מדינת ישראל
משרד החינוך

סוג הבחינה: בגרות לבתי"ס על־יסודיים
מועד הבחינה: קיץ תשע"ו, 2016
מספר השאלון: **899381**
תרגום לערבית (2)

مدعي المحسب

على في تكنية الرفورمة للميدة مشمعوتية

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.
(10x1)+(15x1) — 25 נק'
פרק שני: בפרק זה שלוש שאלות,
ומהן עליך לענות על שתיים.
(25x2) — 50 נק'
פרק שלישי: בפרק זה שאלות בארבעה מסלולים שונים. ענה על שאלה אחת במסלול שלמדת.
(25x1) — 25 נק'
סה"כ — 100 נק'
ג. חומר עזר מותר בשימוש: כל חומר עזר, חוץ ממחשב הניתן לתכנות.
ד. הוראות מיוחדות:
1. את כל התכניות שאתה נדרש לכתוב בשפת מחשב בפרקים הראשון והשני, 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.

دولة إسرائيل
وزارة التّربية والتّعليم

نوع الامتحان: بجروت للمدارس الثّانويّة
موعد الامتحان: صيف 2016
رقم النّموذج: **899381**
ترجمة إلى العربيّة (2)

علم الحاسوب
حسب خطّة الإصلاح: التّعلّم ذي المعنى
تعليمات للممتحَن

أ. مدّة الامتحان: ثلاث ساعات.
ب. مبنى النّموذج وتوزيع الدّرجات:
في هذا النّموذج ثلاثة فصول.
الفصل الأوّل: في هذا الفصل ثلاثة أسئلة، أجب حسب التعليمات في الفصل.
(10x1)+(15x1) – 25 درجة
الفصل الثّاني: في هذا الفصل ثلاثة أسئلة، عليك الإجابة عن اثنين منها.
(25x2) – 50 درجة
الفصل الثّالث: في هذا الفصل أسئلة في أربعة مسارات مختلفة. أجب عن سؤال واحد في المسار الذي تعلّمتَه.
(25x1) – 25 درجة
المجموع – 100 درجة
جـ. موادّ مساعدة يُسمح استعمالها: كلّ مادّة مساعدة، عدا الحاسوب الذي يمكن برمجته.
د. تعليمات خاصّة:
1. اكتب جميع البرامج التي يُطلب منك كتابتها بلغة حاسوب في الفصلين الأوّل والثاني، بلغة واحدة فقط – Java أو C#.
2. **سجّل على الغلاف الخارجيّ** للدّفتر بأيّة **لغة** تكتب – Java أو C#.
3. **سجّل على الغلاف الخارجيّ** للدّفتر، اسم **المسار الذي تعلّمتَه**. المسار هو أحد المسارات الأربعة التّالية: أنظمة حاسوب وأسمبلي، مقدّمة لبحث الأداءات، موديلات حسابيّة، برمجة موجّهة كائنات.

ملاحظة: في البرامج التي تكتبها، لن تُخصم درجات إذا كتبتَ حرفًا كبيرًا بدلاً من حرف صغير أو بالعكس.

اكتب في دفتر الامتحان فقط، في صفحات خاصّة، كلّ ما تريد كتابته مسوّدة (رؤوس أقلام، عمليّات حسابيّة، وما شابه).
اكتب كلمة "مسوّدة" في بداية كلّ صفحة تستعملها مسوّدة. كتابة أيّة مسوّدة على أوراق خارج دفتر الامتحان قد تسبّب إلغاء الامتحان!

التّعليمات في هذا النّموذج مكتوبة بصيغة المذكّر وموجّهة للممتحَنات وللممتحَنين على حدّ سواء.

ב ה צ ל ח ה ! **نتمنّى لك النّجاح!**

الأسئلة

في هذا النموذج ثلاثة فصول.

عليك الإجابة عن أسئلة من ثلاثة الفصول، حسب التعليمات في كلّ فصل.

الفصل الأوّل (25 درجة)

ملاحظة: في كلّ سؤال يُطلب فيه استقبال، لا حاجة لفحص سلامة المدخَلات.

للّذين يحلّون بلغة Java – في كلّ سؤال يُطلب فيه استقبال، افترِضْ أنّ الأمر التالي مكتوب في البرنامج: `Scanner input = new Scanner (System.in);`

أجب عن السؤال 1 – إلزاميّ (10 درجات).

1. معطاة الفئة طالب – **Student** التي لها صفتان: اسم الطالب – name من نمط نصّ، ومصفوفة أحاديّة الأبعاد arrTest بكِبَر 3 من نمط صحيح. في كلّ خليّة في المصفوفة مخزونة علامة الطالب في امتحان معيّن.

اكتب بلغة Java أو بلغة C# عمليّة في الفئة **Student** تُعيد معدّل علامات الطالب في ثلاثة الامتحانات.

ملاحظة: لا حاجة لفحص سلامة المعطيات في المصفوفة.

/ يتبع في صفحة 3 /

أجب عن أحد السؤالين 2-3 (15 درجة) .

2. معطاة الفئة ممثِّل – **Actor** التي لها ثلاث صفات: رقم الهويّة – id من نمط نصّ، وجنس الممثِّل – gender من نمط نصّ ("F" يمثّل أنثى، "M" يمثّل ذكرًا)، وعدد الأفلام التي شارك فيها الممثِّل – numFilms من نمط صحيح.

أمامك واجهة تطبيق الفئة **Actor** المكتوبة بلغة Java وبلغة C# .

وصف العمليّة	عنوان العمليّة بلغة Java
عمليّة تبني ممثِّلاً رقم هويّته id ، وجنسه gender ، وعدد الأفلام التي شارك فيها numFilms .	public Actor(String id, String gender, int numFilms)
عمليّة تُضيف 1 إلى عدد الأفلام التي شارك فيها الممثِّل.	public void addFilm()
عمليّة تُعيد : 1 – إذا كان عدد الأفلام التي شارك فيها الممثِّل الحاليّ أكبر من عدد الأفلام التي شارك فيها الممثِّل other . 2 – إذا كان عدد الأفلام التي شارك فيها الممثِّل الحاليّ أصغر من عدد الأفلام التي شارك فيها الممثِّل other . 3 – إذا كان عدد الأفلام التي شارك فيها الممثِّل الحاليّ يساوي عدد الأفلام التي شارك فيها الممثِّل other .	public int compare(Actor other)

(انتبه : تكملة السؤال في الصفحة التالية .)

