מדינת ישראל
משרד החינוך

סוג הבחינה: בגרות
מועד הבחינה: קיץ תשע"ז, 2017
מספר השאלון: **899205, 603**
תרגום לערבית (2)

מדעי 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: بچروت
موعد الامتحان: صيف 2017
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 وحلقة وحلقة بينارية، بدون تطبيقها.
إذا استعملتَ عمليّات إضافيّة، عليك تطبيقها.

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

1. أمامك تعريف لخمس عمليّات تعمل على مبنى معطيات معيّن.

انتبه: أسماء العمليّات التي أمامك ليست مكتوبة بلغة Java أو بلغة C# .

insert(x) — عمليّة تُدخِل إلى المبنى حدًّا قيمته x من نمط صحيح.

showMin() — عمليّة تُعيد أقلّ قيمة في المبنى، دون تغيير المبنى.

getMax() — عمليّة تُعيد الحدّ الذي قيمته هي الأكبر في المبنى، وتُخرِجه من المبنى.
إذا كان أكثر من حدّ واحد كهذا، تُعيد العمليّة وتُخرِج الحدّ الذي يظهر أوّلاً.

exists(x) — عمليّة بوليانيّة تُعيد true إذا كان الحدّ الذي قيمته x موجودًا في المبنى.
خلاف ذلك – تُعيد العمليّة false.

div7() — عمليّة بوليانيّة تُعيد true إذا كان في المبنى حدّ تقسم قيمته على 7 بدون باقٍ. خلاف ذلك – تُعيد العمليّة false.

نريد اقتراح مباني معطيات تفي بمتطلّبات تعقيدات مختلفة لتطبيق عمليّات من العمليّات الخمس التي عُرِّفت.

مثال: نريد اقتراح مبنى معطيات يمكن أن ننفّذ عليه العمليّتين insert , showMin بتعقيدات O(1) ، والعمليّتين getMax , exists بتعقيدات O(n) . لهذا الغرض نُعرِّف مبنى المعطيات ونشرح كيفيّة تطبيق العمليّات.

انتبه: في هذا المبنى لا حاجة للتطرُّق إلى العمليّة div7.

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 3 /

مبنى المعطيات الملائم مكوَّن من:

- قائمة مرتبطة ذات اتّجاهين، lst من نمط صحيح.
- مؤشّر إلى الحدّ الأدنى الذي أُدخِل إلى القائمة، min .

تُنفَّذ العمليّات على النحو التالي:

العمليّة	شرح كيفيّة تطبيقها	تعليل لماذا يفي التطبيق بمتطلّبات التعقيدات
insert(x)	– إدخال الحدّ x إلى رأس القائمة. – إذا كان الحدّ أصغر من الأدنى حتّى الآن، تعديل المؤشّر min بحيث يؤشّر إلى الحدّ الجديد.	– إدخال حدّ إلى رأس القائمة – O(1) . – إذا أُدخِل حدّ أصغر من الأدنى: تعديل المؤشّر إلى الحدّ الأدنى – O(1) . – إذا أُدخِل حدّ أوّل – إدخال إلى رأس القائمة بـ O(1) وتعديل المؤشّر إلى الحدّ الأوّل الذي هو أيضًا الأدنى. المجموع – O(1).
showMin()	إعادة قيمة الحدّ الذي يؤشّر إليه min .	إعادة القيمة – O(1) .
exists(x)	مسح القائمة lst والبحث عن الحدّ الذي قيمته x .	في أسوأ الحالات – مسح كلّ القائمة – O(n) .
getMax()	مسح القائمة lst، والبحث عن الحدّ الذي قيمته قصوى وإخراجه من القائمة.	– البحث عن الحدّ الذي قيمته قصوى O(n). – إخراجه من القائمة O(1) .

أمامك يندان أ–ب. بالنسبة لكلّ واحد من البندين عليك اقتراح مبنى معطيات ملائم يفي بالمتطلّبات المفضَّلة في البند. يمكن أن يكون المبنى مركَّبًا من دمج عدّة مبانٍ وأنماط تعلّمتَها.

لكلّ واحدة من العمليّات، اشرح كيف تطبّقها، وعلّل لماذا يفي التطبيق بالمتطلّبات (كما عُرض في الجدول الذي في المثال). لا حاجة لتطبيق العمليّات.

أ. تنفيذ العمليّتين insert ، exists بتعقيدات O(n)، وتنفيذ العمليّتين showMin ، getMax بتعقيدات O(1) .

