Государство Израиль
Министерство просвещения
Тип экзамена:
на аттестат зрелости для средних школ
Время проведения экзамена: лето 2016 года, **срок "бет"**
Номер вопросника: 899381
Перевод на русский язык (5)

מדינת ישראל
משרד החינוך
סוג הבחינה:
בגרות לבתי ספר על־יסודיים
מועד הבחינה: קיץ תשע"ו, 2016, **מועד ב**
מספר השאלון: 899381
תרגום לרוסית (5)

Компьютеры
в соответствии с программой реформы "значимое обучение"

Указания экзаменующимся

а. Продолжительность экзамена: три часа.

б. Строение вопросника и ключ к оценке:
в этом вопроснике три раздела.
Раздел первый: в этом разделе три вопроса, отвечайте согласно указаниям этого раздела
$(1 \times 10) + (1 \times 15)$ – 25 баллов
Раздел второй: в этом разделе три вопроса, вы должны ответить на два из них (2×25) – 50 баллов
Раздел третий: в этом разделе вопросы из четырех различных курсов. Вы должны ответить на один вопрос курса, который вы изучали.
(1×25) – 25 баллов
Всего – 100 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", "Введение в исследование операций", "Вычислительные модели", "Объектно-ориентированное программирование".

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

מ ד ע י ה מ ח ש ב
על פי תכנית הרפורמה ללמידה משמעותית

הוראות לנבחן

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח ההערכה:
בשאלון זה שלושה פרקים.
פרק ראשון — בפרק זה שלוש שאלות, ענה על פי ההוראות בפרק.
$(1 \times 10) + (1 \times 15) = 25$ נקודות —
פרק שני — בפרק זה שלוש שאלות, ומהן עליך לענות על שתיים.
$(2 \times 25) = 50$ נקודות —
פרק שלישי — בפרק זה שאלות בארבעה מסלולים שונים. ענה על שאלה אחת במסלול שלמדת. $(1 \times 25) = 25$ נקודות —
סה"כ — 100 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 ה**מסלול 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.

כתוב במחברת הבחינה בלבד, בעמודים נפרדים, כל מה שברצונך לכתוב בטיוטה (ראשי פרקים, חישובים וכדומה). רשום "טיוטה" בראש כל עמוד טיוטה.
רישום טיוטות 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! **ב ה צ ל ח ה !**

В о п р о с ы

В данном вопроснике три раздела.
Вы должны ответить на вопросы трех этих разделов, согласно инструкциям каждого раздела.

РАЗДЕЛ ПЕРВЫЙ (25 баллов)

Указание: в каждом вопросе, в котором требуется ввод, нет необходимости проверять корректность ввода.

Для решающих на языке Java: в каждом вопросе, в котором требуется ввод, предположите, что в программе написана команда:

```
Scanner input = new Scanner (System.in);
```

Ответьте на вопрос 1 – обязательный (10 баллов).

1. Дан главный метод [פעולה ראשית] класса **Stam**. В данном методе определен одномерный массив [מערך חד־ממדי] arr целого типа [טיפוס שלם].
   Напишите в данном главном методе фрагмент [קטע] программы на языке Java или C#, который примет [יקלוט] целое число num и выведет на экран [ידפיס] сумму всех элементов массива, значение которых меньше, чем num.

/продолжение на странице 3/

Ответьте на один из вопросов 2-3 (15 баллов).

2. Определен класс **A** с двумя атрибутами [תכונות]:
n1 – целого типа, и n2 – целого типа.
В данном классе определены: два конструктора [פעולות בונות]
и для каждого атрибута – методы get и set на Java и Get и Set на C# .
В данном классе метод [פעולה] toString на языке Java и ToString на языке C# возвращает строку [מחרוזת], в которой записан пример сложения значений n1 и n2 и полученный результат.
Например, если значение, содержащееся в n1 – число 3 , а значение, содержащееся в n2 – число 9 , данный метод вернет строку: "3 + 9 = 12".
Кроме того, метод void add(int k) на Java и void Add(int k) на C# прибавляет значение k к значению каждого из атрибутов n1 и n2 .
Вывод на экран объекта класса [הדפסת עצם] вызывает на языке Java метод toString соответствующего класса и на языке C# метод ToString соответствующего класса.
Перед вами главный метод [פעולה ראשית] класса Program, написанный на языках Java и C#. Данный метод использует класс **A**.

Java

```
public class Program
{
   public static void main(String[] args)
   {
      A a = new A();
      a.setN1(4);
      a.setN2(5);
      A a1 = new A(a.getN1(), a.getN2());
      System.out.println (a);
      a1.add(4);
      System.out.println(a);
      System.out.println(a1);
   }
}
```

C#

```
public class Program
{
   public static void Main()
   {
      A a = new A();
      a.SetN1(4);
      a.SetN2(5);
      A a1 = new A(a.GetN1(), a.GetN2());
      Console.WriteLine(a);
      a1.Add(4);
      Console.WriteLine(a);
      Console.WriteLine(a1);
   }
}
```

Обратите внимание: продолжение вопроса на следующей странице.

/продолжение на странице 4/

(א) Определите класс **A** в соответствии с пунктами i-v так, чтобы главный метод не вызывал ошибок при компиляции и/или выполнении [ריצה].

i Напишите на языке Java или C# заголовок [כותרת] класса и его атрибуты.

ii Напишите на языке Java или C# заголовки двух конструкторов класса **A**.

iii Реализуйте [ממש] на языке Java все методы get и set или на языке C# все методы Get и Set, необходимые для выполнения главного метода.

iv Реализуйте на языке Java метод add или на языке C# метод Add.

v Реализуйте на языке Java метод toString или на языке C# метод ToString .

(ב) Покажите объекты [עצם] класса, которые были созданы в процессе выполнения данного главного метода. Для каждого объекта класса запишите названия всех его атрибутов и значения этих атрибутов.

Кроме того, покажите, как изменяются объекты класса в результате выполнения команды add(4) на языке Java или Add(4) на языке C#.

Запишите вывод [פלט] главного метода.

/продолжение на странице 5/

3. Перед вами интерфейс [ממשק] класса **Button**, представляющего кнопку. У кнопки три атрибута [תכונות]: написанное на ней число num, размер кнопки size и цвет кнопки color.

