מדינת ישראל
משרד החינוך

סוג הבחינה: א. בגרות לבתי"ס על־יסודיים
ב. בגרות לנבחנים חיצוניים
מועד הבחינה: קיץ תשע"ו, 2016
מספר השאלון: **899205, 603**
תרגום לערבית (2)

مدعي המחשב ב'

2 יחידות לימוד (השלמה ל־5 יח"ל)

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.
(25x2) — 50 נק'
פרק שני: בפרק זה שאלות בארבעה מסלולים שונים. ענה על שאלות רק במסלול שלמדת, לפי ההוראות בקבוצת השאלות במסלול זה.
(25x2) — 50 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 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.

دولة إسرائيل
وزارة التّربية والتّعليم

نوع الامتحان: أ. بجروت للمدارس الثّانويّة
ب. بجروت للممتحَنين الخارجيّين
موعد الامتحان: صيف 2016
رقم النّموذج: **899205، 603**
ترجمة إلى العربيّة (2)

علم الحاسوب "ب"

وحدتان تعليميّتان (تكملة لـ 5 وحدات تعليميّة)

تعليمات للممتحَن

أ. مدّة الامتحان: ثلاث ساعات.
ب. مبنى النّموذج وتوزيع الدّرجات:
في هذا النّموذج فصلان.
الفصل الأوّل: في هذا الفصل أربعة أسئلة، عليك الإجابة عن اثنين منها.
(25x2) – 50 درجة
الفصل الثّاني: في هذا الفصل أسئلة في أربعة مسارات مختلفة. أجب فقط عن سؤالين في المسار الذي تعلّمتَه، حسب التّعليمات في مجموعة الأسئلة في هذا المسار.
(25x2) – 50 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 الفصل الأوّل، 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 فصلان: الفصل الأوّل والفصل الثاني.
عليك الإجابة عن أسئلة من الفصلين، حسب التعليمات في كلّ فصل.

الفصل الأوّل (50 درجة)

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

أجب عن اثنين من الأسئلة 1-4 (لكلّ سؤال – 25 درجة).

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

أ. أمامك عمليّة تتلقّى كبارامتر دورًا q لأعداد صحيحة مرتّبة بترتيب تصاعديّ.
جميع الأعداد تختلف عن بعضها البعض.
افترض أنّ الدور يحوي عددين صحيحين على الأقلّ.

Java

```
public static int what(Queue<Integer > q)
{
    int x = q.remove();
    int y = q.head();
    q.insert(x);
    if (x>y)
        return 0;
    return(what(q) + (y−x));
}
```

C#

```
public static int What(Queue <int > q)
{
    int x = q.Remove();
    int y = q.Head();
    q.Insert(x);
    if (x>y)
        return 0;
    return  (What(q) + (y−x));
}
```

/ يتبع في صفحة 3 /

(1) معطى الدور q1.

تتبّع العمليّة what بلغة Java أو What بلغة C# بالنسبة للدور المعطى q1 واكتب القيمة التي تُعاد. بيّن المتابعة التراجعيّة.
في المتابعة، بيّن قيَم x وقيَم y والدور بعد كلّ استدعاء تراجعيّ.

(2) (i) ماذا تُعيد العمليّة what بلغة Java أو What بلغة C# بالنسبة لدور ما q لأعداد صحيحة، مرتّبة بترتيب تصاعديّ وتختلف عن بعضها البعض؟

(ii) كيف يبدو الدور بعد نهاية العمليّة؟

(3) لو استبدلنا الشرط if(x>y) بالشرط:

if(q.isEmpty()) بلغة Java

أو:

if(q.IsEmpty()) بلغة C#

سيحدث خطأ في زمن التشغيل.
ما هو الخطأ الذي سيحدث؟ فسّر إجابتك.

(انتبه: البند "ب" في الصفحة التالية.)

/ يتبع في صفحة 4 /

ب. (لا علاقة له مع البند "أ".)

أمامك عمليّة تتلقّى مصفوفة أحاديّة الأبعاد ar من نمط صحيح، وعددين صحيحين x وَ y أكبر من 0 أو يساويان 0، وأصغر من كِبَر المصفوفة، وَ $x \le y$.

C#

```
public static int Go(int[] ar , int x , int y)
{
    if ( x == y )   return (ar[x] % 10) ;
    if (( ar[x] % 10) < (ar[y] % 10 ))
        return Go(ar, x + 1, y);
    return   Go(ar, x, y−1);
}
```

Java

```
public static int go(int[] ar , int x , int y)
{
    if ( x == y )   return (ar[x] % 10) ;
    if (( ar[x] % 10) < (ar[y] % 10))
        return go(ar, x + 1, y);
    return   go(ar, x, y−1);
}
```

أمامك المصفوفة ar1 من نمط صحيح.

	0	1	2	3	4	5	6
ar1	21	219	166	61	127	60	34

(1) ماذا يُعيد الاستدعاء بلغة Java go(ar1 , 2 , 5) أو بلغة C# Go(ar1 , 2 , 5) ؟
بيّن المتابعة التراجعيّة. في المتابعة بيّن قيَم:
x , y , ar1[x] , ar1[y] .