/ يتبع في صفحة 4 /

عنوان العمليّة بلغة #C	وصف العمليّة
public Actor(string id, string gender, int numFilms)	عمليّة تبني ممثِّلاً رقم هويّته id، وجنسه gender، وعدد الأفلام التي شارك فيها numFilms .
public void AddFilm()	عمليّة تُضيف 1 إلى عدد الأفلام التي شارك فيها الممثِّل.
public int Compare(Actor other)	عمليّة تُعيد: 1 – إذا كان عدد الأفلام التي شارك فيها الممثِّل الحاليّ أكبر من عدد الأفلام التي شارك فيها الممثِّل other . 2 – إذا كان عدد الأفلام التي شارك فيها الممثِّل الحاليّ أصغر من عدد الأفلام التي شارك فيها الممثِّل other . 3 – إذا كان عدد الأفلام التي شارك فيها الممثِّل الحاليّ يساوي عدد الأفلام التي شارك فيها الممثِّل other .

افترض أنّه تمّ لكلّ صفة تعريف عمليّات get وَ set بلغة Java ، وعمليّات Get وَ Set بلغة C# .

أ. طبِّق العمليّة البنائيّة للفئة **Actor** .

بلغة Java العمليّة البنائيّة:

```
public Actor(String id, String gender, int numFilms)
```

بلغة C# العمليّة البنائيّة:

```
public Actor(string id, string gender, int numFilms)
```

ب. طبِّق بلغة Java العمليّة compare أو بلغة C# العمليّة Compare.

جـ. في العمليّة الرئيسيّة في الفئة Program عُرِّفت مصفوفة أحاديّة الأبعاد actArr بكِبَر 37 من نمط **Actor** .

اكتب بلغة Java أو بلغة C# عمليّة في الفئة Program تتلقّى مصفوفة أحاديّة الأبعاد من نمط **Actor** وعددًا num صحيحًا وأكبر من 0 .

تطبع العمليّة عدد الممثِّلين في المصفوفة الذين شاركوا في أكثر من num أفلام.

/ يتبع في صفحة 5 /

3. معطاة الفئة برنامج تلفزيونيّ – **TvProgram** التي لها ثلاث صفات:
كود البرنامج – code من نمط صحيح، واليوم في الأسبوع الذي يُبثّ فيه البرنامج – day من نمط صحيح، الذي يتلقّى قيَمًا بين 1 حتّى 7 (بما في ذلك 1 وَ 7)، وهل هذا البرنامج هو برنامج رياضيّ – isSport من نمط بوليانيّ، قيمته true إذا كان برنامجًا رياضيًّا، خلاف ذلك – false .
افترض أنّه تمّ لكلّ صفة تعريف عمليّات get وَ set بلغة Java ، وعمليّات Get وَ Set بلغة C# .

أ. اكتب بلغة Java أو بلغة C# عمليّة بنائيّة في الفئة **TvProgram** تتلقّى قيَمًا لكلّ صفة.

ب. معطاة الفئة برامج الأسبوع – **TvWeek** التي لها صفتان:
مصفوفة أحاديّة الأبعاد – arrProg بكِبَر 100 من نمط **TvProgram**،
عدد البرامج الحاليّ في الأسبوع – current من نمط صحيح، قيمته أصغر من كِبَر المصفوفة.
افترض أنّه تمّ لكلّ صفة تعريف عمليّات get وَ set بلغة Java ، وعمليّات Get وَ Set بلغة C# .

(1) اكتب بلغة Java أو بلغة C# عنوان الفئة **TvWeek** وصفاتها.

(2) اكتب بلغة Java أو بلغة C# في الفئة **TvWeek** عمليّة تتلقّى برنامجًا **TvProgram**، وتُضيفه إلى برامج الأسبوع وتُحدِّث عدد البرامج الحاليّ في الأسبوع.
افترض أنّ هناك مكانًا لإضافة البرنامج.

(3) اكتب بلغة Java أو بلغة C# في الفئة **TvWeek** عمليّة تُعيد عدد البرامج الرياضيّة التي في الأسبوع في التلفزيون.

/ يتبع في صفحة 6 /

الفصل الثاني (50 درجة)

انتبه: في كلّ سؤال يُطلب فيه تطبيق، بإمكانك استعمال عمليّات الفئات دور وراصّة وشجرة بيناريّة وحلقة رابطة، بدون تطبيقها.
إذا استعملتَ عمليّات إضافيّة، عليك تطبيقها.

أجب عن اثنين من الأسئلة 4-6 (لكلّ سؤال – 25 درجة).

4. أمامك الفئة خاتم – **Ring** التي لها صفتان:
كِبَر الخاتم من نمط نصّ ("S" – خاتم صغير، "L" – خاتم كبير)؛
عدد صحيح يمثّل لون الخاتم.

C#

```
public class Ring
{
   private string size;  //كِبَر الخاتم
   private int color; //لون الخاتم
   public Ring()
   {
      this.size = "L";
      this.color = 0;
   }
   public Ring(string str, int c)
   {
      this.size = str;
      this.color = c;
   }
   public string GetSize()
   {
      return this.size;
   }
   public int GetColor()
   {
      return this.color;
   }
}
```

Java

```
public class Ring
{
   private String size;  //كِبَر الخاتم
   private int color; //لون الخاتم
   public Ring()
   {
      this.size = "L";
      this.color = 0;
   }
   public Ring(String str, int c)
   {
      this.size = str;
      this.color = c;
   }
   public String getSize()
   {
      return this.size;
   }
   public int getColor()
   {
      return this.color;
   }
}
```

/ يتبع في صفحة 7 /

أمامك واجهة تطبيق الفئة عمود — **Pole**.

العمليّة	C#	Java
عمليّة تبني عمودًا فارغًا. تعقيدات زمن تشغيل العمليّة هي O(1)	public Pole()	public Pole()
عمليّة تُدخِل خاتمًا r إلى رأس العمود. تعقيدات زمن تشغيل العمليّة هي O(1)	public void Add(Ring r)	public void add(Ring r)
عمليّة تُعيد الخاتم الذي في رأس العمود. بالإضافة إلى ذلك، تُخرِج العمليّة الخاتم من رأس العمود. تعقيدات زمن تشغيل العمليّة هي O(1)	public Ring Remove()	public Ring remove()
إذا كان العمود فارغًا تُعيد العمليّة true، خلاف ذلك — تُعيد false. تعقيدات زمن تشغيل العمليّة هي O(1)	public bool IsEmpty()	public boolean isEmpty()
عمليّة ترتّب الخواتم التي على العمود على النحو التالي: الخواتم الكبيرة "موضوعة" في قاع العمود والخواتم الصغيرة فوقها.	public void Sort()	public void sort()

أ. طبِّق العمليّة Sort() بلغة #C أو sort() بلغة Java، التي في الفئة **Pole**.
بإمكانك استعمال العمليّات الإضافيّة للفئة **Pole** دون تطبيقها.
في إجابتك استعمِل فقط الفئتين **Pole** وَ **Ring**.