Заголовок метода на языке Java	**Описание метода**
public Button(int num)	Конструктор кнопки с числом num, размером 5 и цветом "black"
public Button(int num , int size , String color)	Конструктор кнопки с числом num, размером size и цветом color
public int getSize()	Метод, возвращающий размер кнопки
public void addToSize(int x)	Метод, увеличивающий размер кнопки на x
public boolean isSameSize(Button other)	Метод, возвращающий true, если кнопка other имеет такой же размер, как данная кнопка, иначе [אחרת] – метод возвращает false

Заголовок метода на языке C#	**Описание метода**
public Button(int num)	Конструктор кнопки с числом num, размером 5 и цветом "black"
public Button(int num , int size , string color)	Конструктор кнопки с числом num, размером size и цветом color
public int GetSize()	Метод, возвращающий размер кнопки
public void AddToSize(int x)	Метод, увеличивающий размер кнопки на x
public bool IsSameSize(Button other)	Метод, возвращающий true, если кнопка other имеет такой же размер, как данная кнопка, иначе [אחרת] – метод возвращает false

(א) Реализуйте [ממֵש] в классе **Button** на языке Java или C# конструктор Button(int num) .

(ב) Реализуйте в классе **Button** на языке Java или C# метод, увеличивающий размер кнопки на x .

(ג) Реализуйте в классе **Button** на языке Java или C# метод, проверяющий, равен ли размер кнопки other размеру данной кнопки.

Обратите внимание: пункт (ד) на следующей странице.

/продолжение на странице 6/

(ד) Перед вами фрагмент [קטע] главного метода [פעולה ראשית] класса Program, написанный на языках Java и C# .
Проследите за выполнением данного фрагмента и запишите полученный вывод [פלט]. Для каждого экземпляра класса [עצם] запишите названия всех его атрибутов [תכונות], значения всех атрибутов и изменения, которые с ними произошли.

Java

```
Button a1 = new Button(1, 12, "red");
Button a2 = new Button(2, 12, "green");
Button a3 = new Button(3, 12, "blue");
Button a4 = new Button(4);
a1.addToSize(2);
a4.addToSize(9);
if (a1.isSameSize(a3))
        System.out.println("***");
if (a4.isSameSize(a1))
        System.out.println("$$$");
if (a2.isSameSize(a3))
        System.out.println("###");
```

C#

```
Button a1 = new Button(1, 12, "red");
Button a2 = new Button(2, 12, "green");
Button a3 = new Button(3, 12, "blue");
Button a4 = new Button(4);
a1.AddToSize(2);
a4.AddToSize(9);
if (a1.IsSameSize(a3))
        Console.WriteLine("***");
if (a4.IsSameSize(a1))
        Console.WriteLine("$$$");
if (a2.IsSameSize(a3))
        Console.WriteLine("###");
```

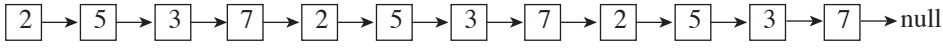
/продолжение на странице 7/

РАЗДЕЛ ВТОРОЙ (50 баллов)

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

Ответьте на <u>два</u> из вопросов 4-6 (за каждый вопрос – 25 баллов)

4. Список [רשימה] L называется **тройным**, если он удовлетворяет следующим условиям:
   * список не пустой;
   * число элементов списка делится на 3 без остатка;
   * элементы в первой трети списка содержат те же значения, что и элементы во второй трети списка, и те же значения, которые содержат элементы в третьей трети списка. Причем, порядок элементов в каждой из третей списка одинаков.
     Например, список L1 , представленный на схеме, является **тройным** списком из 12 элементов.



Напишите на Java или C# внешний метод [פעולה חיצונית], который получит список L с элементами целого типа.

Если L является **тройным** списком, метод вернет true .

Иначе [אחרת] – метод вернет false .

/продолжение на странице 8/

5. **ВНИМАНИЕ:** данный вопрос представлен в двух версиях:
одна – на Java (стр. 8-9), а другая – на C# (стр. 10-11).
Отвечайте в соответствии с языком, который вы изучали.

Для решающих на Java:

Перед вами методы sod и what , получающие массив [מערך] a
с элементами целого типа [טיפוס שלם], отсортированный по возрастанию,
а также целое число k .
У обоих методов одинаковое условие выхода [טענת יציאה].

```
public static boolean sod(int[] a , int k)
{
    for (int i = 0; i < a.length−1; i++)
    {
        int j = i+1;
        while (j < a.length)
        {
            if (a[i] + a[j] == k)
                return true;
            j++;
        }
    }
    return false;
}

public static boolean what(int[] a , int k)
{
    int left = 0 , right = a.length−1;
    while (left < right)
    {
        if (a[left] + a[right] == k)
            return true;
        if (a[left] + a[right] < k)
            left++;
        else
            right−−;
    }
    return false;
}
```

/продолжение на странице 9/

Дан массив a :

a	2	4	7	12	18

(א) Проследите с помощью таблицы трассировки [טבלת מעקב] за выполнением метода sod для данного массива a и числа k = 11 . Запишите возвращаемое значение.
Таблица трассировки должна содержать столбцы для i , j , a[i] , a[j] и дополнительный столбец, в котором будет указано, выполняется или не выполняется условие команды if .

(ב) Проследите с помощью таблицы трассировки за выполнением метода sod для данного массива a и числа k = 10 . Запишите возвращаемое значение. Таблица трассировки должна содержать столбцы, перечисленные в пункте (א).

(ג) Каково условие выхода [טענת יציאה] метода sod ?

(ד) Какова временная сложность [סיבוכיות זמן הריצה] выполнения метода sod ? Обоснуйте ваш ответ.

(ה) Проследите с помощью таблицы трассировки за выполнением метода what для данного массива a и числа k = 11 . Запишите возвращаемое значение.
Таблица трассировки должна содержать столбцы для left , right , a[left] , a[right] и два дополнительных столбца для каждой из команд if . В каждом столбце будет указано, выполняется или не выполняется условие команды if .

(ו) Какова временная сложность [סיבוכיות זמן הריצה] выполнения метода what ? Обоснуйте ваш ответ.