ب. تنفيذ العمليّتين insert ، getMax بتعقيدات O(n) وتنفيذ العمليّة div7 بتعقيدات O(1) .

/ يتبع في صفحة 4 /

2. أمامك برنامج مكتوب بلغة Java وبلغة C# .

العمليّة what بلغة Java وَ What بلغة C# تتلقّى مصفوفة أحاديّة الأبعاد تحوي أعدادًا صحيحة وعددًا صحيحًا k ، $k > 0$. جميع الأعداد في المصفوفة هي بين 0 وَ k (بما في ذلك 0 وَ k).

Java

```
public class Program
{
    public static int[]  what(int[] arr, int k)
    {
       int n = arr.length;
       int[] b = new int [n];
       int[] c = new int [k+1];

       for (int i = 0; i < k+1; i++) c[i] = 0;
       for (int j = 0; j < n; j++)  c[arr[j]] = c[arr[j]] + 1;      //*

       for (int i = 1; i < k+1 ; i++) c[i] = c[i] + c[i−1];        //**

       for (int j = n−1 ; j >= 0; j--)                              //***
       {
          b[c[arr [j]] − 1] = arr[j];
          c[arr[j]] = c[arr[j]] −1;
       }
       return b;
    }


    public static void main(String[] args)
    {
         int[] arr = new int[] {5,0,2,1,3,0,5};
         arr = what(arr, 5);
         for (int i = 0; i < arr.length; i++) System.out.println(arr[i]);
    }
}
```

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 5 /

C#

```
public class Program
{
    public static int[]  What(int[] arr, int k)
    {
       int n = arr.Length;
       int[] b = new int [n];
       int[] c = new int [k+1];

       for (int i = 0; i < k+1; i++) c[i] = 0;
       for (int j = 0; j< n; j++)  c[arr[ j]] = c[arr[ j]] + 1;      //*

       for (int i = 1; i< k+1 ; i++) c[i] = c[i] + c[i−1];          //**

       for (int j = n−1 ; j >= 0; j--)                               //***
       {
          b[c[arr[ j]] −1] = arr[ j];
          c[arr[ j]] = c[arr[ j]] −1;
       }
       return b;
    }

    public static void Main(string[] args)
    {
         int[] arr = new int[] {5,0,2,1,3,0,5};
         arr = What(arr, 5);
         for (int i = 0; i < arr.Length; i++) Console.WriteLine(arr[i]);
    }
}
```

أ. **(1)** ارسم المصفوفة c بعد تنفيذ الحلقة المُشار إليها بِـ *// .

(2) ارسم المصفوفة c بعد تنفيذ الحلقة المُشار إليها بِـ **// .

(3) تتبّع تنفيذ الحلقة المُشار إليها بِـ ***// .

في المتابعة عليك تبيين: j وَ arr[j] وَ c[arr[j]] وَ b[c[arr[j]] − 1] ،
والمصفوفة b والمصفوفة c .

ب. ماذا تنفّذ العمليّة what بلغة Java أو What بلغة C# ؟

جـ. ما هي تعقيدات زمن تشغيل العمليّة what بلغة Java أو What بلغة C# ؟ علّل إجابتك.

/ يتبع في صفحة 6 /

3. **أ.** طبِّق عمليّة خارجيّة exist بلغة Java أو Exist بلغة #C.
تتلقّى العمليّة شجرة بينارية t من نمط صحيح وعددًا صحيحًا x . تُعيد العمليّة true إذا كان في الشجرة عقدة قيمتها x ، خلاف ذلك – تُعيد العمليّة false . إذا كانت الشجرة فارغة – تُعيد العمليّة false .

ب. أمامك العمليّة بلغة Java check(t1, t2) وبلغة #C Check(t1, t2) .
تتلقّى العمليّة شجرتين بيناريّتين غير فارغتين من نمط صحيح، t1 وَ t2 ، وتُعيد قائمة تحوي جميع الأعداد الموجودة في الشجرة t1 وغير موجودة في الشجرة t2 . تستدعي العمليّة عمليّة إضافيّة تتلقّى ثلاثة پارامترات.