(2) ماذا يُعيد الاستدعاء بلغة Java go(ar1, 0, ar1.length −1)
أو بلغة C# Go(ar1, 0, ar1.Length−1) ؟

(3) اكتب ما الذي تنفّذه العمليّة:

بلغة Java: public static int go(int[] ar , int x , int y)

بلغة C#: public static int Go(int[] ar , int x , int y)

بالنسبة لمصفوفة ar ما من نمط صحيح وعددين صحيحين x وَ y ،
أكبر من 0 أو يساويان 0، وأصغر من كِبَر المصفوفة، وَ $x \le y$.

/ يتبع في صفحة 5 /

2. **شجرة أعداد** هي شجرة بينارّية 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.

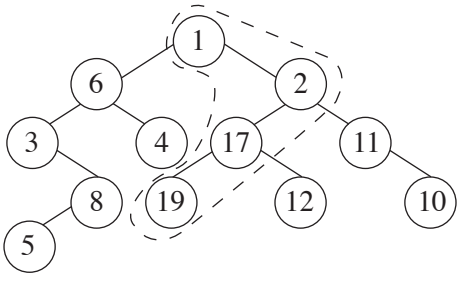
إذا لم تحوِ الشجرة مسارًا كهذا – تُعيد العمليّة false.

على سبيل المثال:

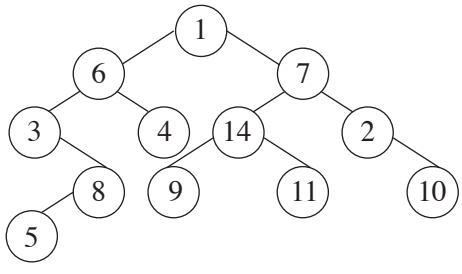
بالنسبة **لشجرة أعداد** tr1 العمليّة "مسار–تصاعديّ" تُعيد true. المسار مُحاط بخطّ متقطّع.

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 6 /

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

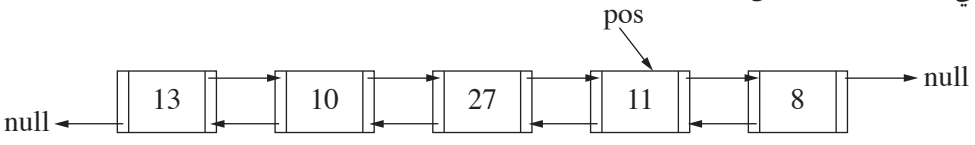
ب. ما هي تعقيدات زمن تشغيل العمليّة التي طبّقتَها في البند "أ"؟ علّل إجابتك.

/ يتبع في صفحة 8 /

4. **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 /

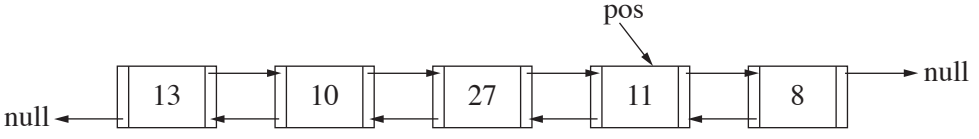
للّذين يحلّون بلغة #C

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 /

الفصل الثاني (50 درجة)

في هذا الفصل أسئلة في أربعة مسارات:
أنظمة حاسوب وأسمبلي، ص 12-18.
مقدّمة لبحث الأداءات، ص 19-28.
موديلات حسابيّة، ص 29-30.
برمجة موجّهة كائنات بلغة Java، ص 31-40؛ برمجة موجّهة كائنات بلغة C# ، ص 41-50.

أجب فقط عن سؤالين في المسار الذي تعلّمتَه.

أنظمة حاسوب وأسمبلي

إذا تعلّمتَ هذا المسار، أجب عن اثنين من الأسئلة 5-8 (لكلّ سؤال – 25 درجة).

5. 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
```

/ يتبع في صفحة 13 /

(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 بـ (*) كُتب الأمر: IMUL AH.
هل يتغيّر تنفيذ قطعة البرنامج؟ علّل إجابتك.

ب. (لا علاقة له مع البند "أ".)
أمامك قطعة برنامج بلغة أسمبلي.

