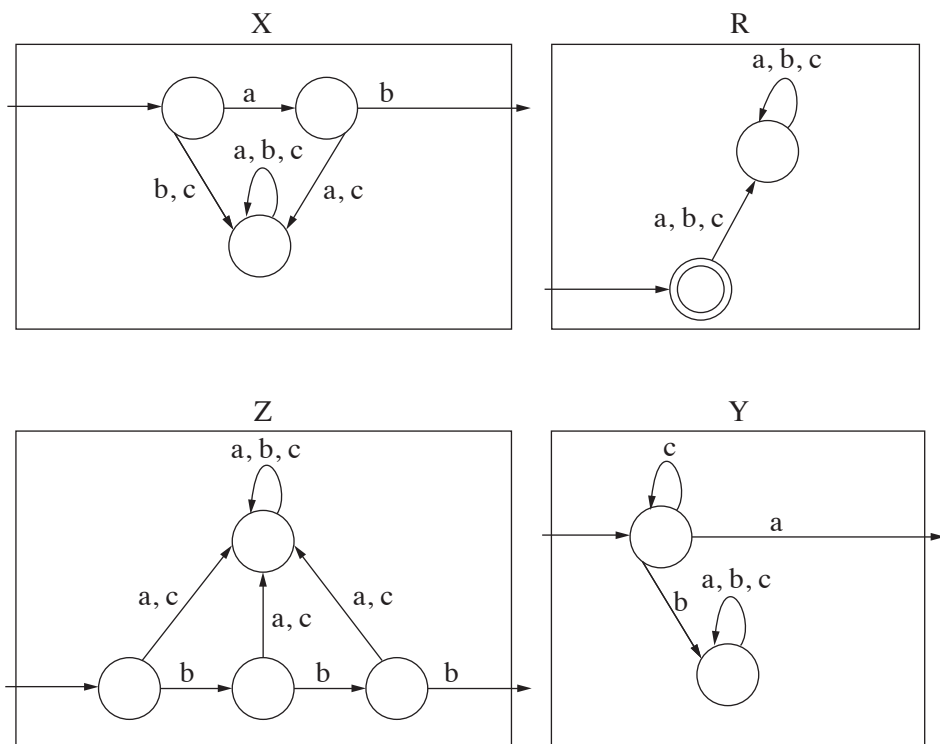
Вычислительные модели

[מודלים חישוביים]

Если вы изучали этот курс, ответьте на два из вопросов 13-16 (за каждый вопрос – 25 баллов).

13. На заводе "Автоматы ЛТД" производят четыре вида компонентов, приведенных ниже, для построения детерминированных конечных автоматов.

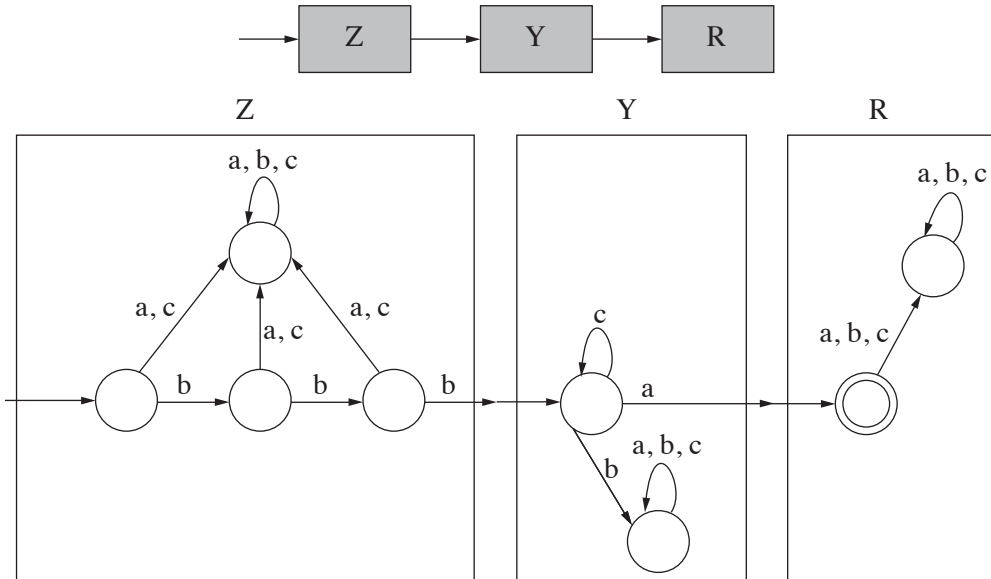
Каждый из компонентов не является автоматом, но из них можно собрать детерминированный конечный автомат, посредством присоединения друг к другу в определенном порядке, слева направо. Можно использовать больше одного компонента каждого вида.