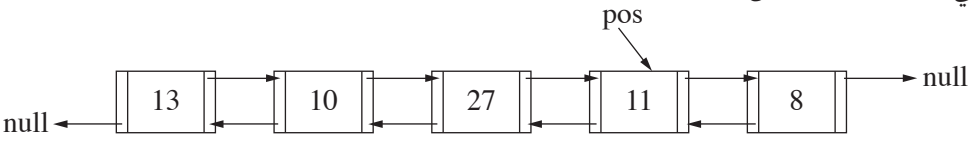
ب. ما هي تعقيدات زمن تشغيل العمليّة التي طبّقتَها في البند "أ"؟ علّل إجابتك.

/ يتبع في صفحة 8 /

5. **انتبه:** لهذا السؤال صيغتان:
الصيغة الأولى بلغة Java في الصفحتين 8-9، والصيغة الثانية بلغة C# في الصفحتين 10-11.
اعمل حسب اللغة التي تعلّمتَها.

للّذين يحلّون بلغة Java

نُعرِّف **قائمة ذات اتّجاهين** كمجموعة منظّمة لحلقات من نمط <BinNode<Integer مرتبطة على النحو التالي:
لكلّ زوج حلقات p2 ,p1 في القائمة، إذا تحقّق p1.getRight() = = p2 ، عندها يتحقّق أيضًا p2.getLeft() = = p1 . **قائمة ذات اتّجاهين** تحوي حلقتين على الأقلّ.
أي: كلّ حلقة في القائمة — باستثناء الحلقة التي في الطرف الأيمن من القائمة والحلقة التي في الطرف الأيسر من القائمة — تشير إلى الحلقة التي قبلها وإلى الحلقة التي بعدها.
أمامك مثال **لقائمة ذات اتّجاهين** ومتغيّر pos من نمط <BinNode<Integer يشير إلى حلقة ما في قائمة ذات اتّجاهين.



العمليّة firstLeft تتلقّى مؤشّرًا pos يختلف عن null من نمط <BinNode<Integer يشير إلى حلقة ما في قائمة ذات اتّجاهين وتُعيد الحلقة التي في أقصى اليسار في القائمة.
العمليّة firstRight تتلقّى مؤشّرًا pos يختلف عن null من نمط <BinNode<Integer يشير إلى حلقة ما في قائمة ذات اتّجاهين وتُعيد الحلقة التي في أقصى اليمين في القائمة.

أ. أمامك هيكل للعمليّة firstLeft.
انسخه إلى دفترك وأكمله، بحيث تنفّذ العمليّة المطلوب.

```
public static BinNode<Integer> firstLeft(BinNode<Integer> pos)
{
    while (_____ )
        pos = _____ ;
    return _____ ;
}
```

/ يتبع في صفحة 9 /

ب. أمامك العمليّة what(BinNode<Integer> pos) التي تتلقّى مؤشّرًا لحلقة ما في **قائمة ذات اتّجاهين** وتُعيد قيمة بوليانيّة. **القائمة ذات الاتّجاهين** تحوي 3 حلقات على الأقلّ.

(1) تتبّع تنفيذ العمليّة بالنسبة للمتغيّر pos والقائمة التي في المثال المعروض في بداية السؤال.

في المتابعة، بيّن القائمة ذات الاتّجاهين وقيَم المتغيّرات pos, left, right, sum.

```
public static boolean what(BinNode<Integer> pos)
{
    BinNode<Integer> left = firstLeft(pos);
    BinNode<Integer> right = firstRight(pos);

    int sum = left.getValue() + right.getValue();
    left = left.getRight();
    right = right.getLeft();

    while ((left != right) && (left.getRight() != right) &&
           (left.getValue() + right.getValue() == sum))
    {
        left = left.getRight();
        right = right.getLeft();
    }
    if (left == right)
        return right.getValue() == sum;

    if (left.getRight() == right)
            return left.getValue() + right.getValue() == sum;
    return false;
}
```

(2) حدِّد إذا كان من الممكن أو من غير الممكن استبدال 3 الأسطر الأخيرة التي في العمليّة — الأسطر المحاطة بإطار — بالأمر:

```
return left.getValue() + right.getValue() == sum;
```

علّل تحديدك.

/ يتبع في صفحة 10 /

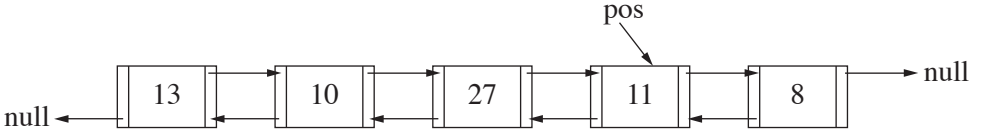
للّذين يحلّون بلغة C#

نُعرِّف **قائمة ذات اتّجاهين** كمجموعة منظّمة لحلقات من نمط <BinNode<int مرتبطة على النحو التالي:

لكلّ زوج حلقات p2, p1 في القائمة، إذا تحقّق p1.GetRight() = = p2 ، عندها يتحقّق أيضًا p2.GetLeft() = = p1 . **قائمة ذات اتّجاهين** تحوي حلقتين على الأقلّ.

أي: كلّ حلقة في القائمة — باستثناء الحلقة التي في الطرف الأيمن من القائمة والحلقة التي في الطرف الأيسر من القائمة — تشير إلى الحلقة التي قبلها وإلى الحلقة التي بعدها.

أمامك مثال **لقائمة ذات اتّجاهين** ومتغيّر pos من نمط <BinNode<int يشير إلى حلقة ما في قائمة ذات اتّجاهين.



العمليّة FirstLeft تتلقّى مؤشّرًا pos يختلف عن null من نمط <BinNode<int يشير إلى حلقة ما في قائمة ذات اتّجاهين وتُعيد الحلقة التي في أقصى اليسار في القائمة.

العمليّة FirstRight تتلقّى مؤشّرًا pos يختلف عن null من نمط <BinNode<int يشير إلى حلقة ما في قائمة ذات اتّجاهين وتُعيد الحلقة التي في أقصى اليمين في القائمة.

أ. أمامك هيكل للعمليّة FirstLeft.

انسخه إلى دفترك وأكمله، بحيث تنفّذ العمليّة المطلوب.

```
public static BinNode<int> FirstLeft(BinNode<int> pos)
{
    while (_____ )
        pos = _____ ;
    return _____ ;
}
```

/ يتبع في صفحة 11 /

ب. أمامك العمليّة What(BinNode<int> pos) التي تتلقّى مؤشّرًا لحلقة ما في **قائمة ذات اتّجاهين** وتُعيد قيمة بوليانيّة. **القائمة ذات الاتّجاهين** تحوي 3 حلقات على الأقلّ.

(1) تتبّع تنفيذ العمليّة بالنسبة للمتغيّر pos والقائمة التي في المثال المعروض في بداية السؤال.