(ז) Какой их двух методов – sod или what – более эффективный? Обоснуйте ваш ответ.

(ח) Условие входа [טענת הכניסה] методов sod и what было изменено так, что им можно передавать неотсортированный массив a .

(1) Изменится ли условие выхода метода sod ? Обоснуйте ваш ответ.

(2) Изменится ли условие выхода метода what ? Обоснуйте ваш ответ.

/продолжение на странице 10/

Для решающих на C#:

Перед вами методы Sod и What , получающие массив [מערך] a с элементами целого типа [טיפוס שלם], отсортированный по возрастанию, а также целое число k .

У обоих методов одинаковое условие выхода [טענת יציאה].

```
public static bool Sod(int[] a , int k)
{
    for (int i = 0; i < a.Length−1; i++)
    {
        int j = i+1;
        while (j < a.Length)
        {
            if (a[i] + a[j] == k)
                return true;
            j++;
        }
    }
    return false;
}

public static bool What(int[] a , int k)
{
    int left = 0 , right = a.Length−1;
    while (left < right)
    {
        if (a[left] + a[right] == k)
            return true;
        if (a[left] + a[right] < k)
            left++;
        else
            right−−;
    }
    return false;
}
```

/продолжение на странице 11/

Дан массив a :

a	2	4	7	12	18

(א) Проследите с помощью таблицы трассировки [טבלת מעקב] за выполнением метода Sod для данного массива a и числа k = 11 . Запишите возвращаемое значение.
Таблица трассировки должна содержать столбцы для: i , j , a[i] , a[j] и дополнительный столбец, в котором будет указано выполняется или не выполняется условие команды if .

(ב) Проследите с помощью таблицы трассировки за выполнением метода Sod для данного массива a и числа k = 10 . Запишите возвращаемое значение. Таблица трассировки должна содержать столбцы, перечисленные в пункте (א).

(ג) Каково условие выхода [טענת יציאה] метода Sod ?

(ד) Какова временная сложность [סיבוכיות זמן הריצה] выполнения метода Sod? Обоснуйте ваш ответ.

(ה) Проследите с помощью таблицы трассировки за выполнением метода What для данного массива a и числа k = 11 . Запишите возвращаемое значение.
Таблица трассировки должна содержать столбцы для: left , right, a[left] , a[right] и два дополнительных столбца для каждой из команд if. В каждом столбце будет указано, выполняется или не выполняется условие команды if .

(ו) Какова временная сложность [סיבוכיות זמן הריצה] выполнения метода What ? Обоснуйте ваш ответ.

(ז) Какой из двух методов – Sod или What – более эффективный? Обоснуйте ваш ответ.

(ח) Условие входа [טענת הכניסה] методов Sod и What было изменено так, что им можно передавать неотсортированный массив a .
(1) Изменится ли условие выхода метода Sod ? Обоснуйте ваш ответ.
(2) Изменится ли условие выхода метода What ? Обоснуйте ваш ответ

/продолжение на странице 12/

6. **ВНИМАНИЕ**: данный вопрос представлен в двух версиях:
одна – на Java (стр. 12-13), а другая – на C# (стр. 14-15).
Отвечайте в соответствии с языком, который вы изучали.

Для решающих на Java:

Игра “Сыграй-ка” – это игра для одного игрока, состоящая из двух этапов. Игрок получает 52 перемешанные карты, отличные друг от друга. На каждой из карт есть число от 1 до 13 и изображение одной из четырех фигур. Фигуры представлены числами от 1 до 4 .

На первом этапе игрок раскладывает карты по 4 стопкам, в соответствии с фигурой, изображеннной на карте, чтобы в каждой стопке оказались карты с одинаковой фигурой. Карты складываются одна на другую.

На втором этапе игрок случайным образом выбирает число от 1 до 4, обозначающее фигуру. Игрок поднимает верхнюю карту со стопки, фигура которой соответствует выбранному числу, и переносит карту в пятую стопку.

Этот этап продолжается до завершения игры.

Игра завершается, когда выбрано число, соответствующее стопке, в которой больше нет карт.

Выигрыш – это ситуация, при которой сумма чисел на картах в пятой стопке делится на 100 без остатка.

После того, как игра окончена, проверяют, есть ли выигрыш.

Для реализации данной игры заданы классы: **Card**, представляющий одну карту, **Deck**, представляющий пять необходимых для игры стопок и **Test**, управляющий игрой.

Также написан метод public static boolean game(Card[] cards) в классе **Test**, который получает массив из 52 разных карт, упорядоченных случайным образом. Метод проводит игру до ее завершения. Метод возвращает “true”, если игрок выиграл в игре, в противном случае метод возвращает “false”.

/продолжение на странице 13/

Перед вами класс **Card**, представляющий карту с числом (value) от 1 до 13 и фигурой (shape) от 1 до 4.

```
public class Card
{
    private int value;
    private int shape;

    public Card (int value, int shape)
    {
        this.value = value;
        this.shape = shape;
    }
    public int getValue () {  return this.value;  }
    public int getShape () {  return this.shape;  }
}
```

(א) i Напишите на Java заголовок [כותרת] класса **Deck** и его атрибуты [תכונות].

ii Напишите на Java в классе **Deck** заголовки следующих методов:

– Конструктор, определяющий все 5 стопок пустыми.

– Метод insert , получающий карту и добавляющий ее в соответствующую стопку на первом этапе игры.

– Метод move, случайным образом выбирающий число, определяющее фигуру. Если соответствующая выбранной фигуре стопка карт пуста, метод вернет “false”. В противном случае метод перенесет в пятую стопку верхнюю карту из стопки, соответствующей выбранной фигуре, и вернет “true”.

– Метод sum, возвращающий сумму чисел на картах, находящихся в пятой стопке в момент вызова данного метода.

Внимание: в этом подпункте нет необходимости реализовывать эти методы

(ב) Реализуйте на Java конструктор класса **Deck**.

(ג) Реализуйте на Java метод sum класса **Deck**.

(ד) Реализуйте на Java метод game класса **Test**.

Вы можете использовать методы insert и move класса Deck не реализуя их. Если вы пользуетесь дополнительным методами, вы должны реализовать их.