```
MOV   AX , 14H
MOV   BL , 0AH
DIV   BL
MOV   CL , 4
ROL   AL , CL
OR    AL , AH
```

(1) تتبّع بواسطة جدول متابعة تنفيذ قطعة البرنامج.
يجب أن يشمل جدول المتابعة عمودًا لكلّ واحد من
الخازنات AX , AL , AH , BL , CL.

(2) ما الذي تنفّذه قطعة البرنامج؟

/ يتبع في صفحة 14 /

6. من أجل أن نفحص إذا كان عدد معيّن يقسم على 6 بدون باقٍ نقسمه على 2.
إذا كان يقسم على 2 بدون باقٍ — نقسم النتيجة على 3.
إذا كانت النتيجة تقسم على 3 بدون باقٍ — فإنّ العدد يقسم على 6 بدون باقٍ.

أ. اكتب بلغة أسمبلي برنامجًا فرعيًّا باسم DIVBYTWO يتلقّى، بواسطة الراصّة، عددًا مكوّنًا من 16 بِتًّا، ويخزن 0 في AL إذا كان العدد يقسم على 2 بدون باقٍ.
خلاف ذلك — يخزن البرنامج الفرعيّ 1 في AL.
في كتابة البرنامج الفرعيّ لا تستعمل الأمر DIV.

ب. اكتب بلغة أسمبلي برنامجًا فرعيًّا باسم DIVBYTHREE يتلقّى، بواسطة الراصّة، عددًا مكوّنًا من 16 بِتًّا، ويخزن 0 في AL إذا كان العدد يقسم على 3 بدون باقٍ.
خلاف ذلك — يخزن البرنامج الفرعيّ 1 في AL.
في كتابة البرنامج الفرعيّ يمكن استعمال أيّ أمر.

جـ. اكتب بلغة أسمبلي برنامجًا فرعيًّا باسم DIVBYSIX يتلقّى، بواسطة الراصّة، عددًا مكوّنًا من 16 بِتًّا، ويخزن 0 في AL إذا كان العدد يقسم على 6 بدون باقٍ.
خلاف ذلك — يخزن البرنامج الفرعيّ 1 في AL.
عليك استعمال البرنامجين الفرعيّين اللذين كتبتَهما في البندين "أ–ب".

د. في مقطع المعطيات عُرّفت مصفوفة ARR فيها 8 حدود كلّ واحد منها بِكبَر كلمة.
المصفوفة تحوي أعدادًا.
اكتب بلغة أسمبلي قطعة برنامج تطبع عدد الحدود في المصفوفة، التي القيَم المخزونة فيها تقسم على 6 بدون باقٍ.
عليك استعمال البرنامج الفرعيّ الذي كتبتَه في البند "جـ".

/ يتبع في صفحة 15 /

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.

(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.

(انتبه: تكملة البند في الصفحة التالية.)

/ يتبع في صفحة 16 /

(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 17 /

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

أ. 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
```

(انتبه: البند "ب" في الصفحة التالية.)

/ يتبع في صفحة 18 /

ب. (لا علاقة له مع البند "أ".)

أمامك قطعة برنامج بلغة أسمبلي.

انتبه: الأعداد هي سداسيّة عشريّة (هكساديسيماليّة).

```
START:  MOV   AX , 0C72AH
        MOV   BX , 0A98DH
        SHR   AX , 1
        OR    AL , 17H
        NOT   BH
        ADD   AX , BX
```

تتبّع بواسطة جدول متابعة تنفيذ قطعة البرنامج.

يجب أن يشمل جدول المتابعة عمودًا لكلّ واحد من الأعلام:

, ZF , SF , OF , CF

ولكلّ واحد من الخازنَيْن:

.BX , AX

/ يتبع في صفحة 19 /

مقدّمة لبحث الأداءات

إذا تعلّمتَ هذا المسار، أجب عن اثنين من الأسئلة 9-12 (لكلّ سؤال – 25 درجة).

9. معطاة مسألة تخطيط خطّيّ:

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

تخضع للاضطرارات التالية:

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

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

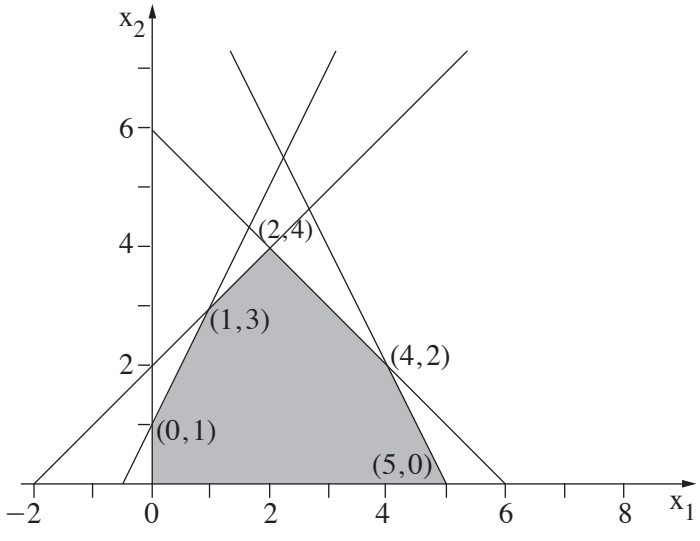
$$(3) \quad -x_1 + x_2 \le 2$$

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

$$x_1 \ge 0$$

$$x_2 \ge 0$$

أمامك رسم لمجال الحلول الممكنة للمسألة المعطاة.



(انتبه: بنود السؤال في الصفحة التالية.)

/ يتبع في صفحة 20 /

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 21 /

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

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؟

ب. (لا علاقة له مع البند "أ".)