في المتابعة، بيّن القائمة ذات الاتّجاهين وقيَم المتغيّرات pos, left, right, sum.

```
public static bool What(BinNode<int> pos)
{
    BinNode<int> left = FirstLeft(pos);
    BinNode<int> right = FirstRight(pos);

    int sum = left.GetValue() + right.GetValue();
    left = left.GetRight();
    right = right.GetLeft();

    while ((left != right) && (left.GetRight() != right) &&
        (left.GetValue() + right.GetValue() == sum))
    {
        left = left.GetRight();
        right = right.GetLeft();
    }
    if (left == right)
        return right.GetValue() == sum;

    if (left.GetRight() == right)
            return left.GetValue() + right.GetValue() == sum;
    return false;
}
```

(2) حدِّد إذا كان من الممكن أو من غير الممكن استبدال 3 الأسطر الأخيرة التي في العمليّة — الأسطر المحاطة بإطار — بالأمر:

```
return left.GetValue() + right.GetValue() == sum;
```

علّل تحديدك.

/ يتبع في صفحة 12 /

6. **شجرة أعداد** هي شجرة بيناريّة غير فارغة من نمط صحيح، القيَم في عُقَدها هي أعداد صحيحة وأكبر من 0 تختلف عن بعضها البعض.

على **شجرة أعداد** معرَّفة العمليّة "مسار–تصاعديّ"، التي تُعيد true إذا كانت الشجرة تحوي مسارًا يبدأ في جذر الشجرة وينتهي في إحدى أوراقها، وقيَم العُقَد مرتّبة بترتيب تصاعديّ من الجذر إلى الورقة.

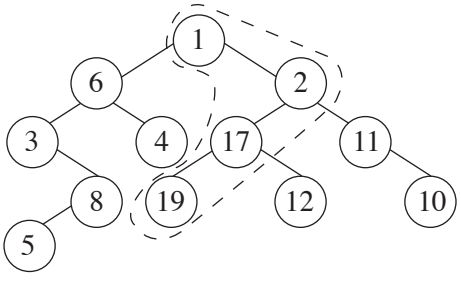
إذا لم تحوِ الشجرة مسارًا كهذا – تُعيد العمليّة false.

على سبيل المثال:

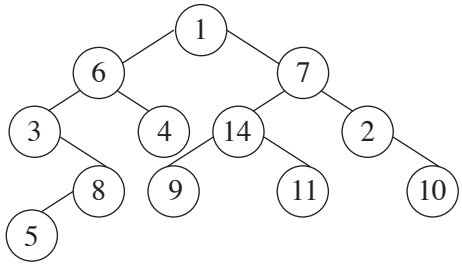
بالنسبة **لشجرة أعداد** tr1 العمليّة "مسار–تصاعديّ" تُعيد true. المسار مُحاط بخطّ متقطّع.

بالنسبة **لشجرة أعداد** tr2 العمليّة "مسار–تصاعديّ" تُعيد false.

tr1



tr2



طبِّق بلغة Java أو بلغة C# العمليّة "مسار – تصاعديّ" بالنسبة **لشجرة أعداد** tr.

عنوان العمليّة بلغة Java: public static boolean upPath(BinNode<Integer> tr)

عنوان العمليّة بلغة C#: public static bool UpPath(BinNode<int> tr)

/ يتبع في صفحة 13 /

الفصل الثالث (25 درجة)

في هذا الفصل أسئلة في أربعة مسارات:
أنظمة حاسوب وأسمبلي، ص 13-16.
مقدّمة لبحث الأداءات، ص 17-21.
موديلات حسابيّة، ص 22.
برمجة موجّهة كائنات بلغة Java، ص 23-27؛ برمجة موجّهة كائنات بلغة #C ، ص 28-32.

أجب عن سؤال واحد في المسار الذي تعلّمتَه.

أنظمة حاسوب وأسمبلي

إذا تعلّمتَ هذا المسار، أجب عن سؤال واحد من السؤالين 7-8 (25 درجة).

7. في هذا السؤال بندان، "أ–ب"، لا علاقة بينهما. أجب عن البندين.

أ. اكتب بلغة أسمبلي قطعة برنامج لتشفير رمز.

الرمز معرَّف على النحو التالي في مقطع المعطيات:

```
TAV    DB ?
```

يتمّ التشفير على النحو التالي:

نَعُدُّ عدد البِتّات المضيئة في الرمز.

إذا كان العدد زوجيًّا – تُنفَّذ إزاحة دائريّة باتّجاه اليمين للبِتّات التي في الرمز، عددًا من المرّات يساوي عدد البِتّات المضيئة.

إذا كان العدد فرديًّا – تُنفَّذ إزاحة دائريّة باتّجاه اليسار للبِتّات التي في الرمز، عددًا من المرّات يساوي عدد البِتّات المضيئة.

ب. (لا علاقة له مع البند "أ".)

بالنسبة لكلّ واحد من الأقوال (1)-(6) التي أمامك، حدِّد إذا كان صحيحًا أم غير صحيح.

إذا كان القول غير صحيح، فسّر لماذا.

(1) أمامك قطعة برنامج بلغة أسمبلي.

```
MOV AX , 8
MOV BX , 2
DIV  BX
```

عند انتهاء قطعة البرنامج، الخازن AX يحوي 4 بالضرورة.

(انتبه: تكملة البند في الصفحة التالية.) / يتبع في صفحة 14 /

(2) أمامك قطعة برنامج بلغة أسمبلي.

```
        MOV     AL , 56
        ADD      AL , 200
        JZ       STOP
        INC      AL
STOP:
```

عند انتهاء قطعة البرنامج، الخازن AX يحوي 1.

(3) أمامك قطعة برنامج بلغة أسمبلي.

```
ARRAY   DW 1, 2, 3, 4
MOV   BX , ARRAY
ADD   BX , 2
MOV   AX , [BX]
```

عند انتهاء قطعة البرنامج، الخازن AX يحوي 3.

(4) أمامك قطعة برنامج بلغة أسمبلي.

```
    MOV  CX , 3
    MOV  AX , 1
DO:
    SHL   AX , 1
    LOOP DO
```

عند انتهاء قطعة البرنامج، الخازن AX يحوي 8.

(5) أمامك قطعة برنامج بلغة أسمبلي.

```
MOV  AX , 11000001B
MOV  BX , 01000001B
```

العدد الموجود في AX يكون بالضرورة أكبر من العدد الموجود في BX.

(6) أمامك الأمر:

```
OR  AL , 3
```

بعد تنفيذ الأمر، تكون قيمة AL فرديّة دائمًا.

/ يتبع في صفحة 15 /

8. في هذا السؤال بندان، "أ–ب"، لا علاقة بينهما. أجب عن البندين.