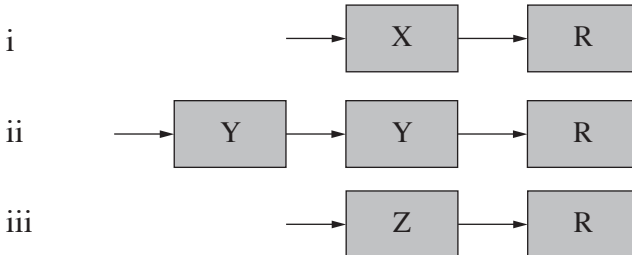
Обратите внимание: продолжение вопроса на следующей странице.

Например, чтобы построить автомат, принимающий [מקבל]

язык $\{b^3c^na \mid n \geq 0\}$, нужно соединить приведенные ниже компоненты в указанном порядке, слева направо:



(א) Запишите язык, который принимает каждый из приведенных ниже автоматов i-iii.



(ב) Даны языки L_1, L_2, L_3 :

$$L_1 = \{ab^4c^na \mid n \geq 0\}$$

$$L_2 = \{c^na^2b^7ab \mid n \geq 0\}$$

$$L_3 = \{b^6c^na^2b^4 \mid n \geq 0\}$$

Для каждого из языков L_1, L_2, L_3 запишите требуемые компоненты из списка компонентов X, Y, Z, R в порядке соединения, слева направо, чтобы собрать детерминированный конечный автомат, принимающий данный язык. Можно использовать больше одного компонента каждого вида.

/продолжение на странице 32/

14. В данном вопросе три не связанных между собой пункта (א)-(ג).

Ответьте на вопросы всех пунктов.

(א) Дан язык L над алфавитом $\{a, b\}$.

L – это набор слов, в каждом из которых модуль разности между количеством символов a и количеством символов b равен 1, и для каждой приставки* $[אישא]$ каждого слова, принадлежащего языку L , модуль данной разности не превышает 1.

* Приставка $[אישא]$ слова w – это любое слово, получаемое при удалении любого количества символов с конца слова w , включая пустое слово и само слово w .

Например: $w = abba$

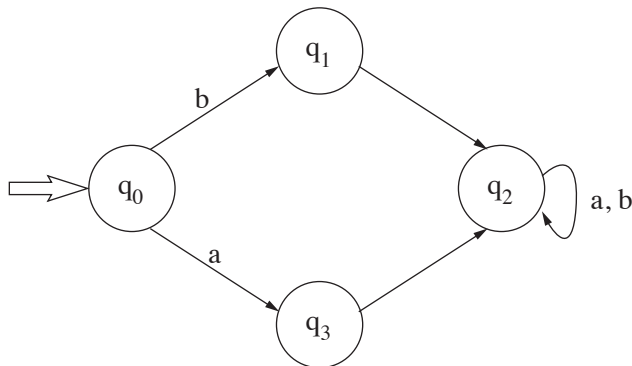
Все приставки слова w таковы: $\epsilon, a, ab, abb, abba$.

Пример слова, принадлежащего языку L : $ababa$

Пример слов, не принадлежащих языку L : $\epsilon, baab, aaabb$.

Перед вами частичное изображение детерминированного конечного автомата, принимающего $[מקבל]$ язык L .

В изображении недостает переходов $[מעברים]$, входных символов $[סימני קלט]$ и принимающих состояний $[מצבים מקבלים]$.



Чертеж содержит все состояния $[מצבים]$ автомата.