$G = (V , E)$ هو رسم بيانيّ **ليس موجّهًا** ممثَّل بواسطة قائمة المجاوَرات التي أمامك:

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

(1) ارسم الرسم البيانيّ G الممثَّل بواسطة قائمة المجاوَرات.

(2) كم مركِّب ربط (Connected Components) يوجد في الرسم البيانيّ G ، وما هي؟

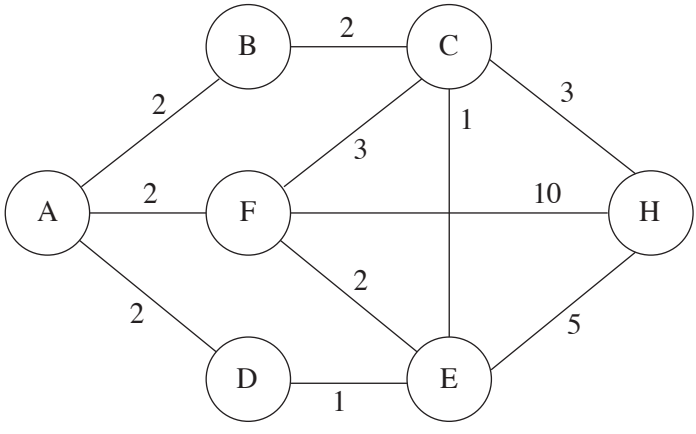
(3) فعِّل ألغوريثم تمشيط عميق (DFS) على الرسم البيانيّ G ابتداءً من الرأس a .
ارسم في دفترك فقط الشجرة الامتداديّة (DFS) / الغابة الامتداديّة (DFS) التي تنتج.
اعتمد على التمثيل المعطى بواسطة قائمة المجاوَرات.

(4) فعِّل ألغوريثم تمشيط عرضيّ (BFS) على الرسم البيانيّ G ابتداءً من الرأس a .
ارسم في دفترك فقط الشجرة الامتداديّة (BFS) / الغابة الامتداديّة (BFS) التي تنتج.
اعتمد على التمثيل المعطى بواسطة قائمة المجاوَرات.

/ يتبع في صفحة 22 /

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

أ. أمامك الشبكة $G = (V , E)$ بحيث V هو مجموعة الرؤوس في الشبكة وَ E هو مجموعة الأقواس في الشبكة. دالّة الوزن $w: E \to R^+$ تحدّد وزنًا (عددًا أكبر من 0) لكلّ قوس في الشبكة.



(1) جد جميع أقصر مسارات في الشبكة المعطاة، ابتداءً من الرأس A وحتّى الرأس H.
اعرض وصفًا تخطيطيًّا لكلّ واحد من المسارات التي وجدتَها.

(2) جد الشجرة الامتداديّة الصغرى حسب ألغوريثم كروسكل، من بين جميع الأشجار الامتداديّة التي تحوي القوسين التاليين: (A, B) وَ (B, C).
ارسم في دفترك الشجرة الامتداديّة الصغرى المطلوبة.

/ يتبع في صفحة 23 /

ب. (لا علاقة له مع البند "أ" .)

معطاة الشبكة $G = (V , E)$ بحيث V هو مجموعة الرؤوس في الشبكة وَ E هو مجموعة الأقواس في الشبكة. دالّة الوزن $w: E \to R^+$ تحدّد وزنًا (عددًا أكبر من 0) لكلّ قوس في الشبكة.

كما ومعطى القوس $e = (a , b)$، والرأسان s وَ t في الشبكة المعطاة.

أمامك ألغوريثم يُعيد القيمة 1، إذا كان القوس المعطى e موجودًا على جميع أقصر مسارات، من الرأس s حتّى الرأس t . خلاف ذلك – يُعيد الألغوريثم القيمة 0.

في الألغوريثم **تعبيران** ناقصان، أُشير إليهما بِـ (1)، (2).

الألغوريثم:

الخطوة 1: فعِّل ألغوريثم **ديكسترا** على الشبكة G ، واحفظ في المتغيّر M1 طول أقصر مسار من الرأس s حتّى الرأس t .

الخطوة 2: ابنِ شبكة جديدة $G1 = (V1 , E1)$ فيها $V1 = V$ ، $E1 =$ ____(1)____ .

الخطوة 3: فعِّل ألغوريثم **ديكسترا** على الشبكة G1، واحفظ في المتغيّر M2 طول أقصر مسار من الرأس s حتّى الرأس t .

الخطوة 4: إذا كان ____(2)____ أَعِد 1.

خلاف ذلك – أَعِد 0.

انسخ إلى دفترك الرقمين (1) وَ (2)، واكتب بجانب كلّ واحد منهما التعبير الناقص الذي يمثّله.

/ يتبع في صفحة 24 /

12. في هذا السؤال خمسة بنود، "أ–هـ"، لا علاقة بينها. أجب عن جميع البنود.