أ. في مقطع المعطيات عُرِّفت المعطيات على النحو التالي:

```
VEC1  DB  45H , 26H , 32H , 82H
VEC2  DB  4 DUP(0)
```

أمامك قطعة برنامج بلغة أسمبلي.

```
START:  MOV  CL , 4
        XOR  CH , CH
        XOR  SI , SI
NEXT:   PUSH CX
        MOV  AL , VEC1[SI]
        MOV  AH , AL
        AND  AL , 0FH
        MOV  CL , 4
        SHR  AH , CL
        MUL  AH       ;(*)
        MOV  VEC2[SI] , AL
        INC  SI
        POP  CX
        LOOP NEXT
SOF:    NOP
```

(1) تتبّع بواسطة جدول متابعة تنفيذ قطعة البرنامج وارسم VEC2 بعد التنفيذ.
يجب أن يشمل جدول المتابعة عمودًا لكلّ واحد من الخازنات AL , AH , CL , CH , SI.
ارسم الراصّة في كلّ مرحلة.

(2) ما الذي تنفّذه قطعة البرنامج؟

(3) في قطعة البرنامج، بدلاً من الأمر المشار إليه بِـ(*) كُتب الأمر: IMUL AH.
هل يتغيّر تنفيذ قطعة البرنامج؟ علّل إجابتك.

(انتبه: البند "ب" في الصفحة التالية.)

/ يتبع في صفحة 16 /

ب. في الخازن AL مخزون العدد 2 ، وفي الخازن BL مخزون العدد 5.

يجب خزن مجموع الأعداد من AL حتّى BL (بما في ذلك AL وَ BL) في الخازن DX.

أمامك ثلاث قِطَع i -iii بلغة أسمبلي. تتبّع بواسطة جدول متابعة تنفيذ كلّ واحدة من القِطَع، وحدِّد إذا كانت تنفّذ المطلوب أم لا تنفّذ المطلوب.

```
i            MOV    DX , 0
             MOV    AH , 0
             MOV    CL , AL
             SUB    CL , BL
     AGAIN:  ADD    DX , AX
             INC    AL
             LOOPE  AGAIN
             NOP

ii           XOR    DX , DX
             MOV    BH , 0
     AGAIN:  ADD    DX , BX
             DEC    BL
             CMP    BL , AL
             JGE    AGAIN
             NOP

iii          XOR    DX , DX
             XOR    AX , AX
             MOV    BX , 0
     AGAIN:  ADD    DX , AX
             ADD    AX , 1
             CMP    AX , BX
             JL     AGAIN
             NOP
```

/ يتبع في صفحة 17 /

مقدّمة لبحث الأداءات

إذا تعلّمتَ هذا المسار، أجب عن سؤال واحد من السؤالين 9-10 (25 درجة) .

9. معطاة مسألة تخطيط خطّيّ :

$$\max\{z = (2 + 2k)x_1 + 2x_2\}$$

تخضع للاضطرارات التالية :

$$(1) \quad 2x_1 + x_2 \le 10$$

$$(2) \quad x_1 + x_2 \le 6$$

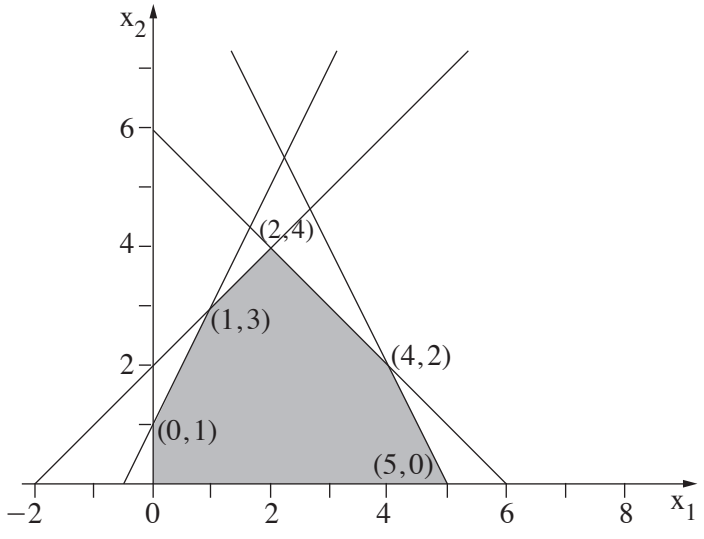
$$(3) \quad -x_1 + x_2 \le 2$$

$$(4) \quad -2x_1 + x_2 \le 1$$

$$x_1 \ge 0$$

$$x_2 \ge 0$$

أمامك رسم لمجال الحلول الممكنة للمسألة المعطاة .



(انتبه : بنود السؤال في الصفحة التالية .)

/ يتبع في صفحة 18 /

كلّ واحد من البنود "أ–د" يتطرّق إلى مسألة التخطيط الخطّيّ المعطاة.
البنود لا تتعلّق ببعضها البعض. أجب عن جميع البنود.

أ. أمامك بندان فرعيّان (1)-(2) **لا علاقة بينهما**. في كلّ واحد منهما معطاة قيمة معيّنة للبارامتر k.

(1) $k = -1$

(2) $k = -3$

بالنسبة لكلّ واحد من البندين الفرعيّين (1)-(2) معطاة أربعة الأقوال i-iv التي أمامك:

i الحلّ الأمثل هو حلّ وحيد.

ii توجد لانهاية من الحلول المثلى.

iii الحلّ الأمثل ليس محصورًا.

iv لا يوجد حلّ أمثل.

وفي كلّ بند فرعيّ فقط أحد الأقوال i-iv صحيح.

بالنسبة لكلّ واحد من البندين الفرعيّين (1)-(2)، حدِّد أيّ قول هو الصحيح، انسخه إلى دفترك، وعلّل تحديدك.

- إذا حدّدتَ أنّ القول i هو الصحيح – عليك إيجاد الحلّ الأمثل الوحيد في هذا البند الفرعيّ، وقيمة دالّة الهدف في الحلّ الذي وجدتَه.
- إذا حدّدتَ أنّ القول ii هو الصحيح –عليك إيجاد الحلّ الأمثل العامّ للمسألة، وقيمة دالّة الهدف في مجال الحلول المثلى.

ب. بالنسبة لأيّة قيَم k يكون $(4\ ,\ 2)$ الحلّ الأمثل لمسألة التخطيط الخطّيّ المعطاة في بداية السؤال؟ علّل إجابتك.

جـ. نلغي الاضطرار $x_1 \geq 0$ الذي في المسألة المعطاة في بداية السؤال، أي $-\infty \leq x_1 \leq \infty$. هل توجد قيمة لـ k يكون بالنسبة لها الحلّ الأمثل ليس محصورًا؟ علّل إجابتك.