/продолжение на странице 14/

Для решающих на C#:

Игра “Сыграй-ка” – это игра для одного игрока, состоящая из двух этапов. Игрок получает 52 перемешанные карты, отличные друг от друга. На каждой из карт есть число от 1 до 13 и изображение одной из четырех фигур. Фигуры представлены числами от 1 до 4 .

На первом этапе игрок раскладывает карты по 4 стопкам, в соответствии с фигурой, изображеннной на карте, чтобы в каждой стопке оказались карты с одинаковой фигурой. Карты складываются одна на другую.

На втором этапе игрок случайным образом выбирает число от 1 до 4, обозначающее фигуру. Игрок поднимает верхнюю карту со стопки, фигура которой соответствует выбранному числу, и переносит карту в пятую стопку.

Этот этап продолжается до завершения игры.

Игра завершается, когда выбрано число, соответствующее стопке, в которой больше нет карт.

Выигрыш – это ситуация, при которой сумма чисел на картах в пятой стопке делится на 100 без остатка.

После того, как игра окончена, проверяют, есть ли выигрыш.

Для реализации данной игры заданы классы: **Card**, представляющий одну карту, **Deck**, представляющий пять необходимых для игры стопок и **Test**, управляющий игрой.

Также написан метод public static boolean game(Card[] cards) в классе **Test**, который получает массив из 52 разных карт, упорядоченных случайным образом. Метод проводит игру до ее завершения. Метод возвращает “true”, если игрок выиграл в игре, в противном случае метод возвращает “false”.

/продолжение на странице 15/

Перед вами класс **Card**, представляющий карту с числом (value) от 1 до 13 и фигурой (shape) от 1 до 4.

```
public class Card
{
    private int value;
    private int shape;

    public Card (int value, int shape)
    {
        this.value = value;
        this.shape = shape;
    }
    public int GetValue () {   return this.value;    }
    public int GetShape () {   return this.shape;    }
}
```

(א) i Напишите на C# заголовок [כותרת] класса **Deck** и его атрибуты [תכונות].

ii Напишите на C# в классе **Deck** заголовки следующих методов:

– Конструктор, определяющий все 5 стопок пустыми.

– Метод Insert , получающий карту и добавляющий ее в соответствующую стопку на первом этапе игры.

– Метод Move, случайным образом выбирающий число, определяющее фигуру. Если соответствующая выбранной фигуре стопка карт пуста, метод вернет “false”. В противном случае метод перенесет в пятую стопку верхнюю карту из стопки, соответствующей выбранной фигуре, и вернет “true”.

– Метод Sum, возвращающий сумму чисел на картах, находящихся в пятой стопке в момент вызова данного метода.

Внимание: в этом подпункте нет необходимости реализовывать эти методы

(ב) Реализуйте на C# конструктор класса **Deck**.

(ג) Реализуйте на C# метод Sum класса **Deck**.

(ד) Реализуйте на C# метод Game класса **Test**.

Вы можете использовать методы Insert и Move класса Deck не реализуя их. Если вы пользуетесь дополнительным методами, вы должны реализовать их.

/продолжение на странице 16/

РАЗДЕЛ ТРЕТИЙ (25 баллов)

В данном разделе вопросы четырех разных курсов:

- Компьютерные системы и ассемблер [מערכות מחשב ואסמבלי], стр. 16-17.
- Введение в исследование операций [מבוא לחקר ביצועים], стр. 18-24.
- Вычислительные модели [מודלים חישוביים], стр. 25-26.
- Объектно-ориентированное программирование [תכנות מונחה עצמים] на языке Java, стр. 27-32; объектно-ориентированное программирование на языке C#, стр. 33-38.

Ответьте только на вопросы того курса, который вы изучали.

Компьютерные системы и Ассемблер [מערכות מחשב ואסמבלי]

Если вы изучали этот курс, ответьте на один из вопросов 7-8. (25 баллов)

7. В данном вопросе два не связанных между собой пункта (א) и (ב). Ответьте на оба пункта.

   (א) Перед вами фрагмент программы на Ассемблере:

   Обратите внимание: данные шестнадцатиричные [הקסדצימליים].

```
START:   MOV   AX , C83BH
         MOV   BX , A89CH
         SHL   AX , 1
         OR    AL , 33H
         NOT   BL
         ADD   AX , BX
```

   Проследите с помощью таблицы трассировки [טבלת מעקב] за выполнением данного фрагмента программы. В таблице трассировки детализируйте содержимое CF , SF , ZF , BX , AX на каждом этапе.

/продолжение на странице 17/

(ב) (Нет связи с пунктом (א)).

Во фрагменте данных [מקטע הנתונים] определен массив:

```
ARR    DB     5    DUP (?)
```

Перед вами 4 фрагмента [קטעים] (i)-(iv) на Ассемблере, цель которых записать 0 во все ячейки массива ARR .

Проследите с помощью таблицы трассировки [טבלת מעקב] за выполнением каждого из приведенных ниже фрагментов (i)-(iv) и определите, выполняет он требуемое действие или не выполняет.

i

```
        MOV     SI , 0
        MOV     CX , 4
A1:     MOV     ARR[SI] , 0
        INC     SI
        LOOP    A1
```

---

ii

```
        MOV     CX , 5
        LEA     BX , ARR
        MOV     AL , 0
A1:     MOV     [BX] , AL
        INC     BX
        LOOP    A1
```

---

iii

```
        MOV     BX , 5
        DEC     BX
A1:     MOV     ARR[BX] , 0
        DEC     BX
        JNZ     A1
```

---

iv

```
        MOV     DI , 0
A1:     MOV     ARR[DI] , 0
        INC     DI
        CMP     DI , 5
        JC      A1
```

8. В регистре [אוגר] AX хранится двоичное [בינרי] число.
Напишите на Ассемблере фрагмент программы [קטע תכנית] , который запишет в регистр BL число появлений последовательности 1011 в двоичном числе из регистра AX.
Например: если в регистре AX хранится двоичное число 0110110010110110 , то в регистр BL будет записано число 3 .

/продолжение на странице 18/

Введение в исследование операций [מבוא לחקר ביצועים]

Если вы изучали этот курс, ответьте на два из вопросов 9-10
(за каждый вопрос – 25 баллов).