أ. 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 25 /

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

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

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 26 /

جـ. في الجدول الذي أمامك معطاة مسألة نقل، ومعطاة قيَم $u_1, u_2, u_3, v_1, v_2, v_3$.

u_i	**العرض**	**الغايات**			**المصادر**
		3	2	1	
1	30	1	8	8	1
6	40	6	12	10	2
1	74	2	7	2	3
		50	64	30	**الطلب**
		0	6	1	v_j

(1) انسخ الجدول إلى دفترك، واكتب فيه حلّاً أساسيًّا ممكنًا مع الأخذ بالحسبان قيَم $u_1, u_2, u_3, v_1, v_2, v_3$.

(2) نُغيِّر في الجدول المعطى تكاليف النقل من المصدر 1 إلى الغاية 2 من 8 إلى 7.
على ضوء هذا التغيير، حدِّد أيّ قول من الأقوال i-iv التي أمامك هو القول الصحيح، وانسخه إلى دفترك. علّل تحديدك.

i الحلّ الأساسيّ لا يتغيّر، ويبقى حلّاً وحيدًا يلائم قيَم u_i وَ v_j المعطاة.

ii يوجد حلّان أساسيّان ممكنان **فقط** يلائمان قيَم u_i وَ v_j المعطاة.

iii يوجد لانهاية من الحلول الأساسيّة الممكنة التي تلائم قيَم u_i وَ v_j المعطاة.

iv على ضوء التغيير، لا يمكن إيجاد حلّ أساسيّ ممكن يلائم قيَم u_i وَ v_j المعطاة.

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 27 /

د. في الجدول الذي أمامك معطى حلّ ممكن لمسألة نقل، ومعطاة قيَم $u_1, u_2, u_3, v_1, v_2, v_3$ التي تلائم هذا الحلّ.

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

حدِّد أيّ قول من الأقوال i-iv التي أمامك هو القول الصحيح، وانسخه إلى دفترك.
علّل تحديدك.

i الحلّ المعطى ليس حلّاً أساسيًّا ممكنًا.
ii الحلّ المعطى هو حلّ أساسيّ ممكن لكنّه ليس حلّاً أمثل.
iii الحلّ المعطى هو حلّ أمثل وحيد.
iv الحلّ المعطى هو حلّ أمثل لكنّه ليس حلّاً أمثل وحيدًا.

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 28 /

هـ. في الجدول الذي أمامك معطى حلّ ليس أمثل نتج بعد k تكرارات بالنسبة لمسألة نقل معيّنة، ومعطاة قيَم $u_1, u_2, u_3, v_1, v_2, v_3, v_4$.

المصادر	الغايات				العرض	u_i
	1	2	3	4		
1	4 (6)	4 (4)	14	25	10	3
2	6	1 (5)	13 (10)	14 (3)	18	0
3	0	8	14	12 (2)	2	−2
الطلب	6	9	10	5		
v_j	1	1	13	14		

عليك تنفيذ تكرار إضافيّ، أي التكرار $k+1$.

(1) ما هو المتغيّر الذي يخرج من القاعدة في هذا التكرار؟

(2) ارسم في دفترك جدولاً جديدًا، واكتب فيه الحلّ الذي ينتج بعد هذا التكرار.

/ يتبع في صفحة 29 /

موديلات حسابيّة

إذا تعلّمتَ هذا المسار، أجب عن اثنين من الأسئلة 13-16 (لكلّ سؤال – 25 درجة).

13. 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.

14. أمامك اللغات L_1 , L_2 , L_3 , L_4 ، فوق الأبجديّة $\{1 , 0\}$.

$L_1 = \{w \mid$ تحوي التسلسل 010 $w\}$

$L_2 = \{w \mid$ لا تحوي التسلسل 00 $w\}$

$L_3 = \{w \mid$ عدد الأصفار في w لا يساوي عدد الآحاد في $w\}$

$L_4 = \{w \mid$ عدد الأصفار في w هو فرديّ $\}$

أمامك الادّعاءات (1)-(6).

بالنسبة لكلّ واحد من الادّعاءات، حدِّد إذا كان صحيحًا أم غير صحيح.

إذا كان الادّعاء صحيحًا – فسّر لماذا.

إذا كان الادّعاء غير صحيح – فسّر لماذا أو اكتب كلمة تدحض الادّعاء وفسّر لماذا تدحضه.

(1) $L_1 \cup L_3 \cup L_4$ تتبع للّغة ε

(2) $L_1 \cap \overline{L_4}$ تتبع للّغة 00100

(3) $\overline{L_2} = \{w \mid$ تحوي التسلسل 11 $w\}$

(4) $L_1 \subseteq L_3$

(5) $L_4 = R(L_4)$

(6) $L_3 \cap L_4 = L_2$

/ يتبع في صفحة 30 /

15. معطاة اللغة L فوق الأبجديّة $\{a , b , c\}$.

$$L = \{a^{2n} c^{k} b^{3n+1} \mid n > 0 , k > 0\}$$

أ. اكتب أقصر كلمة في اللغة L .

ب. ابنِ أوتومات راصّة يتلقّى اللغة L .