Перенесите в вашу тетрадь этот чертеж и дополните его так, чтобы автомат принимал язык L .

Вы должны добавить недостающие переходы и недостающие входные символы, а также отметить все принимающие состояния.

Внимание: не разрешается добавлять в автомат новые состояния или удалять существующие состояния.

Обратите внимание: продолжение вопроса на следующей странице.

/продолжение на странице 33/

(ב) Дано два регулярных языка [שפות רגולריות] L_1 , L_2 .

Определим $L_3 = \{w_1 w_2 \mid w_1 \in L_1, w_2 \in L_2, |w_1| = |w_2|\}$.

Приведите пример двух таких языков L_1 , L_2 , что

L_1 и L_2 – регулярные языки, а L_3 – **не** регулярный [לא רגולרית] язык.

Объясните свой ответ.

(ג) Перед вами регулярные языки L_1 , L_2 , L_3 над алфавитом $\{a, b, c\}$.

$$L_1 = \{a^n b^k \mid n, k \geq 0\}$$

$$L_2 = \{c^n \mid n \geq 0\}$$

$$L_3 = \{\varepsilon\}$$

Перед вами язык L .

$$L = \{c^n b^k a^j \mid n > 0, k, j \geq 0\}$$

Покажите, что язык L является регулярным только при помощи языков L_1 , L_2 , L_3 и свойств замкнутости [תכונות סגירות].

Объясните свой ответ.

15. Перед вами язык L .

$$L = \left\{ a^s b^{2s} a^{i_1} b^{j_1} a^{i_2} b^{j_2} \dots a^{i_n} b^{j_n} \mid \begin{array}{l} s \geq 1, n \geq 1 \\ \text{для любого } k, 1 \leq k \leq n \text{ выполняется } i_k \geq 1, j_k \geq 1 \end{array} \right\}$$

(א) Запишите самое короткое слово языка L .

(ב) Постройте стек-автомат (автомат с магазинной памятью)

[אוטומט מחסנית], принимающий [מקבל] язык L .

16. В данном вопросе два не связанных между собой пункта (א)-(ב).
Ответьте на вопросы обоих этих пунктов.

(א) Перед вами языки L_1 , L_2 над алфавитом $\{a, b, c\}$.

$$L_1 = \{w \in \{0, 1\}^* \mid \#_0(w) \leq \#_1(w) \leq 2 \times \#_0(w)\}$$

$$L_2 = \{0^s \mid s \geq 0\}$$

$\{0, 1\}^*$ – это набор всех слов над алфавитом $\{0, 1\}$, включая пустое слово,

$\#_0(w)$ обозначает число нулей в слове w ,

$\#_1(w)$ обозначает число единиц в слове w .

Дано два слова из L_2 : $w_1 = 0^i$ и $w_2 = 0^j$, так что $i < j$.

Найдите такое слово w , чтобы выполнялись условия:

$$w_1 \cdot w \in L_1$$

$$w_2 \cdot w \notin L_1$$

(ב) Нет связи с пунктом (א).

Перед вами язык L над алфавитом $\{0, 1\}$.

$$L = \{10^i(11)^j \mid i, j \geq 0\} \cup \{10^k 1 \mid k \geq 0\}$$

Перед вами частичное изображение **недетерминированного** [לא דטרמיניסטי] конечного автомата, принимающего [מקבל] язык L .

В изображении недостает переходов [מעברים], входных символов [סימני קלט] и принимающих состояний [מצבים מקבלים].



Чертеж содержит все состояния [מצבים] этого **недетерминированного** автомата.

Перенесите в вашу тетрадь этот чертеж и дополните его так, чтобы автомат принимал язык L .

Вы должны добавить недостающие переходы и недостающие входные символы и обозначить недостающие принимающие состояния.

Внимание: не разрешается добавлять в автомат новые состояния или удалять уже существующие состояния.

Объектно-ориентированное программирование [תכנות מונחה עצמים]

Если вы изучали этот курс и пишете на языке Java, ответьте на два вопроса из вопросов 17-20 (за каждый вопрос – 25 баллов).