9. Дана задача линейного программирования [תכנון לינארי]:

$$\text{Max}\{z = 5x_1 - x_2\}$$

при условиях: $2x_1 + 3x_2 \le 18$

$x_1 - x_2 \le 2$

$x_1 \le 3$

$x_2 \ge 0$

x_1 без ограничения знака.

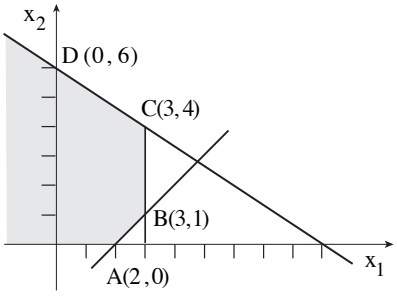
Известно также, что решение (3 , 1) является оптимальным решением данной задачи.

Все последующие пункты (א)-(ה) относятся к данной задаче линейного программирования.

Ответы на вопросы этих пунктов не зависят друг от друга.

Ответьте на вопросы всех пунктов.

(א) Перед вами чертеж области возможных решений данной задачи.



Скопируйте чертеж в вашу тетрадь и выполните следующие шаги:

шаг 1: [צעד 1] Добавьте к чертежу в вашей тетради проекцию высоты целевой функции [היטל הגובה של פונקציית המטרה] для $z = 5$. Вычислите координаты точек пересечения этой проекции с осями x_1 и x_2 и обозначьте их на чертеже.

шаг 2: [צעד 2] Добавьте к чертежу в вашей тетради проекцию высоты целевой функции для $z = 15$. Вычислите координаты точек пересечения этой проекции с осями x_1 и x_2 и обозначьте их на чертеже.

шаг 3: [צעד 3] При помощи стрелки обозначьте на чертеже в вашей тетради направление возрастания целевой функции.

Показывают ли шаги 1-3 , что решение (3 , 1) является единственным оптимальным решением? Обоснуйте ваш ответ.

/продолжение на странице 19/

(ב) К задаче, описанной в начале вопроса, добавили условие:

$$x_1 - x_2 \geq 2$$

Начертите в вашей тетради новую систему осей и обозначьте на ней только область возможных решений, полученную после добавления этого условия.

(ג) К задаче, описанной в начале вопроса, добавляют некоторое условие, после чего областью возможных решений становится отрезок [קטע] между точками $(3\ ,\ 1)$ и $(3\ ,\ 4)$. Изменится ли данное оптимальное решение $(3\ ,\ 1)$? Обоснуйте ваш ответ.

(ד) К задаче, описанной в начале вопроса, добавляют новое условие:

$$3x_1 + 2x_2 \leq 14\ .$$

Ученик утверждал, что данное оптимальное решение $(3\ ,\ 1)$ не изменится.

Прав ли был ученик? Обоснуйте ваш ответ, не изображая на чертеже новую область возможных решений, полученную после добавления этого условия.

(ה) В задаче, описанной в начале вопроса, изменяют только целевую функцию на $z = 3x_1$.

Перед вами утверждения i-iv, только одно из которых является верным. Перепишите в вашу тетрадь верное утверждение и обоснуйте ваш выбор.

i Единственным оптимальным решением будет $(3\ ,\ 1)$.

ii Единственным оптимальным решением будет $(3\ ,\ 4)$.

iii Новое оптимальное решение будет таким:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \lambda\binom{3}{1} + (1-\lambda)\binom{3}{4}$$

где $0 \leq \lambda \leq 1$

iv Оптимальное решение будет неограниченным [לא חסום].

/продолжение на странице 20/

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

(א) Перед вами транспортная задача [בעיית תובלה] с 2-мя пунктами отправления [מקורות] и 4-мя пунктами назначения [יעדים]. Стоимости перевозки единицы груза из каждого пункта отправления в каждый пункт назначения приведены в таблице:

Пункты отправления	**Пункты назначения**				**Предло-жение**
	1	2	3	4	
1	8	9	4	6	100
2	14	12	13	8	80
Спрос	10	40	30	70	

В соответствии с этой таблицей, невозможно применить метод северо-западного угла [שיטת הפינה הצפון-מערבית], чтобы найти возможное решение данной транспортной задачи. Объясните почему.

(ב) В приведенной ниже таблице дана часть возможного базового решения транспортной проблемы: $x_{12} = 2$, $x_{11} = 10$.

Пункты отправления	**Пункты назначения**			**Предло-жение**
	1	2	3	
1	4 (10)	5 (2)	10	12
2	6	3	6	12
3	6	2	6	6
Спрос	10	10	10	

Перепишите эту таблицу в вашу тетрадь и дополните ее в соответствии с методом северо-западного угла .

/продолжение на странице 21/

(ג) В приведенной ниже таблице дано возможное базовое решение транспортной проблемы. Значение, установленное для u_1 , равно 0 . Перепишите эту таблицу в вашу тетрадь, вычислите значения v_1 , v_2 , v_3 , u_2 , u_3 и запишите полученные значения в соответствующие ячейки таблицы.

Пункты отправления	**Пункты назначения**			**Предложение**	u_i
	1	2	3		
1	4 (10)	5 (2)	10	12	0
2	6	3 (2)	6 (10)	12	
3	6	2 (6)	6	6	
Спрос	10	10	10		
v_j					

Обратите внимание: продолжение вопроса на следующей странице

/продолжение на странице 22/

(ד) В приведенной ниже таблице дано возможное базовое решение транспортной проблемы и даны значения v_1 , v_2 , v_3 , u_1 , u_2 , u_3 .

Пункты отправления	**Пункты назначения**			**Предло-жение**	u_i
	1	2	3		
1	4 (20)	5 (4)	10	24	0
2	2	6 (16)	3 (4)	20	1
3	6	2	6 (8)	8	4
Спрос	20	20	12		
v_j	4	5	2		

Является ли это решение оптимальным? Обоснуйте ваш ответ.

/продолжение на странице 23/

(ה) В приведенной ниже таблице дано решение транспортной проблемы и даны значения v_1 , v_2 , v_3 , u_1 , u_2 , u_3 , соответствующие этому решению.