16. 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 x , y , z التي تتلقّاها العمليّة، والمُدخَل مكتوب على الشريط من بدايته. كلّ عدد مكتوب بشكل آحاديّ (أونارِيّ). 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 (أونارِيّة) بين إشارتَي $.

/ يتبع في صفحة 31 /

برمجة موجّهة كائنات

إذا تعلّمتَ هذا المسار وتكتب بلغة Java، أجب عن اثنين من الأسئلة 17-20 (لكلّ سؤال – 25 درجة).

17. 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 32 /

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 33 /

18. في حديقة حيوانات توجد الحيوانات التالية: أسود Lion، وبطّ Platypus، وثعابين Snake وسلاحف Turtle.

الأسود والبطّ هي ثدييّات، الثعابين والسلاحف هي زواحف.

إناث جميع الزواحف في حديقة الحيوانات تضع بيضًا، وكذلك أنثى البطّ تضع بيضًا.

يرغبون في بناء منظومة مُحَوْسَبة تعرض الحيوانات في حديقة الحيوانات. تعرض المنظومة صورًا عاديّة وصورًا متحرّكة.

لهذا الغرض عُرِّفت الفئات **Animal** — حيوان، **Mammal** — ثدييّ، **Reptile** — زاحف.

لكلّ حيوان توجد صفة: اسم — name، وعمليّتان: void move() وَ void turn().

للحيوانات التي تضع بيضًا توجد الصفة: معدّل عدد البيض في مرّة وضع البيض – numOfEggs وعمليّة: int getNumOfEggs().

أ. ارسم هرميّة فئات حديقة الحيوانات وواجهات التطبيق، إذا دعت الحاجة، بالطريقة الأكثر ملاءمة لمبادئ البرمجة الموجّهة كائنات (احتواء — encapsulation، توريث — inheritance، تعدّد الأشكال — polymorphism).

أشر في الرسم إلى الأربطة التي بين الفئات، وإلى الأربطة التي بين الفئات وبين واجهات التطبيق، إذا كانت هناك واجهات تطبيق.

لا حاجة لشمل صفات وعمليّات في الرسم.

ب. بالنسبة لكلّ واحدة من الفئات التي في رسمك، اكتب بلغة Java عنوان الفئة وصفاتها، وعناوين لجميع العمليّات.

بالنسبة لكلّ واحدة من واجهات التطبيق التي في رسمك، اكتب بلغة Java عنوانها، وعناوين لجميع العمليّات.

لستَ بحاجة لكتابة عمليّات get وَ set وعمليّات بنائيّة.

(انتبه: تكملة السؤال في الصفحة التالية.) /يتبع في صفحة 34/

جـ. في جميع الفئات عُرِّفت العمليّة String details().
هذه العمليّة تُعيد نصًّا مكوّنًا من اسم الحيوان، بحيث تُذكَر قبل اسم الحيوان أسماء الفئات في هرميّة توريث الحيوان ابتداءً من **Animal**.
على سبيل المثال:
بالنسبة للكائن ln من نمط **Lion** الذي اسمه Simba، العمليّة details تُعيد النصّ (من اليسار إلى اليمين):
Animal , Mammal , Lion , Simba
بالنسبة للكائن plp من نمط **Platypus** الذي اسمه Perry، العمليّة details تُعيد النصّ (من اليسار إلى اليمين):
Animal , Mammal , Platypus , Perry
طبِّق بلغة Java العمليّة details في الفئات **Animal** وَ **Mammal** وَ **Lion** وَ **Platypus**, كي تُعاد النصوص الموصوفة. طبِّق العمليّة بالطريقة الأكثر ملاءمة لمبادئ البرمجة الموجّهة كائنات (احتواء – encapsulation، توريث – inheritance، تعدّد الأشكال – polymorphism).

د. العمليّة toString تستدعي العمليّة details. إذا كان الحيوان يضع بيضًا، العمليّة toString تُضيف إلى النصّ الكلمات Number of eggs ومعدّل عدد البيض في مرّة وضع البيض، وتُعيد النّصّ.
طبِّق بلغة Java العمليّة toString في كلّ فئة تظهر فيها، بالطريقة الأكثر ملاءمة لمبادئ البرمجة الموجّهة كائنات (احتواء – encapsulation، توريث – inheritance، تعدّد الأشكال – polymorphism).