17. В данном вопросе три не связанных между собой пункта (א)-(ג).

Ответьте на вопросы всех пунктов.

(א) Перед вами фрагмент кода главного метода класса Run:

```
C c = new A();
```

```
B b1 = (B) (new A());
```

```
B b2 = new D();
```

```
A a = new D();
```

(1) Считайте, что A , B , C , D – это классы, и известно, что у каждого из них есть не более одного наследующего класса. Напишите два варианта иерархии классов, при которых данный фрагмент кода будет корректным.

(2) Дано также, что метод f() определен только в классе B.

И дано, что в главной программе содержатся также следующие команды:

```
c.f();
```

```
b1.f();
```

Определите, для какого из вариантов иерархии классов, написанных вами в пункте (א)-1, данные команды будут корректными.

Скопируйте эту иерархию классов в тетрадь и объясните, почему для другой иерархии данные команды не являются корректными.

(ג) Перед вами фрагмент программы главного метода класса Program:

```
CC c = new AA();
```

```
BB b1 = (BB) (new AA(2));
```

Определите, какой из элементов AA, BB, CC не может быть интерфейсом [רשומ]. Объясните почему.

Обратите внимание: пункт (ג) данного вопроса на следующей странице.

- (ג) Перед вами классы A и B , а также класс Run с главным методом main. Запишите, каким будет вывод [פלט] главного метода.

```
public class A
{
    public A()
    {
        f();
    }
    public void f()
    {
        System.out.println("Class A");
    }
}
```

```
public class B extends A
{
    public B()
    {
        super();
    }
    public void f()
    {
        System.out.println("Class B");
    }
}
```

```
public Class Run
{
    public static void main(String[] args)
    {
        A a = new B();
    }
}
```

18. Перед вами классы AA, BB, CC, DD, а также класс Run с главным методом.

```
public class AA
{
    protected int i;
    public AA(int i)
    {
        this.i = i;
    }
}

public class BB extends AA
{
    public BB(int i)
    {
        super(i + 1);
    }
    public boolean what(Object other)
    {
        return ((other != null) &&
                (other instanceof BB) &&
                (this.i == ((BB)other).i));
    }
    public boolean what (BB other, int k)
    {
        return ((other != null) &&
                (this.i - k == ((BB)other).i));
    }
}

public class CC extends BB
{
    public CC(int i)
    {
        super(i);
    }
}
```

```
public class DD extends BB
{
    public DD(int i)
    {
        super(i + 1);
    }
    public boolean what(BB other, int k)
    {
        return super.what(other, 1);
    }
}

public class Run
{
    public static void main(String[] args)
    {
        AA a = new AA(1);
        BB b = new BB(1);
        CC c = new CC(1);
        DD d = new DD(1);
        BB b1 = new DD(1);
        AA c1 = new CC(1);
        /***/
    }
}
```

Обратите внимание: продолжение вопроса на следующей странице.

/продолжение на странице 38/

(א) Начертите иерархию классов AA, BB, CC, DD.

(ב) Проследите за выполнением главного метода.

Трассировка [מקור] должна содержать значения переменных, а также для каждого объекта класса [מזן] необходимо написать значения его атрибутов.

(ג) Перед вами команды i-iii.

i a = c;

ii b = a;

iii c = (CC) b1;

Для каждой из команд i-iii:

– Запишите эту команду вместо строки `/**` главного метода класса Run.

– Определите, является эта команда корректной или некорректной. Если команда корректна, покажите, какие изменения произошли с объектами классов в результате данной команды.

Если команда некорректна, обоснуйте ваше решение и определите вид ошибки: ошибка компиляции [הידור] или выполнения [ריצה].

(ד) Перед вами команды i-iv.

i System.out.println(b1.what(b));

ii System.out.println(b1.what(b, 1));

iii System.out.println(((CC) c1).what(c));

iv System.out.println(d.what(a));

Для каждой из команд i-iv:

– Запишите эту команду вместо строки `/**` главного метода класса Run.

– Запишите, каким будет вывод [פלט] в результате каждой из команд.

19. Перед вами интерфейс [ממשק] IThing , классы A , B , C , D и главный метод класса Run.