د. بدلاً من الاضطرار $x_2 \geq 0$ الذي في المسألة المعطاة في بداية السؤال نكتب: $x_2 \geq 4$. هل يوجد للمسألة حلّ أمثل بعد تغيير الاضطرار؟ إذا كانت إجابتك نعم – اكتب ما هو الحلّ، وإذا كانت إجابتك لا – فسّر لماذا لا يوجد حلّ أمثل.

/ يتبع في صفحة 19 /

10. في هذا السؤال ثلاثة بنود، "أ – جـ"، لا علاقة بينها. أجب عن ثلاثة البنود.

أ. $G = (V , E)$ هو رسم بيانيّ **موجّه** ممثَّل بواسطة مصفوفة المجاوَرات التي أمامك:

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

(1) ارسم الرسم البيانيّ G الممثَّل بواسطة المصفوفة.

(2) جد مركِّبات الربط القويّ (Strong Connected Components) التي في الرسم البيانيّ المعطى. بالنسبة لكلّ مركِّب ربط قويّ وجدتَه، اكتب مجموعة رؤوسه.

(3) جد في الرسم البيانيّ المعطى دائرة بطول أدنى وعدد الأقواس فيها زوجيّ، وارسمها في دفترك.

(4) ما هو العدد الأدنى للأقواس التي يجب إضافتها إلى الرسم البيانيّ المعطى حتّى يحوي الرسم البيانيّ مركِّب ربط قويّ واحدًا فقط؟ ما هو القوس أو ما هي الأقواس التي يجب إضافتها؟

(انتبه: تكملة السؤال في الصفحة التالية.) / يتبع في صفحة 20 /

ب. في الجدول الذي أمامك معطى جزء من حلّ أساسيّ ممكن لمسألة النقل: $x_{11} = 10$.

المصادر	**الغايات**				**العرض**
	1	2	3	4	
1	20 (10)	20	30	21	20
2	22	17	29	30	30
3	10	24	26	38	10
الطلب	10	20	20	10	

(1) انسخ الجدول إلى دفترك، وأكمل فيه الحلّ الأساسيّ الممكن حسب طريقة الزاوية الشماليّة الغربيّة.

(2) نُغيِّر السعر الذي في الخليّة (1 , 3) من 10 إلى 20 . هل يتغيّر الحلّ الأساسيّ الممكن الذي وجدتَه في البند الفرعيّ **(1)** ؟ علّل.

(انتبه: تكملة السؤال في الصفحة التالية.) /يتبع في صفحة 21/

جـ. في الجدول الذي أمامك معطى حلّ أساسيّ ممكن لمسألة نقل، ومعطاة قيمة u_3 .

u_i	**العرض**	**الغايات**			**المصادر**
		3	2	1	
	10	3 (10)	8	10	1
	3	9 (3)	15	12	2
0	90	1 (30)	7 (40)	2 (20)	3
		43	40	20	**الطلب**
					v_j

(1) انسخ الجدول إلى دفترك، وأكمل فيه قيَم $u_1 , u_2 , v_1 , v_2 , v_3$.

(2) فسّر لماذا الحلّ المعطى ليس حلّاً أمثل.

/ يتبع في صفحة 22 /

موديلات حسابيّة

إذا تعلّمتَ هذا المسار، أجب عن سؤال واحد من السؤالين 11-12 (25 درجة) .

11. أمامك اللغتان النظاميّتان L_1 وَ L_2 .

$L_1 = \{a^{2n} \mid n \geq 0\}$ فوق الأبجديّة {a}

$L_2 = \{b^{2n+1} \mid n \geq 0\}$ فوق الأبجديّة {b}

معطاة اللغة L فوق الأبجديّة {a , b} .

$L = \{a^n b^k \mid n \geq 0 , k \geq 0$, { n زوجيّ وَ k فرديّ أو n فرديّ وَ k زوجيّ }

أ. برهن، بواسطة اللغتين L_1 وَ L_2 فقط وبواسطة صفات انغلاق فقط، أنّ اللغة L هي نظاميّة.

ب. ابنِ أوتوماتًا نهائيًّا غير محدود يتلقّى اللغة L .

12. معطاة عمليّة مكتوبة بلغة Java وبلغة C# .

العمليّة تتلقّى ثلاثة أعداد صحيحة أكبر من 0.

C#

```
public static int Foo(int x , int y , int z)
{
    if ((x % 3) == 0) return x;
    if ((x % 3) == 1) return y;
    return z;
}
```

Java

```
public static int foo(int x , int y , int z)
{
    if ((x % 3) == 0) return x;
    if ((x % 3) == 1) return y;
    return z;
}
```

اكتب آلة تيورينج تطبّق العمليّة المعطاة.

مُدخَل الآلة هو ثلاثة الأعداد z , y , x التي تتلقّاها العمليّة، والمُدخَل مكتوب على الشريط من بدايته. كلّ عدد مكتوب بشكل آحاديّ (أونواريّ) . بين عدد وآخر تفصل الإشارة # .

على سبيل المثال إذا كانت العمليّة تتلقّى 2 لِـ x ، وَ 3 لِـ y ، وَ 1 لِـ z ، يبدو الشريط على النحو التالي:

⊢	1	1	#	1	1	1	#	1	Δ	Δ	Δ	.....

↑

رأس يقرأ يكتب

المُخرَج هو القيمة التي تُعيدها العمليّة، ويُكتب على الشريط في مكان ما كقيمة آحاديّة (أونواريّة) بين إشارتَي $.

/ يتبع في صفحة 23 /

برمجة موجّهة كائنات

إذا تعلّمتَ هذا المسار وتكتب بلغة Java، أجب عن سؤال واحد من السؤالين 13-14 (25 درجة).

13. في هذا السؤال بندان، "أ–ب"، لا علاقة بينهما. أجب عن البندين.

أ. معطاة خمس فئات Run , Z , Y , X , B . البنود الفرعيّة (1)-(4) التي أمامك تتطرّق إلى هذه الفئات. لا علاقة بين البنود الفرعيّة. أجب عن جميعها.

(1) في كلّ واحدة من الفئات Z , Y , X , B عُرِّفت العمليّة public void foo() {}
وفي الفئة Run عُرِّفت العمليّة public void bar (Object g) {g.foo(); }

هل ينتج خطأ في التجميع (مسح أوامر)؟ إذا كانت إجابتك نعم – فسّر لماذا.

(2) افترض أنّ الفئات Z , Y , X تَرِثُ الفئة B .
في كلّ واحدة من الفئات Z , Y , X عُرِّفت العمليّة public void foo() {}
وفي الفئة Run عُرِّفت العمليّة public void bar(B g) {g.foo(); }

هل ينتج خطأ في التجميع (مسح أوامر)؟ إذا كانت إجابتك نعم – فسّر لماذا.