/ يتبع في صفحة 35 /

19. أمامك الفئات **Bill** , **PhoneBill** , **Shopping** , **MyBills** ، وعمليّة رئيسيّة في الفئة Tester .
تتبّع تنفيذ العمليّة الرئيسيّة، واكتب المُخرَج الذي ينتج.
يجب أن تشمل المتابعة قيَم المتغيّرات، وبالنسبة لكلّ كائن قيَم صفاته.

```
public class Bill {
	public final static char DOLLAR = 'd';
	public final static char SHEKEL = 's';
	private static double dollarRate = 4.00;
	private double total;
	private char curr;
	public double getTotal() {
		if (curr == DOLLAR)
		  return total * dollarRate;
		return total;
	}
	public Bill(double total, char curr) {
		this.total = total;
		this.curr = curr;
	}
	public String toString() {
		String s = getTotal() + " Shekels";
		return s;
	}
}

public class PhoneBill extends Bill {
	private String company;
	public PhoneBill(String company, double value, char curr) {
		super(value, curr);
		this.company = company;
	}
	public PhoneBill(PhoneBill pb) {
		super(pb.getTotal(), SHEKEL);
		company = pb.company;
	}
	public String toString() {
		String s = "Pay: " + super.toString() + " to: " + company;
		return s;
	}
}
```

(انتبه: تكملة السؤال في الصفحة التالية.) / يتبع في صفحة 36 /

```
public class Shopping extends Bill {
    private String name;
    private String list;
    public Shopping(String name, String list, double value, char curr) {
        super(value, curr);
        this.name = name;
        this.list = list;
    }
    public Shopping(Shopping s) {
        super(s.getTotal(), SHEKEL);
        name = s.name;
        list = s.list;
    }
    public String toString() {
        String s = name + " bought: " + list + ". Pay: " + super.toString();
        return s;
    }
}

public class MyBills {
    private int num;
    private Bill[] bills;
    private final static int MAX = 4;
    public MyBills() {
        bills = new Bill[MAX];
        num = 0;
    }
    public void addBill(Bill b) {
        if (num < MAX) {
          if (b instanceof PhoneBill)
            bills[num] = new PhoneBill((PhoneBill) b);
          else if (b instanceof Shopping)
            bills[num] = new Shopping((Shopping) b);
          else num--;
          num++;
        }
    }
```

/ يتبع في صفحة 37 /

```
    public double total() {
        double t = 0;
        for (int i = 0; i < num; i++)
            t = t + bills[i].getTotal();
        return t;
    }
    public void print() {
        for (int i = 0; i < num; i++)
            System.out.println(bills[i].toString());
    }
}

public class Tester {
    public static void main(String[] args) {
        MyBills baba = new MyBills();
        Shopping s = new Shopping("Baba", "Books", 250, Bill.DOLLAR);
        PhoneBill t = new PhoneBill("TeleBaba", 2000, Bill.SHEKEL);
        PhoneBill u = new PhoneBill("Galil Telecom", 200, Bill.DOLLAR);
        baba.addBill(t);
        baba.addBill(s);
        baba.addBill(u);
        baba.print();
        System.out.println(baba.total());
    }
}
```

/ يتبع في صفحة 38 /

20. أمامك الفئتان B, A.

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+";";      }
}
```

/ يتبع في صفحة 39 /

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 40/

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 41 /

برمجة موجّهة كائنات

إذا تعلّمتَ هذا المسار وتكتب بلغة C#، أجب عن اثنين من الأسئلة 21-24 (لكلّ سؤال – 25 درجة).

21. 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 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(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 42 /

ب. (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 43 /

22. في حديقة حيوانات توجد الحيوانات التالية: أسود Lion، وبطّ Platypus، وثعابين Snake وسلاحف Turtle.

الأسود والبطّ هي ثدييّات، الثعابين والسلاحف هي زواحف.

إناث جميع الزواحف في حديقة الحيوانات تضع بيضًا، وكذلك أنثى البطّ تضع بيضًا.

يرغبون في بناء منظومة مُحَوْسَبة تعرض الحيوانات في حديقة الحيوانات. تعرض المنظومة صورًا عاديّة وصورًا متحرّكة.

لهذا الغرض عُرِّفت الفئات **Animal** – حيوان، **Mammal** – ثدييّ، **Reptile** – زاحف.

لكلّ حيوان توجد صفة: اسم – name، وعمليّتان: void Move() وَ void Turn().

للحيوانات التي تضع بيضًا توجد الصفة: معدّل عدد البيض في مرّة وضع البيض – numOfEggs وعمليّة: int GetNumOfEggs().

أ. ارسم هرميّة فئات حديقة الحيوانات وواجهات التطبيق، إذا دعت الحاجة، بالطريقة الأكثر ملاءمة لمبادئ البرمجة الموجّهة كائنات (احتواء – encapsulation، توريث – inheritance، تعدّد الأشكال – polymorphism).

أشر في الرسم إلى الأربطة التي بين الفئات، وإلى الأربطة التي بين الفئات وبين واجهات التطبيق، إذا كانت هناك واجهات تطبيق.

لا حاجة لشمل صفات وعمليّات في الرسم.

ب. بالنسبة لكلّ واحدة من الفئات التي في رسمك، اكتب بلغة C# عنوان الفئة وصفاتها، وعناوين لجميع العمليّات.

بالنسبة لكلّ واحدة من واجهات التطبيق التي في رسمك، اكتب بلغة C# عنوانها، وعناوين لجميع العمليّات.

لستَ بحاجة لكتابة عمليّات Get وَ Set وعمليّات بنائيّة.

(انتبه: تكملة السؤال في الصفحة التالية.) /يتبع في صفحة 44/

جـ. في جميع الفئات عُرِّفت العمليّة String Details().
هذه العمليّة تُعيد نصًّا مكوّنًا من اسم الحيوان، بحيث تُذكَر قبل اسم الحيوان أسماء الفئات في هرميّة توريث الحيوان ابتداءً من **Animal**.
على سبيل المثال:
بالنسبة للكائن ln من نمط **Lion** الذي اسمه Simba، العمليّة Details تُعيد النصّ (من اليسار إلى اليمين):
Animal , Mammal , Lion , Simba
بالنسبة للكائن plp من نمط **Platypus** الذي اسمه Perry ، العمليّة Details تُعيد النصّ (من اليسار إلى اليمين):
Animal , Mammal , Platypus , Perry
طبِّق بلغة C# العمليّة Details في الفئات **Animal** وَ **Mammal** وَ **Lion** وَ **Platypus**, كي تُعاد النصوص الموصوفة. طبِّق العمليّة بالطريقة الأكثر ملاءمة لمبادئ البرمجة الموجّهة كائنات (احتواء – encapsulation ، توريث – inheritance ، تعدّد الأشكال – polymorphism).

د. العمليّة ToString تستدعي العمليّة Details. إذا كان الحيوان يضع بيضًا، العمليّة ToString تُضيف إلى النصّ الكلمات Number of eggs ومعدّل عدد البيض في مرّة وضع البيض، وتُعيد النّصّ.
طبِّق بلغة C# العمليّة ToString في كلّ فئة تظهر فيها، بالطريقة الأكثر ملاءمة لمبادئ البرمجة الموجّهة كائنات (احتواء – encapsulation ، توريث – inheritance ، تعدّد الأشكال – polymorphism).

/ يتبع في صفحة 45 /

23. أمامك الفئات **Bill** , **PhoneBill** , **Shopping** , **MyBills** ، وعمليّة رئيسيّة في الفئة Tester .
تتبّع تنفيذ العمليّة الرئيسيّة، واكتب المُخرَج الذي ينتج.
يجب أن تشمل المتابعة قيَم المتغيّرات، وبالنسبة لكلّ كائن قيَم صفاته.