Java

```
public static Node<Integer> check(BinNode<Integer> t1, BinNode<Integer> t2)
{
    Node<Integer> first = new Node<Integer> (−1);
    first = check(t1 , t2 , first);
    return first.getNext();
}
```

C#

```
public static Node<int> Check(BinNode<int> t1 , BinNode<int> t2)
{
    Node<int> first = new Node<int> (−1);
    first = Check(t1 , t2 , first);
    return first.GetNext();
}
```

طبِّق العمليّة:
بلغة Java:

```
public static Node<Integer> check( BinNode<Integer> t1,
                                   BinNode<Integer> t2 , Node<Integer> list)
```

أو بلغة #C:

```
public static Node<int> Check(BinNode<int> t1 , BinNode<int> t2 , Node<int> list)
```

بإمكانك استعمال العمليّة التي طبّقتَها في البند "أ".

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

/ يتبع في صفحة 7 /

4. الصورة التي نراها على شاشة الحاسوب مركَّبة من نقاط ملوَّنة – بكسلات. كلّ بكسل مركَّب من ثلاثة مركِّبات لون: أحمر، أخضر، أزرق (Red Green Blue — RGB). كلّ مركِّب لون ممثَّل بواسطة عدد صحيح بين 0 وَ 255 ، والدمج بين ثلاثة المركِّبات يحدّد لون البكسل.
مثال: اللون الأبيض ممثَّل بواسطة الأعداد (255 , 255 , 255) ، اللون الأحمر ممثَّل بواسطة الأعداد (255 , 0 , 0) ، اللون الأزرق – (0 , 0 , 255) ، اللون الأسود – (0 , 0 , 0).

أ. اكتب بلغة Java أو بلغة C# عنوان وصفات الفئة **Pixel** التي تمثّل بكسلاً.
أمامك تخطيط جزئيّ لواجهة تطبيق الفئة **Pixel** مكتوب بلغة Java وبلغة C# .

عمليّة بنائيّة تُكوِّن اللون الأسود.	public Pixel()
عمليّة بنائيّة تتلقّى 3 أعداد صحيحة هي عبارة عن قيَم ثلاثة مركِّبات لون البكسل. إذا كان أحد (على الأقلّ) البارامترات غير موجود في مجال القيَم المسموح به (0-255 [بما في ذلك 0 وَ 255]) – يُبنى كائن لونه أسود.	public Pixel(int red , int green , int blue)
تتلقّى العمليّة كائنًا من نمط **Pixel** ، وتغيّر قيَم ألوان الكائن الحاليّ إلى اللون الذي ينتج من خلط اللونين (لون الكائن الحاليّ ولون الكائن الذي نتج كبارامتر). خلط الألوان الذي ينتج هو معدّل المركِّبين.	**Java**: public void mix(Pixel pxl) **C#**: public void Mix(Pixel pxl)

انتبه: عندما ينتج معدّل هو عبارة عن كسر (مثلاً 113.5) يُؤخذ فقط القسم الصحيح للقيمة الجديدة (في هذه الحالة – 113).
مثال: إذا كان لون الكائن الحاليّ هو أحمر (0 , 0 , 255) ولون الكائن الذي نتج كبارامتر هو أخضر (0 , 255 , 0) ، عندها اللون الذي ينتج من الخلط هو أصفر (0 , 127 , 127) ، وسيكون من الآن لون الكائن الحاليّ.

ب. طبِّق بلغة Java أو بلغة C# العمليّات التي تظهر في واجهة تطبيق الفئة **Pixel** .

جـ. (1) أضِف فئة باسم **Pic50** تمثّل صورة بِكبَر 50 × 50.
يوجد للفئة الصفة: pic – مصفوفة ثنائيّة الأبعاد بِكبَر 50 × 50 من نمط **Pixel** .
اكتب بلغة Java أو بلغة C# عنوان الفئة وصفتها.

(2) اكتب في الفئة **Pic50** عمليّة mix بلغة Java أو Mix بلغة C# ، تتلقّى صورة. تخلط العمليّة ألوان الصورة التي نتجت مع ألوان الصورة الحاليّة وتُعدِّل الصورة الحاليّة. يتمّ الخلط بين البكسلات التي مؤشّراتها متساوية.

/ يتبع في صفحة 8 /

الفصل الثاني (50 درجة)

في هذا الفصل أسئلة في أربعة مسارات:
أنظمة حاسوب وأسمبلي، ص 8-13.
مقدّمة لبحث الأداءات، ص 14-21.
موديلات حسابيّة، ص 22-24.
برمجة موجّهة كائنات بلغة Java، ص 25-32؛ برمجة موجّهة كائنات بلغة C# ، ص 33-41.