- (א) Проследите за выполнением главного метода и запишите полученный вывод [פלט]. Трассировка [מעקב] должна содержать значения переменных, а также для каждого объекта класса [פזו] значения его атрибутов.

```
public interface IThing
```

```
{  
    public int value();  
}
```

```
public class A implements IThing
```

```
{  
    private int a;  
    public A(int a)  
    {  
        this.a = a;  
    }  
    public int value()  
    {  
        return this.a;  
    }  
}
```

```
public class B implements IThing
```

```
{  
    protected IThing a , b;  
    public B(IThing a , IThing b)  
    {  
        this.a = a;  
        this.b = b;  
    }  
    public int value()  
    {  
        return this.a.value() + this.b.value();  
    }  
}
```

Обратите внимание: продолжение этого пункта на следующей странице.

/продолжение на странице 40/

```
public class C extends B
{
    public C(IThing a, IThing b)
    {
        super(a , b);
    }
    public int value()
    {
        return this.a.value() * this.b.value();
    }
}
```

```
public class D implements IThing
{
    private IThing[] things;
    private int limit;
    public D(IThing[] things , int limit)
    {
        this.things = things;
        this.limit = limit;
    }
    public int value()
    {
        int val = 0;
        for (int i = 0; i < limit; i++)
        {
            val += this.things[i].value();
        }
        return val;
    }
}
```

Обратите внимание: продолжение этого пункта на следующей странице.


```
public class Run
{
    public static void main(String[] args)
    {
        IThing[] things = new IThing[5];
        things[0] = new A(2);
        things[1] = new B(things[0] , things[0]);
        things[2] = new C(things[0] , things[1]);
        things[3] = things[2];
        things[4] = new D(things , 4);    //    (*)
        for(int i = 0; i < things.length; i++)
        {
            System.out.println(things[i].value());
        }
    }
}
```

- (ב) קратко объясните, как повлияет на выполнение программы замена в строке, обозначенной (*), существующей команды на команду, приведенную ниже:

```
things[4] = new D(things , 5);
```

20. Фирма "**Игрушки Это Они**" собирается компьютеризировать ассортимент игрушек на центральном складе.

Для этого были определены два класса, соответствующие двум типам игрушек:

класс **Doll** для кукол, и класс **Car** для машинок.

В классе **Doll** пять атрибутов [לכונות] и два метода.

Атрибуты: название куклы, базовая цена куклы, цвет волос, количество сопутствующих аксессуаров, цена аксессуара (у всех аксессуаров одинаковая цена).

Методы:

- (i) Возврат рыночной цены куклы. Рыночная цена – это базовая цена куклы, к которой добавлена цена всех сопутствующих аксессуаров (вычисляется как количество аксессуаров, умноженное на цену одного аксессуара).
- (ii) Обновление базовой цены куклы посредством ее увеличения на p процентов, где p – вещественное число [מספר ממשי], получаемое в качестве параметра.

В классе **Car** четыре атрибута и два метода.

Атрибуты: название машинки, базовая цена машинки, код размера машинки, цвет машинки.

Код размера машинки представлен целым числом: 0 – маленькая машинка, 1 – средняя машинка, 2 – большая машинка.

Методы:

- (i) Возврат рыночной цены машинки. Рыночная цена устанавливается на основании размера.
Рыночная цена маленькой машинки равна ее базовой цене.
Рыночная цена средней машинки равна ее базовой цене + 15 шек.
Рыночная цена большой машинки равна ее базовой цене + 30 шек.
- (ii) Обновление базовой цены машинки посредством ее увеличения на p процентов, где p – вещественное число [מספר ממשי], получаемое в качестве параметра.

Обратите внимание: продолжение вопроса на следующей странице.