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		

Перед вами утверждения i-iv , только одно из которых является верным. Перепишите в вашу тетрадь верное утверждение и обоснуйте ваш выбор.

i Данное решение не является возможным решением.

ii Данное решение является возможным базовым решением, но не является оптимальным.

iii Данное решение является единственным оптимальным решением.

iv Данное решение является оптимальным, но это не единственное оптимальное решение.

Обратите внимание: продолжение вопроса на следующей странице

/продолжение на странице 24/

(ו) В приведенной ниже таблице дана часть возможного базового решения транспортной проблемы, вычисленного методом северо-западного угла, и даны значения v_1 , v_2 , v_3 , u_1 , u_2 , u_3 , вычисленные в соответствии с этим решением.

Пункты отправления	**Пункты назначения**			**Поставка**	u_i
	1	2	3		
1	12 (20)	15	17	20	2
2	10	18 (10)	14	10	0
3	20	10 (5)	18 (10)	15	−8
Спрос	20	15	10		
v_j	10	18	26		

Что нужно добавить в эту таблицу, чтобы получить возможное базовое решение, соответствующее всем представленным в таблице данным?

Выберите среди предложенных вариантов i-iv правильный ответ и перепишите его в вашу тетрадь. Обоснуйте ваш ответ.

i $X_{12} = 0$.

ii $X_{21} = 0$.

iii $X_{13} = 0$.

iv $X_{31} = 0$.

/продолжение на странице 25/

Вычислительные модели [מודלים חישוביים]

Если вы изучали этот курс, ответьте на один из вопросов 11-12 . (25 баллов)

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

$$L = \left\{ 0^3\$0^{i_1}\$0^{i_2}\$...0^{i_k}\$ \;\middle|\; \begin{array}{l} k \geq 1 \text{ ,} \\ \text{Для всех } m \text{ от } 1 \text{ до } k: \\ i_m \geq 0 \text{ и } i_m \text{ делится на 3} \\ \text{без остатка} \end{array} \right\}$$

(א) Запишите самое короткое слово в языке L .

(ב) Постройте детерминированный конечный автомат, принимающий язык L .

/продолжение на странице 26/

12. Имеется машина Тьюринга, вычисляющая функцию f(m, n).
Машина находит меньшее из двух чисел m и n .
Машина получает в качестве ввода [קלט] два целых числа m и n , больших, чем 0 . Два числа записаны на ленте как унарные числа [מספרים אונריים] (унарное число m – это число, представленное как m единиц), между двумя числами записан символ # .
Машина возвращает меньшее из двух чисел. Результат записывается на ленту как унарное число между двумя символами $. Левый символ $ будет записан вместо символа # , и число будет записано справа от него.
В процессе работы машина может пользоваться символами a и b .
Например: для ввода m = 3 и n = 5 лента памяти перед началом вычисления будет иметь следующий вид:

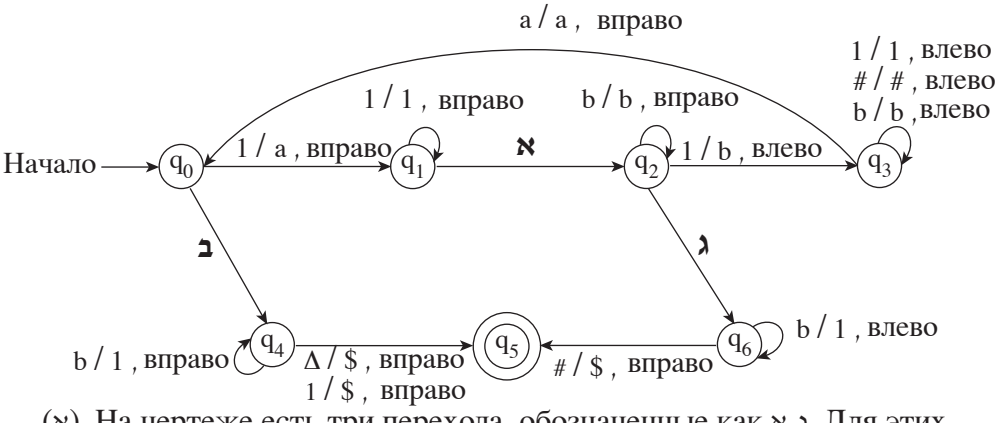
⊢	1	1	1	#	1	1	1	1	1	Δ	Δ	Δ	…

После окончания вычислений, лента будет иметь следующий вид:

⊢	…	…	…	$	1	1	1	$	…

Примечание: ячейки, помеченные как ... – это ячейки, содержимое которых не важно.

Перед вами частичный чертеж этой машины.



(א) На чертеже есть три перехода, обозначенные как א-ג. Для этих переходов не указаны входные символы [סימני הקלט] и команды [ההוראות]. Перенесите чертеж в вашу тетрадь и дополните три недостающих перехода так, чтобы машина вычисляла функцию f(m, n).

(ב) Покажите процесс вычисления машины для ввода: m = 1 , n = 1 . На каждом этапе записывайте состояние ленты, обозначайте место нахождения головки и указывайте, в каком из состояний q_0 - q_6 находится машина.

/продолжение на странице 27/

Объектно-ориентированное программирование [תכנות מונחה עצמים]

Если вы изучали этот курс и пишете на языке Java, ответьте на один из вопросов 13-14. (25 баллов)

13. Перед вами часть проекта, занимающегося транспортными средствами и содержащего следующие классы:

класс Vehicle	Представляет транспортные средства
класс Train	Представляет поезд, который является транспортным средством
класс Boat	Представляет лодку, которая является транспортным средством
класс Airplane	Представляет самолет, который является транспортным средством
класс TransportationCompany	Представляет компанию, владеющую разными видами транспортных средств

```
public class Vehicle
{
    private String type;            //  среда (суша / вода / воздух)
    private String way;             //  тип дороги (шоссе / рельсы / река / ...)
    private int maxSpeed;           //  максимальная скорость

    public Vehicle(String type, String way, int maxSpeed)
    {
        this.type = type;
        this.way = way;
        this.maxSpeed = maxSpeed;
    }
}
```

Обратите внимание: продолжение вопроса на следующих страницах.