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. أمامك قطعة برنامج بلغة أسمبلي التي فيها يوجد في AL وفي AH أعداد تختلف عن بعضها البعض، بدون إشارة.

```
        MOV  AX , 7F80H
        CMP  AH , AL
        JL   SMALL
BIG:    MOV  BL , AH
        JMP  SOF
SMALL:  MOV  BL , AL
SOF:
```

(1) ماذا ستكون قيمة BL في نهاية تنفيذ قطعة البرنامج؟

(2) ماذا تنفّذ قطعة البرنامج؟

(3) في قطعة البرنامج، بدلاً من الأمر JL SMALL كُتب الأمر JB SMALL.
هل في نهاية تنفيذ قطعة البرنامج بعد الإبدال ستكون قيمة BL مختلفة عن قيمته في نهاية تنفيذ قطعة البرنامج قبل إبدال الأمر؟ فسّر إجابتك.

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 9 /

ب. (لا علاقة له مع البند "أ".)

أمامك ستّة أقوال. بالنسبة لكلّ واحد منها حدِّد إذا كان صحيحًا أم غير صحيح. إذا كان القول غير صحيح، فسّر لماذا.

(1) $10011110101110_2 > 27AE_{16}$

(2) $27AE_{16} > 10159_{10}$

(3) وظيفتا الخازنين BP وَ SP هما:

BP يحدّد في أيّ عنوان تُسجَّل كلمة في الراصّة، ومن أيّ عنوان في الراصّة تُنتزَع كلمة.

SP يُمكِّن البرنامج الفرعيّ من انتزاع پارامترات من الراصّة، بدون إشراك BP.

(4) في نهاية استدعاء وتنفيذ كلّ أمر، قيمة IP تكبر دائمًا بِـ 1.

(5) أمامك قطعتا برنامج بلغة أسمبلي.

القطعة 1	القطعة 2
MOV DX , 0000H	MOV DX , 0000H
MOV AX , 0064H	MOV AX , 0064H
DIV AL	DIV AX

في نهاية تنفيذ قطعتَي البرنامج محتوى AX سيكون 0001H.

(6) أمامك قطعتا برنامج بلغة أسمبلي.

القطعة 1	القطعة 2
MOV AL , 5BH	MOV AL , 5BH
MOV CL , 8	MOV CL , 9
ROL AL , CL	ROL AL , CL

في نهاية تنفيذ القطعة 1 ستكون قيمة AL مساوية للقيمة التي عُوِّضت فيه في الأمر الأوّل، وكذلك في نهاية تنفيذ القطعة 2 ستكون قيمة AL مساوية للقيمة التي عُوِّضت فيه في الأمر الأوّل.

/ يتبع في صفحة 10 /

6. أ. اكتب بلغة أسمبلي برنامجًا فرعيًّا باسم CALC يتلقّى كپارامتر عددًا بِكبَر كلمة.
العدد أصغر من 9_{10} وأكبر من 0 .
يحسب البرنامج الفرعيّ عامِليّ (עצרת) هذا العدد، ويخزن النتيجة ككلمة – مضاعفة مركَّبة من الخازنين المقترنين DX:AX .

ب. كُتب في مقطع المعطيات:

```
ARR   DW  3, 10, 0, 8
RES   DW  8  DUP (0)
```

- ARR هي مصفوفة لـ 4 أعداد كلّ واحد منها بِكبَر كلمة؛
- RES هي مصفوفة لِـ 4 النتائج التي يُعيدها البرنامج الفرعيّ CALC . كلّ نتيجة هي بِكبَر كلمة – مضاعفة.

(1) اكتب برنامجًا ينفّذ المهمّات التالية:

– يقرأ البرنامج 4 الأعداد من المصفوفة ARR .

– إذا كان العدد أصغر من 9_{10} وكذلك أكبر من 0 – يُرسله البرنامج كپارامتر إلى البرنامج الفرعيّ CALC الذي كتبتَه في البند "أ". يُسجِّل البرنامج النتيجة التي يُعيدها البرنامج الفرعيّ كعدد سداسيّ عشريّ (هكساديسيماليّ) بِكبَر كلمة – مضاعفة، حسب الترتيب، في مصفوفة النتائج RES .

– إذا لم يكن العدد في المجال المطلوب – يكتب البرنامج حسب الترتيب في مصفوفة النتائج RES الكلمة 0000H ، وبعدها – العدد كعدد سداسيّ عشريّ.

(2) انسخ الجدول الذي أمامك إلى دفترك، وأكمله بحيث تبدأ عناوين إزاحة خلايا المصفوفة RES بالإزاحة 0010H .