Перед вами схема классов компьютеризированного ассортимента игрушек:

Doll
— String name — double basePrice — String color — int accNums — double accPrice
+ double price() + void updatePrice(double percent) + Doll(String name , double basePrice , String color , int accNum , double accPrice)

Car
— String name — double basePrice — String color — int size
+ double price() + void updatePrice(double percent) + Car (String name , double basePrice , String color , int size)

Обозначения:

- обозначает private
- + обозначает public

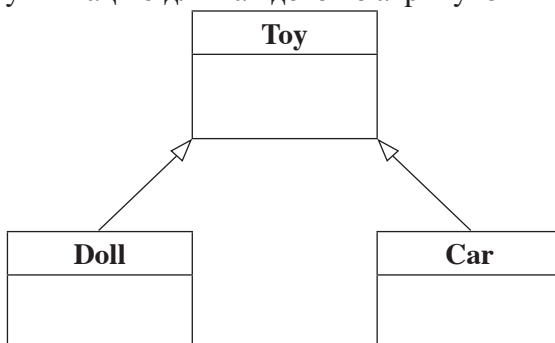
Обратите внимание: пункты это вопроса на следующей странице.

- (⌘) Определен новый класс **Toy**, наследниками которого являются классы **Doll** и **Car**.

Перед вами пустая схема классов для классов **Toy**, **Doll**, **Car**

Перенесите эту схему в вашу тетрадь и запишите в ней атрибуты и заголовки методов для каждого из классов в максимальном соответствии с принципами объектно-ориентированного программирования (инкапсуляция [הכמסה] – encapsulation, наследование [הורשה] – inheritance, полиморфизм – polymorphism).

Напишите документацию для каждого из атрибутов.



В пунктах (ב)-(ג) отвечайте по схеме, предложенной вами в пункте (⌘).

- (ב) Реализуйте на Java метод `price` в каждом из классов, где он присутствует.
- (ג) Напишите на Java главный метод, который выполнит две приведенных ниже задачи.
- (1) Создание одного объекта [פזו] типа **Doll** и одного объекта [פזו] типа **Car** (со значениями по вашему выбору).
 - (2) Вычисление и вывод на экран суммы рыночных цен двух этих игрушек (куклы и автомобиля).

Объектно-ориентированное программирование
[תכנות מונחה עצמים]

Если вы изучали этот курс и пишете на языке C#, ответьте на два вопроса из вопросов 21-24 (за каждый вопрос – 25 баллов).

21. В данном вопросе три не связанных между собой пункта (א)-(ג).

Ответьте на вопросы всех пунктов.

(א) Перед вами фрагмент кода главного метода класса Run:

```
C c = new A();
```

```
B b1 = (B) (new A());
```

```
B b2 = new D();
```

```
A a = new D();
```

(1) Считайте, что A , B , C , D – это классы, и известно, что у каждого из них есть не более одного наследующего класса.

Напишите два варианта иерархии классов, при которых данный фрагмент кода будет корректным.

(2) Дано также, что метод f() определен только в классе B.

И дано, что в главной программе содержатся также следующие команды:

```
c.f();
```

```
b1.f();
```

Определите, для какого из вариантов иерархии классов, написанных вами в пункте (א)-1, данные команды будут корректными.

Скопируйте эту иерархию классов в тетрадь и объясните, почему для другой иерархии данные команды не являются корректными.

(ג) Перед вами фрагмент программы главного метода класса Program:

```
CC c = new AA();
```

```
BB b1 = (BB) (new AA(2));
```

Определите, какой из элементов AA, BB, CC не может быть интерфейсом [רשומ]. Объясните почему.

Обратите внимание: пункт (ג) данного вопроса на следующей странице.

- (ג) Перед вами классы A и B , а также класс Run с главным методом main. Запишите, каким будет вывод [פלט] главного метода.

```
public class A
{
    public A()
    {
        F();
    }
    public virtual void F()
    {
        Console.WriteLine("Class A");
    }
}
```

```
public class B : A
{
    public B(): base() { }
    public override void F()
    {
        Console.WriteLine("Class B");
    }
}
```

```
public Class Run
{
    public static void Main()
    {
        A a = new B();
    }
}
```

22. Перед вами классы AA , BB , CC , DD , а также класс Run с главным методом.