/продолжение на странице 28/

```
public class Train extends Vehicle
{
    private int numOfCarriages;          // количество вагонов
    public Train(int maxSpeed, int numOfCarriages)
    {
        super("land", "tracks", maxSpeed);
        this.numOfCarriages = numOfCarriages;
    }
    public void incNumOfCarriages(int n)  // увеличивает на n число вагонов поезда
    {
        this.numOfCarriages = this.numOfCarriages + n;
    }
}
```

```
public class Boat extends Vehicle
{
    public Boat(String way, int maxSpeed)
    {
        super("water", way, maxSpeed);
    }
}
```

```
public class Airplane extends Vehicle
{
    private int maxHeight;    // максимальная высота полета
    public Airplane(int maxSpeed, int maxHeight)
    {
        super("sky" , "air" , maxSpeed);
        this.maxHeight = maxHeight;
    }
}
```

/продолжение на странице 29/

```
public class TransportationCompany
{
    private Vehicle[] vehicles = new Vehicle[50];    // массив транспортных
                                                        средств компании
    private int counter = 0;                         // число имеющихся в
                                                    действительности транспортных средств
    public TransportationCompany()
    {
    }

    public void addVehicle (Vehicle v) // добавляет транспортное средство в массив
                                                        транспортных средств компании
    {                                  // считайте, что место для добавления
                                                        транспортного средства выделено

        this.vehicles[counter] = v;
        this.counter++;
    }
}
```

(א) Реализуйте на языке Java главный класс [מחלקה ראשית] Program, содержащий главный метод, который выполнит следующие задачи:
   (i) Создаст экземпляр класса “Транспортная компания” - **TransportationCompany**, с именем company1.
   (ii) Добавит в company1 одну лодку и один поезд. Выберите для атрибутов класса [תכונות] значения [ערכים] по Вашему желанию.

(ב) В классе TransportationCompany определен следующий метод:

```
public void display()
{
    for (int i=0; i<this.counter; i++)
    {
        System.out.println((i+1) + ":" + this.vehicles[i]);
    }
}
```

/продолжение на странице 30/

Реализуйте на языке Java методы, которые обеспечат корректное выполнение метода display() так, чтобы для каждого транспортного средства на экран выводились все его атрибуты. Определите эти методы в максимальном соответствии с принципами объектно-ориентированного программирования (encapsulation - инкапсуляция, inheritance - наследование, polymorphism - полиморфизм).
Для каждого выполненного Вами метода укажите, к какому классу он принадлежит.
Нельзя менять метод display().

(ג) Напишите на языке Java метод, который получит целое число n и добавит n вагонов ко всем поездам, принадлежащим компании, которая владеет разными видами транспортных средств. Снабдите метод комментариями и укажите, в каком классе он должен быть определен.
Нельзя менять имеющиеся в проекте методы.

/продолжение на странице 31/

14. Перед вами классы **AA** и **BB** :

```
public class AA
{
    private String st;

    public AA()                 {  this.st = "excellent";   }
    public AA(String st)        {  this.st = st;            }
    public String getSt()       {  return this.st;          }
    public void setSt(String st){  this.st = st;            }
    public String toString()    {  return "st = " + this.st; }
}

public class BB extends AA
{
    private int num;

    public BB()                 {  super();  this.num = 1;  }
    public BB(int num , String st) {  super(st); this.num = Math.abs(num);   }
    public int getNum()         {  return this.num;         }
    public void setNum(int num) {  this.num = num;          }
    public String toString()    {  return super.toString() + " num = " + this.num;}
}
```

(א) В классе **AA** определите метод isLike(Object obj) , получающий экземпляр [עצם] obj типа Object. Если экземпляр obj принадлежит типу **AA** и содержимое строки st в obj такое же, как содержимое строки st в данном экземпляре класса, то метод вернет true, иначе – метод вернет false.

(ב) В классе **BB** определите метод, замещающий [דורס] метод, определенный вами в пункте (א). Если экземпляр obj принадлежит типу **BB** и значение атрибута num в нем такое же, как значение атрибута num в данном экземпляре класса, то метод вернет true, иначе – метод вернет false.

/продолжение на странице 32/

(ג) Перед вами фрагмент главного метода [פעולה ראשית]:

```
AA  a = new AA("excellent");
BB  b = new BB();
a = b;
if (a.isLike(b))  System.out.println(a);
```

Является ли данный фрагмент корректным [תקין]?
Если да – каким будет вывод данного фрагмента [פלט]? Запишите, какая версия метода isLike будет применена – версия из AA или из BB.
Если нет – объясните, в чем заключается ошибка и когда она обнаружится: во время компиляции или во время выполнения [זמן ריצה].

(ד) Перед вами фрагмент главного метода:

```
AA  aa = new AA();
BB  bb = new BB(2 , "excellent");
bb = aa;
if (bb.isLike(aa))  System.out.println(bb);
```

Является ли данный фрагмент корректным [תקין]?
Если да – каким будет вывод данного фрагмента [פלט]? Запишите, какая версия метода isLike будет применена – версия из AA или из BB.
Если нет – объясните, в чем заключается ошибка и когда она обнаружится: во время компиляции или во время выполнения [זמן ריצה].

(ה) Напишите внешний метод [פעולה חיצונית] longString, получающий массив [מערך] экземпляров типа Object. Метод возвращает строку, представляющую собой склейку [שרשור] атрибутов st экземпляров типа AA в данном массиве следующим образом:

- Если у экземпляра есть только атрибут st , строка из атрибута st будет добавлена один раз.
- Если у экземпляра есть также атрибут num , то строка из атрибута st будет добавлена num раз.
- Если в массиве нет ни одного экземпляра типа AA, будет возвращена пустая строка.