القيمة التي يُعيدها البرنامج الفرعيّ CALC	العناوين
	0010 -11
	0012 -13
	0014 -15
	0016 -17

/ يتبع في صفحة 11 /

7. معطاة المصفوفتان ARR1 وَ ARR2 . المصفوفتان هما بِكبَر متساوٍ وكلّ واحد من حدودهما هو بِكبَر كلمة. في كلّ حدّ، في كلّ واحدة من المصفوفتين، مخزون عدد سداسيّ عشريّ (هكساديسيماليّ) صحيح مكوّن من أربعة أرقام، في المجال 1000H - FFFFH .
كما ومعطى المتغيّر K بِكبَر بايت مخزون فيه كِبَر المصفوفتين ARR1 وَ ARR2 .

اكتب بلغة أسمبلي قطعة برنامج تخزن في الخازن DX عدد أزواج الأعداد (أحدهما من المصفوفة ARR1 والآخر من المصفوفة ARR2) التي تحقّق الشرطين التاليين:

(i) للعددين في الزوج يوجد نفس المؤشّر (الإندكس) في المصفوفة.

(ii) العددان في الزوج مكوّنان من نفس الأرقام لكن بترتيب عكسيّ.

مثال: بالنسبة للمصفوفتين ARR1 وَ ARR2 بِكبَر 5 اللتين أمامك، يُخزَن العدد 2 في الخازن DX.

ARR1	2025H	1061H	1492H	5777H	1948H
ARR2	1984H	1601H	2914H	9999H	8491H

/ يتبع في صفحة 12 /

8. أ. في مقطع المعطيات عُرِّفت المعطيات على النحو التالي:

```
ARR  DB  04H , 0FH ,  07H
```

انتبه: الأعداد هي سداسيّة عشريّة (هكساديسيماليّة).

أمامك برنامج بلغة أسمبلي.

```
BEGIN: MOV   SI , 0
       MOV   BP, 10H
       MOV   CX, 2

NEXT1: MOV   DI , BP
       XOR   AX , AX
       MOV   AL , [SI]
       MOV   [DI] , AL
       MOV   BX , AX

       MOV   DL , 2
       DIV   DL
       MOV   DL , AL

CHECK: MOV   AX , BX
       DIV   DL
       CMP   AH , 0
       JNZ   NEXT2

       INC   DI
       MOV   [DI] , DL
NEXT2: DEC   DL
       JNZ   CHECK

       INC   SI
       ADD   BP , 10H
       LOOP  NEXT1
SOF:
```

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 13 /

خارطة مقطع المعطيات:

0 1 2 3 4 5 6 7 8 9 A B C D E F

DS: 0000 04 0F
DS: 0010 __ __ __
DS: ____ __ __ __ __

(1) تتبّع بواسطة جدول متابعة تنفيذ البرنامج بالنسبة للمصفوفة ARR المعطاة، واكتب ماذا سيكون محتوى كلّ من:
SI , BP , CX , DI , AH , AL , BX , DL.

(2) انسخ إلى دفترك خارطة مقطع المعطيات، وأكملها بالنسبة لجميع المعطيات التي تنتج خلال تنفيذ البرنامج بالنسبة للمصفوفة ARR المعطاة.

(3) اكتب ماذا ينفّذ البرنامج.

ب. (1) محوا من البرنامج الأمر: XOR AX,AX .
هل ينفّذ البرنامج المطلوب؟ فسّر إجابتك.

(2) أبدلوا في البرنامج الأمر XOR AX,AX بالأمر XOR AL,AL .
هل ينفّذ البرنامج المطلوب؟ فسّر إجابتك.

(3) أبدلوا في البرنامج الأمر XOR AX,AX بالأمر XOR AH,AH .
هل ينفّذ البرنامج المطلوب؟ فسّر إجابتك.

/ يتبع في صفحة 14 /

مقدّمة لبحث الأداءات

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

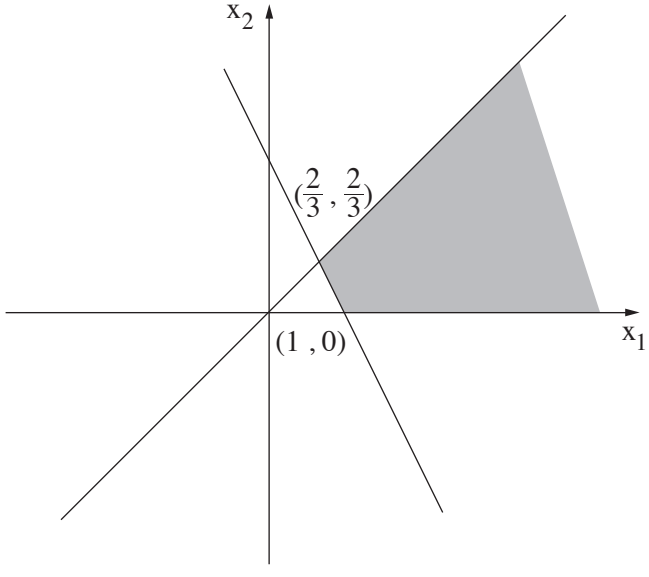
9. معطاة مسألة تخطيط خطّيّ:

$$\max\{z = 5x_1 - x_2\}$$

تخضع للاضطرارات التالية:

$$x_2 \le x_1$$
$$2x_1 + x_2 \ge 2$$
$$x_1 \ge 0$$
$$x_2 \ge 0$$

أمامك رسم لمجال الحلول الممكنة للمسألة المعطاة.



كلّ واحد من البنود "أ–هـ" التي في الصفحة التالية يتطرّق إلى مسألة التخطيط الخطّيّ المعطاة.
البنود "أ–هـ" لا تتعلّق ببعضها البعض. أجب عن جميع البنود.
بالنسبة للبند "أ–د" معطاة أربعة أقوال i-iv، ولكلّ بند فقط قول واحد صحيح.

i الحلّ الأمثل هو وحيد.

ii توجد لا نهاية من الحلول المثلى.

iii الحلّ الأمثل ليس محصوراً.

iv لا يوجد حلّ أمثل.

(انتبه: تكملة السؤال في الصفحة التالية.) / يتبع في صفحة 15 /

في كلّ واحد من البنود "أ–د"، حدِّد أيّ قول صحيح، انسخه إلى دفترك، وعلّل تحديدك.

- إذا اخترتَ القول i في أحد البنود، عليك إيجاد الحلّ الأمثل الوحيد وقيمة دالّة الهدف في هذا الحلّ.
- إذا اخترتَ القول ii في أحد البنود، عليك كتابة الحلّ الأمثل العامّ للمسألة وقيمة دالّة الهدف في مجال الحلول المثلى.

أ. أيّ قول هو الصحيح بالنسبة لمسألة التخطيط الخطّيّ المعطاة في بداية السؤال؟
علّل إجابتك.

ب. نغيّر فقط دالّة هدف المسألة المعطاة في بداية السؤال
إلى $\max\{z = -5x_1 + x_2\}$.
أيّ قول هو الصحيح بعد تغيير دالّة الهدف؟ علّل إجابتك.

جـ. نغيّر فقط دالّة هدف المسألة المعطاة في بداية السؤال
إلى $\min\{z = -5x_1 + x_2\}$.
أيّ قول هو الصحيح بعد تغيير دالّة الهدف؟ علّل إجابتك.

د. نغيّر فقط دالّة هدف المسألة المعطاة في بداية السؤال
إلى $\max\{z = -2x_1 + 2x_2\}$.
أيّ قول هو الصحيح بعد تغيير دالّة الهدف؟ علّل إجابتك.

هـ. نُضيف اضطرارًا إضافيًّا إلى المسألة المعطاة في بداية السؤال، وهو: $2x_1 - x_2 \le 2$
ونغيّر دالّة هدف المسألة المعطاة في بداية السؤال إلى $\max\{z = ax_1 + x_2\}$ عندما $(-1) < a < 2$، والقيمة القصوى التي تحصل عليها دالّة الهدف هي 4 في مجال الحلول الممكنة.
احسب قيمة a.

/ يتبع في صفحة 16 /

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

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

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

(1) ارسم الرسم البيانيّ G الممثَّل بواسطة المصفوفة.

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

(3) ما هو العدد الأقصى للأقواس التي يمكن إزالتها من الرسم البيانيّ المعطى حتّى يحوي الرسم البيانيّ نفس عدد مركِّبات الربط القويّ التي وجدتَها في البند السابق؟
ما هو القوس أو ما هي الأقواس التي يمكن إزالتها؟

ب. (لا علاقة له مع البند "أ".)

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

a → b → c → d → ||
b → a → c → ||
c → a → b → ||
d → a → f → e → ||
e → d → f → ||
f → e → d → ||

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

(2) هل الرسم البيانيّ المعطى G هو رسم بيانيّ قابل للربط؟ علّل.

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

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

/ يتبع في صفحة 17 /

11. 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: $x_{12} = 20$ ، $x_{11} = 100$.