(3) افترض أنّ الفئات Z , Y , X تَرِثُ الفئة B .
في كلّ واحدة من الفئات Z , Y , X , B عُرِّفت العمليّة public void foo() {}
وفي الفئة Run عُرِّفت العمليّة public void bar(B g) {g.foo(); }

هل ينتج خطأ في التجميع (مسح أوامر)؟ إذا كانت إجابتك نعم – فسّر لماذا.

(4) افترض أنّ الفئات Z , Y , X تَرِثُ الفئة B .
في كلّ واحدة من الفئات Z , Y , X , B عُرِّفت العمليّة public void foo() {}
وفي الفئة Run عُرِّفت العمليّة public void bar(Object g) {g.foo(); }

هل ينتج خطأ في التجميع (مسح أوامر)؟ إذا كانت إجابتك نعم – فسّر لماذا.

(انتبه: البند "ب" في الصفحة التالية.)

/ يتبع في صفحة 24 /

ב. (لا علاقة له مع البند "أ" .)

أمامك الفئة **Singer** التي تَرِثُ الفئة **Artist** وواجهة التطبيق **IPerform** التي تطبّقها الفئة **Singer**.

```
interface IPerform
{
   void act();
   int train();
}
public class Singer extends Artist implements IPerform
{
      private int hits;
      public Singer(String name, double sal)
      {
         super(sal, name);
         this.hits = 5;
      }
      public Singer(double sal, int hits)
      {
         super(sal, "Singer Name");
         this.hits = hits;
      }
      public Singer(int hits)
      {
         super(6532.6, "Some", "One");
         this.hits = hits;
      }
      public double value() {return this.hits * this.price();}
      public int getNum()    {return Artist.num;                    }
      public double calc(double d)    {return d * super.calc(10.2);  }
      public void print()      {super.print(); System.out.println("Singer");}
      public void act()        {System.out.println("I am singing");  }
}
```

اكتب بلغة Java عنوان الفئة **Artist** ، والصفات وعناوين العمليّات التي توجبها الفئة **Singer** وواجهة التطبيق **IPerform** المعطاتان. لا حاجة لتطبيق عمليّات الفئة **Artist**.

يجب عدم تغيير الفئة **Singer** وواجهة التطبيق **IPerform**.

/ يتبع في صفحة 25 /

14. أمامك الفئتان A, B.

ثلاثة البنود "أ–جـ" تتطرّق إلى هاتين الفئتين، لكن لا علاقة بينها. أجب عن البنود الثلاثة.

```
public class A    {
      private int x;
      public A()                { this.x = 0;       }
      public A(int x)           { this.x = x;                }
      public int getX()         { return x;                  }
      public void doubleX()  { this.x = 2 * getX();          }
      public void tripleX()     { this.x = 3 * getX();       }
      public void sub()         { this.x = x - 1;            }
      public void calc()        { sub();                     }
      public String toString()     { return "xA="+this.x; }
}
public class B extends A  {
      private int x;
      public B()                    { super();   this.x = 1;          }
      public B(int x)               { super(x);  this.x = - x;        }
      public B(int xA, int xB)      { super(xA); this.x = xB;         }
      public int getX()             { return x;                       }
      public int baseX()       { return super.getX();                 }
      public void tenTimesX()       { this.x = 10 * getX();           }
      public void tripleX()         { this.x = 3 * getX();            }
      public void sub()             { this.x = x - 2;                 }
      public String toString()      { return super.toString()+" xB="+this.x+";";      }
}
```

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 26 /

أ. أمامك سلسلة أوامر التي المُخرَج بالنسبة لها يجب أن يكون:

xA=1 xB=20; xA=1 xB=20; xA=1 xB=20; xA=1 xB=20;

وقع خطأ في التجميع (مسح أوامر) في سلسلة الأوامر. صحِّح الخطأ كي ينتج المُخرَج الصحيح.

```
A a1 = new B(1, 20);
Object obj = a1;
B b1 = a1;
A a2 = a1;
System.out.println(a1+" "+obj+" "+a2+" "+b1);
```

ب. أمامك سلسلة أوامر. اعرض الكائنات التي تنتج، وبالنسبة لكلّ كائن اعرض قيَم صفاته.

```
A aa = new B(3, 10);
aa.sub();
Object[] ar = new Object[6];
ar[0] = new A();
ar[1] = new A(5);
ar[2] = new B();
ar[3] = new B(5);
ar[4] = new B(2, 4);
ar[5] = aa;
((A)ar[3]).tripleX();
((B)ar[4]).tenTimesX();
```

(انتبه: تكملة السؤال في الصفحة التالية.) /يتبع في صفحة 27/

جـ. أمامك عمليّة رئيسيّة.

```
public static void main(String[] args)
{
    A a1 = new A(1);
    A a2 = new B(2, 99);
    //***
}
```

أمامك القِطَع (i)-(vi).

(i)
```
a2.doubleX();
System.out.println(a2);
```

(ii)
```
a2.tenTimesX();
System.out.println(a2.tenTimesX());
```

(iii)
```
if (a2 instanceof B)
{
  a2.tenTimesX();
  System.out.println(a2);
}
```

(iv)
```
((B)a1).tenTimesX();
System.out.println(a1);
```

(v)
```
a2.calc();
System.out.println(a2);
```

(vi)
```
B bb = (B)a2;
System.out.println(bb.baseX());
```

بالنسبة لكلّ واحدة من القِطَع (i)-(vi) نفِّذ:

- اكتب القطعة في المكان ***// في العمليّة الرئيسيّة.
- حدِّد إذا كانت القطعة سليمة أم غير سليمة.
- إذا كانت القطعة سليمة – اكتب المُخرَج الذي ينتج في أعقاب تنفيذها.

إذا كانت القطعة غير سليمة – اكتب إذا كان الخطأ هو خطأ في التجميع (مسح أوامر) أم خطأ في زمن التشغيل (تنفيذ أوامر).

/ يتبع في صفحة 28 /

برمجة موجّهة كائنات

إذا تعلّمتَ هذا المسار وتكتب بلغة C#، أجب عن سؤال واحد من السؤالين 15-16 (25 درجة) .

15. في هذا السؤال بندان، "أ–ب"، لا علاقة بينهما. أجب عن البندين.

أ. معطاة خمس فئات Run , Z , Y , X , B . البنود الفرعيّة (1)-(4) التي أمامك تتطرّق إلى هذه الفئات. لا علاقة بين البنود الفرعيّة. أجب عن جميعها.

(1) في كلّ واحدة من الفئات Z , Y , X , B عُرِّفت العمليّة public virtual void Foo(){}

وفي الفئة Run عُرِّفت العمليّة public void Bar(Object g) {g.Foo(); }

هل ينتج خطأ في التجميع (مسح أوامر)؟ إذا كانت إجابتك نعم – فسّر لماذا.