/продолжение на странице 33/

Объектно-ориентированное программирование
[תכנות מונחה עצמים]

Если вы изучали этот курс и пишете на языке C#, ответьте на один из вопросов 15-16. (25 баллов)

15. Перед вами часть проекта, занимающегося транспортными средствами и содержащего следующие классы:

класс Vehicle	Представляет транспортные средства
класс Train	Представляет поезд, который является транспортным средством
класс Boat	Представляет лодку, которая является транспортным средством
класс Airplane	Представляет самолет, который является транспортным средством
класс TransportationCompany	Представляет компанию, владеющую разными видами транспортных средств

```
public class Vehicle
{
    private string type;        // среда (суша / вода / воздух)
    private string way;         // тип дороги (шоссе / рельсы / река / ...)
    private int maxSpeed;   // максимальная скорость

    public Vehicle(string type, string way, int maxSpeed)
    {
        this.type = type;
        this.way = way;
        this.maxSpeed = maxSpeed;
    }
}
```

/продолжение на странице 34/

```
public class Train : Vehicle
{
    private int numOfCarriages;    // количество вагонов
    public Train(int maxSpeed, int numOfCarriages) : base("land", "tracks",maxSpeed)
    {
        this.numOfCarriages = numOfCarriages;
    }
    public void IncNumOfCarriages(int n)  // увеличивает на n число вагонов в
                                                    поезде
    {
        this.numOfCarriages = this.numOfCarriages + n;
    }
}
```

```
public class Boat : Vehicle
{
    public Boat(string way, int maxSpeed) : base("water", way, maxSpeed)
    {

    }
}
```

```
public class Airplane : Vehicle
{
    private int maxHeight;    // максимальная высота полета
    public Airplane(int maxSpeed, int maxHeight) : base("sky", "air", maxSpeed)
    {
        this.maxHeight = maxHeight;
    }
}
```

/продолжение на странице 35/

```
public class TransportationCompany
{
    private Vehicle[] vehicles = new Vehicle[50];   // массив транспортных
                                                       средств компании
    private int counter = 0;                        // число имеющихся в
                                                    действительности транспортных средств

    public TransportationCompany()
    {
    }

    public void addVehicle (Vehicle v) // добавляет транспортное средство в массив
                                                       транспортных средств компании
    {                                  // считайте, что место для добавления
                                                       транспортного средства выделено
        this.vehicles[counter] = v;
        this.counter++;
    }
}
```

(א) Реализуйте на языке C# главный класс [מחלקה ראשית] Program, содержащий главный метод, который выполнит следующие задачи:
   (i) Создаст экземпляр класса “Транспортная компания” – **TransportationCompany**, с именем company1.
   (ii) Добавит в company1 одну лодку и один поезд. Выберите для атрибутов класса [תכונות] значения [ערכים] по Вашему желанию.

(ב) В классе TransportationCompany определен следующий метод:

```
public void Display()
{
    for (int i=0; i<this.counter; i++)
    {
        Console.Writeln((i+1) + ":" + this.vehicles[i]);
    }
}
```

/продолжение на странице 36/

Реализуйте на языке C# методы, которые обеспечат корректное выполнение метода Display() так, чтобы для каждого транспортного средства на экран выводились все его атрибуты. Определите эти методы в максимальном соответствии с принципами объектно-ориентированного программирования (encapsulation - инкапсуляция, inheritance - наследование, polymorphism - полиморфизм).
Для каждого выполненного Вами метода укажите, к какому классу он принадлежит.
Нельзя менять метод Display().

(ג) Напишите на языке C# метод, который получит целое число n и добавит n вагонов ко всем поездам, принадлежащим компании, которая владеет разными видами транспортных средств. Снабдите метод комментариями и укажите, в каком классе он должен быть определен.
Нельзя менять имеющиеся в проекте методы.

/продолжение на странице 37/

16. Перед вами классы **AA** и **BB**:

```
public class AA
{
    private string st;

    public AA()                          { this.st = "excellent";    }
    public AA(string st)                 { this.st = st;             }
    public string GetSt()                { return this.st;           }
    public void SetSt (string st)        { this.st = st;             }
    public override string ToString()    { return "st = " + this.st; }
}

public class BB : AA
{
    private int num;

    public BB() : base()                          { this.num = 1;    }
    public BB(int num, string st) : base(st)      { this.num = Math.Abs(num);      }
    public int GetNum()                           { return this.num; }
    public void SetNum(int num)                   { this.num = num;}
    public override string ToString()             { return base.ToString() + " num = "
                                                    + this.num;}
}
```

(א) В классе **AA** определите метод IsLike(Object obj) , получающий экземпляр [עצם] obj типа Object. Если экземпляр obj принадлежит типу **AA** и содержимое строки st в obj такое же, как содержимое строки st в данном экземпляре класса, то метод вернет true, иначе – метод вернет false.

(ב) В классе **BB** определите метод, замещающий [דורס] метод, определенный вами в пункте (א). Если экземпляр obj принадлежит типу **BB** и значение атрибута num в нем такое же, как значение атрибута num в данном экземпляре класса, то метод вернет true, иначе – метод вернет false.

/продолжение на странице 38/

(ג) Перед вами фрагмент главного метода [פעולה ראשית]:

```
AA a = new AA("excellent");
BB b = new BB();
a = b;
if (a.IsLike(b)) Console.WriteLine(a);
```

Является ли данный фрагмент корректным [תקין]?

Если да – каким будет вывод данного фрагмента [פלט]? Запишите, какая версия метода IsLike будет применена – версия из AA или из BB.

Если нет – объясните, в чем заключается ошибка и когда она обнаружится: во время компиляции или во время выполнения [זמן ריצה].

(ד) Перед вами фрагмент главного метода:

```
AA  aa = new AA();
BB  bb = new BB(2 , "excellent");
bb = aa;
if (bb.IsLike(aa)) Console.WriteLine(bb);
```

Является ли данный фрагмент корректным [תקין]?

Если да – каким будет вывод данного фрагмента [פלט]? Запишите, какая версия метода IsLike будет применена – версия из AA или из BB.

Если нет – объясните, в чем заключается ошибка и когда она обнаружится: во время компиляции или во время выполнения [זמן ריצה].

(ה) Напишите внешний метод [פעולה חיצונית] LongString, получающий массив [מערך] экземпляров типа Object. Метод возвращает строку, представляющую собой склейку [שרשור] атрибутов st экземпляров типа AA в данном массиве следующим образом:

– Если у экземпляра есть только атрибут st , строка из атрибута st будет добавлена один раз.

– Если у экземпляра есть также атрибут num , то строка из атрибута st будет добавлена num раз.

– Если в массиве нет ни одного экземпляра типа AA, будет возвращена пустая строка.

Желаем успеха! **בהצלחה!**

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

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