المصادر	الغايات			العرض
	1	2	3	
1	10 (100)	15 (20)	17	120
2	10	18	14	100
3	15	20	18	100
الطلب	100	110	110	

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

ب. في الجدول الذي أمامك معطى جزء من حلّ أساسيّ ممكن لمسألة نقل، ومعطاة قيَم v_3 ، v_2 ، v_1 ، u_3 ، u_2 ، u_1.

المصادر	الغايات			العرض	u_i
	1	2	3		
1	2 (20)	5	7	20	2
2	1	1 (10)	4	10	0
3	0	1	8 (10)	15	0
الطلب	20	15	10		
v_j	0	1	8		

انسخ الجدول إلى دفترك، وأكمله مع الأخذ بالحسبان قيَم u_i وقيَم v_j بحيث ينتج حلّ أساسيّ ممكن.

(انتبه: تكملة السؤال في الصفحة التالية.) / يتبع في صفحة 18 /

جـ. في الجدول الذي أمامك معطى حلّ أساسيّ ممكن لمسألة نقل، ومعطاة قيَم v_3 ، v_2 ، v_1 ، u_3 ، u_2 ، u_1 .

المصادر	الغايات			العرض	u_i
	1	2	3		
1	14	15 (130)	17 (50)	180	0
2	10 (80)	8 (20)	14	100	-7
3	15	20	18 (80)	80	1
الطلب	80	150	130		
v_j	17	15	17		

هل الحلّ هو أمثل؟ علّل إجابتك.

د. في الجدول الذي أمامك معطى حلّ أساسيّ ممكن لمسألة نقل، ومعطى $v_1 = 0$.

المصادر	الغايات			العرض	u_i
	1	2	3		
1	10 (20)	25	30	20	
2	10 (30)	22	14 (20)	50	
3	18	20 (40)	20 (20)	60	
الطلب	50	40	40		
v_j	0				

انسخ الجدول إلى دفترك، وأكمل فيه قيَم u_1 وَ u_2 وَ u_3 وَ v_2 وَ v_3 .

(انتبه: تكملة السؤال في الصفحة التالية.) / يتبع في صفحة 19 /

هـ. في الجدول الذي أمامك معطى حلّ أمثل لمسألة نقل، ومعطاة قيَم v_3 ، v_2 ، v_1 ، u_3 ، u_2 ، u_1 التي تلائم هذا الحلّ.

u_i	العرض	الغايات			المصادر
		3	2	1	
20	40	26	35	20 (40)	1
10	90	14 (40)	22	10 (50)	2
26	100	30 (40)	30 (60)	28	3
		80	60	90	الطلب
		4	4	0	v_j

(1) هل الحلّ الأمثل هو وحيد؟

(2) نُنقص 4 من تكلفة الوحدة من المصدر 3 لكلّ غاية ونحصل على مسألة جديدة.
أمامك أربعة أقوال، واحد منها صحيح فقط. انسخ إلى دفترك القول الصحيح وعلّل اختيارك.

i الحلّ المعطى ليس حلّاً ممكنًا بالنسبة للمسألة الجديدة.

ii الحلّ المعطى هو حلّ أساسيّ ممكن، لكنّه ليس أمثل بالنسبة للمسألة الجديدة.

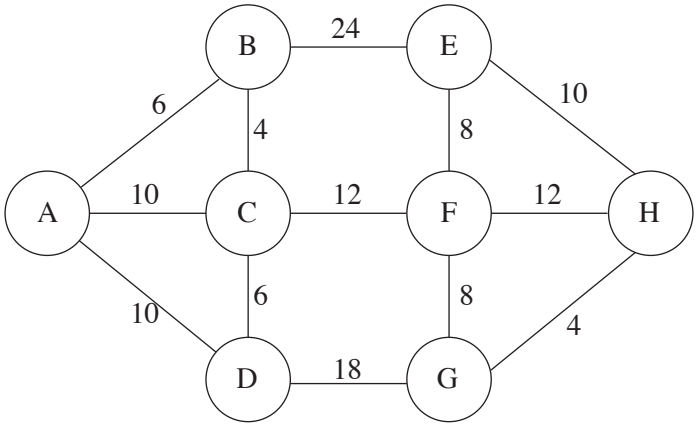
iii الحلّ المعطى هو حلّ أمثل وحيد بالنسبة للمسألة الجديدة.

iv الحلّ المعطى هو حلّ أمثل، لكنّه ليس حلّاً أمثل وحيدًا بالنسبة للمسألة الجديدة.