<pre>public class AA { protected int i; public AA(int i) { this.i = i; } } public class BB : AA { public BB(int i) : base(i + 1) { } public virtual bool What(Object other) { return ((other != null) && (other is BB) && (this.i == ((BB)other).i)); } public virtual bool What (BB other, int k) { return ((other != null) && (this.i - k == ((BB)other).i)); } } public class CC : BB { public CC(int i) : base (i) { }</pre>	<pre>public class DD : BB { public DD(int i) : base(i + 1){ } public override bool What (BB other , int k) { return base.What(other , 1); } } public class Run { public static void Main() { AA a = new AA(1); BB b = new BB(1); CC c = new CC(1); DD d = new DD(1); BB b1 = new DD(1); AA c1 = new CC(1); /*** } }</pre>
--	--

Обратите внимание: продолжение вопроса на следующей странице.

(ח) Начертите иерархию классов AA, BB, CC, DD.

(ט) Проследите за выполнением главного метода.

Трассировка [טרעק] должна содержать значения переменных, а также для каждого объекта класса [אובייקט] необходимо написать значения его атрибутов.

(א) Перед вами команды i-iii.

i a = c;

ii b = a;

iii c = (CC) b1;

Для каждой из команд i-iii:

– Запишите эту команду вместо строки `/**` главного метода класса Run.

– Определите, является эта команда корректной или некорректной. Если команда корректна, покажите, какие изменения произошли с объектами классов в результате данной команды.

Если команда некорректна, обоснуйте ваше решение и определите вид ошибки: ошибка компиляции [הידור] или выполнения [ריצה].

(ב) Перед вами команды i-iv.

i Console.WriteLine(b1.What(b));

ii Console.WriteLine(b1.What(b, 1));

iii Console.WriteLine(((CC) c1).What(c));

iv Console.WriteLine(d.What(a));

Для каждой из команд i-iv:

– Запишите эту команду вместо строки `/**` главного метода класса Run.

– Запишите, каким будет вывод [פלט] в результате каждой из команд.

23. Перед вами интерфейс `IThing`, классы `A`, `B`, `C`, `D` и главный метод класса `Run`.

(*) Проследите за выполнением главного метода и запишите полученный вывод [פלט].

Трассировка [מקרב] должна содержать значения переменных, а также для каждого объекта класса [פזז] значения его атрибутов.

```
public interface IThing
```

```
{  
    int Value();  
}
```

```
public class A : IThing
```

```
{  
    private int a;  
    public A(int a)  
    {  
        this.a = a;  
    }  
    public int Value()  
    {  
        return this.a;  
    }  
}
```

```
public class B : IThing
```

```
{  
    protected IThing a , b;  
    public B(IThing a , IThing b)  
    {  
        this.a = a;  
        this.b = b;  
    }  
    public virtual int Value()  
    {  
        return this.a.Value() + this.b.Value();  
    }  
}
```

Обратите внимание: продолжение этого пункта на следующей странице.

/продолжение на странице 50/

```
public class C : B
{
    public C(IThing a, IThing b) : base(a , b) { }
    public override int Value()
    {
        return this.a.Value() * this.b.Value();
    }
}
```

```
public class D : IThing
{
    private IThing[] things;
    private int limit;
    public D (IThing[] things , int limit)
    {
        this.things = things;
        this.limit = limit;
    }
    public int Value()
    {
        int val = 0;
        for (int i = 0; i < limit; i++)
        {
            val += this.things[i].Value();
        }
        return val;
    }
}
```

Обратите внимание: продолжение этого пункта на следующей странице.

```
public class Run
{
    public static void Main()
    {
        IThing[] things = new IThing[5];
        things[0] = new A(2);
        things[1] = new B(things[0] , things[0]);
        things[2] = new C(things[0] , things[1]);
        things[3] = things[2];
        things[4] = new D(things , 4);    //  (*)
        for(int i = 0; i < things.Length; i++)
        {
            Console.WriteLine(things[i].Value());
        }
    }
}
```

- (ב) קратко объясните, как повлияет на выполнение программы замена в строке, обозначенной (*), существующей команды на команду, приведенную ниже:

```
things[4] = new D(things , 5);
```

24. Фирма "**Игрушки Это Они**" собирается компьютеризировать ассортимент игрушек на центральном складе.