```
public class Bill {
    public const char DOLLAR = 'd';
    public const char SHEKEL = 's';
    private static double dollarRate = 4.00;
    private double total;
    private char curr;
    public double GetTotal()    {
        if (curr == DOLLAR)
            return total * dollarRate;
        return total;
    }
    public Bill(double total, char curr)  {
        this.total = total;
        this.curr = curr;
    }
    public override string ToString()    {
        string s = GetTotal() + " Shekels";
        return s;
    }
}

public class PhoneBill : Bill {
    private string company;
    public PhoneBill(string company, double value, char curr) : base(value, curr) {
        this.company = company;
    }
    public PhoneBill(PhoneBill pb) : base(pb.GetTotal(), SHEKEL){
        company = pb.company;
    }
    public override string ToString() {
        string s = "Pay: " + base.ToString() + " to: " + company;
        return s;
    }
}
```

(انتبه: تكملة السؤال في الصفحة التالية.) / يتبع في صفحة 46 /

```
public class Shopping : Bill {
    private string name;
    private string list;
    public Shopping(string name, string list, double value, char curr) : base(value, curr){
        this.name = name;
        this.list = list;
    }
    public Shopping(Shopping s):base (s.GetTotal(), SHEKEL) {
        name = s.name;
        list = s.list;
    }
    public override string ToString() {
        string s = name + " bought: " + list + ". Pay: " + base.ToString();
        return s;
    }
}
public class MyBills {
    private int num;
    private Bill[] bills;
    private const int MAX = 4;
    public MyBills() {
      bills = new Bill[MAX];
      num = 0;
    }
    public void AddBill(Bill b) {
    if (num < MAX) {
        if (b is PhoneBill)
                bills[num] = new PhoneBill((PhoneBill) b);
        else if (b is Shopping)
            bills[num] = new Shopping((Shopping) b);
        else num--;
        num++;
        }
    }
```

/ يتبع في صفحة 47 /

```
        public double Total() {
            double t = 0;
            for (int i = 0; i < num; i++)
                t = t + bills[i].GetTotal();
            return t;
        }
        public void Print() {
            for (int i = 0; i < num; i++)
                Console.WriteLine(bills[i].ToString());
        }
}


public class Tester  {
        public static void Main(string[] args) {
            MyBills baba = new MyBills();
            Shopping s = new Shopping("Baba", "Books", 250, Bill.DOLLAR);
            PhoneBill t = new PhoneBill("TeleBaba", 2000, Bill.SHEKEL);
            PhoneBill u = new PhoneBill("Galil Telecom", 200, Bill.DOLLAR);
            baba.AddBill(t);
            baba.AddBill(s);
            baba.AddBill(u);
            baba.Print();
            Console.WriteLine(baba.Total());
        }
}
```

/ يتبع في صفحة 48 /

24. أمامك الفئتان B, A.

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 49 /

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 50 /

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).

ב ה צ ל ח ה!

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.