(2) افترض أنّ الفئات Z , Y , X تَرِثُ الفئة B .

في كلّ واحدة من الفئات Z , Y , X عُرِّفت العمليّة public virtual void Foo() {}

وفي الفئة Run عُرِّفت العمليّة public void Bar(B g) {g.Foo();

هل ينتج خطأ في التجميع (مسح أوامر)؟ إذا كانت إجابتك نعم – فسّر لماذا.

(3) افترض أنّ الفئات Z , Y , X تَرِثُ الفئة B .

في كلّ واحدة من الفئات Z , Y , X , B عُرِّفت العمليّة public virtual void Foo(){}

وفي الفئة Run عُرِّفت العمليّة public void Bar(B g) {g.Foo(); }

هل ينتج خطأ في التجميع (مسح أوامر)؟ إذا كانت إجابتك نعم – فسّر لماذا.

(4) افترض أنّ الفئات X , Y , Z تَرِثُ الفئة B .

في كلّ واحدة من الفئات Z , Y , X , B عُرِّفت العمليّة public virtual void Foo() {}

وفي الفئة Run عُرِّفت العمليّة public void Bar(Object g) {g.Foo(); }

هل ينتج خطأ في التجميع (مسح أوامر)؟ إذا كانت إجابتك نعم – فسّر لماذا.

/ يتبع في صفحة 29 /

ב. (لا علاقة له مع البند "أ" .)

أمامك الفئة **Singer** التي تَرِثُ الفئة **Artist** وواجهة التطبيق **IPerform** التي تطبّقها الفئة **Singer**.

```
interface IPerform
{
    void Act();
    int Train();
}
public class Singer : Artist , IPerform
{
    private int hits;
    public Singer(string name, double sal) : base(sal, name)
    {
        this.hits = 5;
    }
    public Singer(double sal, int hits) : base(sal, "Singer Name")
    {
        this.hits = hits;
    }
    public Singer(int hits) : base(6532.6, "Some", "One")
    {
        this.hits = hits;
    }
    public double Value()          {return this.hits * this.Price(); }
    public int GetNum()            {return Artist.num;               }
    public override double Calc(double d)  {return d * base.Calc(10.2); }
    public override void Print()   {base.Print(); Console.WriteLine("Singer");      }
    public void Act()              {Console.WriteLine("I am singing");       }
}
```

اكتب بلغة C# عنوان الفئة **Artist** ، والصفات وعناوين العمليّات التي توجبها الفئة **Singer** وواجهة التطبيق **IPerform** المعطاتان. لا حاجة لتطبيق عمليّات الفئة **Artist**.

يجب عدم تغيير الفئة **Singer** وواجهة التطبيق **IPerform**.

/ يتبع في صفحة 30 /

16. أمامك الفئتان A, B.

ثلاثة البنود "أ–جـ" تتطرّق إلى هاتين الفئتين، لكن لا علاقة بينها. أجب عن البنود الثلاثة.

```
public class A    {
      private int x;
      public A()                        { this.x = 0;              }
      public A(int x)                   { this.x = x;              }
      public virtual int GetX()         { return x;                }
      public void DoubleX()             { this.x = 2 * GetX();     }
      public virtual void TripleX()     { this.x = 3 * GetX();     }
      public virtual void Sub()         { this.x = x - 1;          }
      public void Calc()                { Sub();                   }
      public override string ToString(){ return "xA="+this.x;  }
}
public class B:A  {
      private int x;
      public B() : base()               { this.x = 1;              }
      public B(int x) : base(x)         { this.x = -x;             }
      public B(int xA, int xB) : base(xA)       { this.x = xB;     }
      public override int GetX()        { return x;                }
      public int BaseX()                { return base.GetX();      }
      public void TenTimesX()           { this.x = 10 * GetX();    }
      public override void TripleX()  { this.x = 3 * GetX();     }
      public override void Sub()        { this.x = x - 2;          }
      public override string ToString(){ return base.ToString()+" xB="+this.x+";";}
}
```

/ يتبع في صفحة 31 /

أ. أمامك سلسلة أوامر التي المُخرَج بالنسبة لها يجب أن يكون:

xA=1 xB=20; xA=1 xB=20; xA=1 xB=20; xA=1 xB=20;

وقع خطأ في التجميع (مسح أوامر) في سلسلة الأوامر. صحِّح الخطأ كي ينتج المُخرَج الصحيح.

```
A a1 = new B(1, 20);
Object obj = a1;
B b1 = a1;
A a2 = a1;
Console.WriteLine(a1+" "+obj+" "+a2+" "+b1);
```

ب. أمامك سلسلة أوامر. اعرض الكائنات التي تنتج، وبالنسبة لكلّ كائن اعرض قيَم صفاته.

```
A aa = new B(3, 10);
aa.Sub();
Object[] ar = new Object[6];
ar[0] = new A();
ar[1] = new A(5);
ar[2] = new B();
ar[3] = new B(5);
ar[4] = new B(2, 4);
ar[5] = aa;
((A)ar[3]).TripleX();
((B)ar[4]).TenTimesX();
```

(انتبه: تكملة السؤال في الصفحة التالية.) /يتبع في صفحة 32/

جـ. أمامك عمليّة رئيسيّة.

```
public static void Main()
{
    A a1 = new A(1);
    A a2 = new B(2, 99);
    //***
}
```

أمامك القِطَع (i)-(vi).

```
(i)   a2.DoubleX();
      Console.WriteLine(a2);
(ii)  a2.TenTimesX();
      Console.WriteLine(a2.TenTimesX());
(iii) if (a2 is B)
      {
      a2.TenTimesX();
      Console.WriteLine(a2);
      }
(iv)  ((B)a1).TenTimesX();
      Console.WriteLine(a1);
(v)   a2.Calc();
      Console.WriteLine(a2);
(vi)  B bb = (B)a2;
      Console.WriteLine(bb.BaseX());
```

بالنسبة لكلّ واحدة من القِطَع (i)-(vi) نفِّذ :

- اكتب القطعة في المكان ***// في العمليّة الرئيسيّة.
- حدِّد إذا كانت القطعة سليمة أم غير سليمة.
- إذا كانت القطعة سليمة – اكتب المُخرَج الذي ينتج في أعقاب تنفيذها.

إذا كانت القطعة غير سليمة – اكتب إذا كان الخطأ هو خطأ في التجميع (مسح أوامر) أم خطأ في زمن التشغيل (تنفيذ أوامر).

בהצלחה!
نتمنّى لك النّجاح!

זכות היוצרים שמורה למדינת ישראל.
אין להעתיק או לפרסם אלא ברשות משרד החינוך.
حقوق الطّبع محفوظة لدولة إسرائيل.
النّسخ أو النّشر ممنوعان إلّا بإذن من وزارة التّربية والتّعليم.