Для этого были определены два класса, соответствующие двум типам игрушек:

класс **Doll** для кукол, и класс **Car** для машинок.

В классе **Doll** пять атрибутов [לכונות] и два метода.

Атрибуты: название куклы, базовая цена куклы, цвет волос, количество сопутствующих аксессуаров, цена аксессуара (у всех аксессуаров одинаковая цена).

Методы:

- (i) Возврат рыночной цены куклы. Рыночная цена – это базовая цена куклы, к которой добавлена цена всех сопутствующих аксессуаров (вычисляется как количество аксессуаров, умноженное на цену одного аксессуара).
- (ii) Обновление базовой цены куклы посредством ее увеличения на p процентов, где p – вещественное число [מספר ממשי], получаемое в качестве параметра.

В классе **Car** четыре атрибута и два метода.

Атрибуты: название машинки, базовая цена машинки, код размера машинки, цвет машинки.

Код размера машинки представлен целым числом: 0 – маленькая машинка, 1 – средняя машинка, 2 – большая машинка.

Методы:

- (i) Возврат рыночной цены машинки. Рыночная цена устанавливается на основании размера.
Рыночная цена маленькой машинки равна ее базовой цене.
Рыночная цена средней машинки равна ее базовой цене + 15 шек.
Рыночная цена большой машинки равна ее базовой цене + 30 шек.
- (ii) Обновление базовой цены машинки посредством ее увеличения на p процентов, где p – вещественное число [מספר ממשי], получаемое в качестве параметра.

Обратите внимание: продолжение вопроса на следующей странице.

Перед вами схема классов компьютеризированного ассортимента игрушек:

Doll
— string name — double basePrice — string color — int accNums — double accPrice
+ double Price() + void UpdatePrice(double percent) + Doll(string name , double basePrice , string color , int accNum , double accPrice)

Car
— string name — double basePrice — string color — int size
+ double Price() + void UpdatePrice(double percent) + Car (string name , double basePrice , string color , int size)

Обозначения:

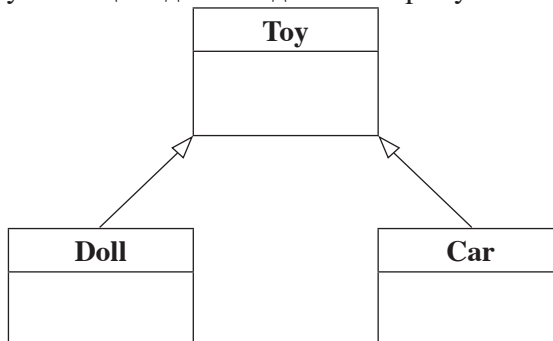
- обозначает private
- + обозначает public

Обратите внимание: пункты это вопроса на следующей странице.

- (א) Определен новый класс **Toy**, наследниками которого являются классы **Doll** и **Car**.

Перед вами пустая схема классов для классов **Toy**, **Doll**, **Car**

Перенесите эту схему в вашу тетрадь и запишите в ней атрибуты и заголовки методов для каждого из классов в максимальном соответствии с принципами объектно-ориентированного программирования (инкапсуляция [הכמסה] – encapsulation, наследование [הורשה] – inheritance, полиморфизм – polymorphism).
Напишите документацию для каждого из атрибутов.



В пунктах (ב)-(ג) отвечайте по схеме, предложенной вами в пункте (א).

- (ב) Реализуйте на C# метод `price` в каждом из классов, где он присутствует.
- (ג) Напишите на C# главный метод, который выполнит две приведенных ниже задачи.
- (1) Создание одного объекта [פזז] типа **Doll** и одного объекта [פזז] типа **Car** (со значениями по вашему выбору).
 - (2) Вычисление и вывод на экран суммы рыночных цен двух этих игрушек (куклы и автомобиля).

Желаем успеха!

Авторские права принадлежат Государству Израиль.
Копировать или публиковать можно только
с разрешения Министерства просвещения.

בהצלחה!

זכות היוצרים שמורה למדינת ישראל.
אין להעתיק או לפרסם
אלא ברשות משרד החינוך.