/ يتبع في صفحة 20 /

12. 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.



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

ب. (لا علاقة له مع البند "أ".)

$G = (V , E)$ هو رسم بيانيّ **موجَّه**. أمامك وصف جزئيّ لألغوريثم يفحص إذا كان الرسم البيانيّ المعطى يحوي دائرة.

مُخرَج الألغوريثم هو: **توجد دائرة** أو **لا توجد دائرة**.

الألغوريثم:

الخطوة 1: فعِّل ألغوريثم **[1]** على الرسم البيانيّ G.

الخطوة 2: إذا كان / كانت **[2]** يحوي / تحوي **[3]**، عندها

2.1 المُخرَج هو: في الرسم البيانيّ G لا توجد دائرة.

2.2 خلاف ذلك – المُخرَج هو: في الرسم البيانيّ G توجد دائرة.

في الألغوريثم هناك ثلاثة تعابير ناقصة أُشير إليها بِـ [1]، [2]، [3].

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 21 /

كلّ واحد من البنود الفرعيّة (1)-(3) التي أمامك يتطرّق إلى أحد التعابير الناقصة، ويحوي إجابات ممكنة بالنسبة له.

في كلّ بند فرعيّ معطاة أربع إجابات، i-iv ، واحدة منها فقط صحيحة. انسخ الألغوريثم إلى دفترك، وأكمله بواسطة الإجابات الصحيحة.

(1) الإجابة الصحيحة بالنسبة للتعبير [1] في الألغوريثم هي:

i ديكسترا

ii تمشيط عرضيّ (BFS)

iii تمشيط عميق (DFS)

iv إيجاد مركّبات الربط القويّ

(2) الإجابة الصحيحة بالنسبة للتعبير [2] في الألغوريثم هي:

i أقصر مسار

ii الشجرة الامتداديّة (BFS)

iii الشجرة الامتداديّة (DFS)

iv كلّ مركّب ربط قويّ

(3) الإجابة الصحيحة بالنسبة للتعبير [3] في الألغوريثم هي:

i رأسًا واحدًا فقط

ii رأسين على الأقلّ

iii جميع رؤوس الرسم البيانيّ

iv جميع أقواس الرسم البيانيّ

/ يتبع في صفحة 22 /

موديلات حسابيّة

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

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

أ. كلّ جزيء DNA مركَّب من 4 حجارة بناء أساسيّة يُشار إليها بالأحرف: A ، T ، C ، G.
يمكن تمثيل كلّ جزيء ككلمة مركَّبة من الأحرف A ، T ، C ، G.
في بحث جزيء DNA معيّن، وُجد أنّه إذا كانت الكلمة التي تمثّل الجزيء تبدأ بالتسلسل AAG ، ويظهر في الكلمة التسلسل AT مرّتين أو أكثر – فإنّ احتمال المرض X عالٍ.
مثال لجزيء DNA احتمال المرض فيه عالٍ: الكلمة التي تمثّل جزيء الـ DNA : **AAG**TGA**AT**G**AT**C**AT**GC.
ابنِ أوتوماتًا نهائيًّا محدودًا يتلقّى لغة جميع الكلمات التي تمثّل جزيئات DNA التي احتمال المرض X فيها عالٍ.

ب. (لا علاقة له مع البند "أ".)
اللغات L_2 ، L_3 ، L_6 ، معرَّفة فوق الأبجديّة $\{0,1,2,3,4,5,6,7,8,9\}$.
L_2 هي مجموعة الكلمات التي تمثّل الأعداد التي تقسم على 2 بدون باقٍ.
L_3 هي مجموعة الكلمات التي تمثّل الأعداد التي تقسم على 3 بدون باقٍ.
L_6 هي مجموعة الكلمات التي تمثّل الأعداد التي تقسم على 6 بدون باقٍ.
أمامك ثلاثة ادّعاءات i-iii. حدِّد بالنسبة لكلّ واحد من الادّعاءات 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 هذه الكلمة تدحضه.

i $L_6 \subset L_3$

ii $L_2 \cap L_3 = L_6$

iii $L_2 \cdot L_3 = L_6$

/ يتبع في صفحة 23 /

14. في كلّ واحد من البنود "أ – هـ" التي أمامك معرَّفة لغة فوق الأبجديّة {0,1,2,3,4,5,6,7,8,9}.
الكلمات في كلّ واحدة من اللغات تمثّل أعدادًا أكبر من 0 .
في كلّ واحد من هذه البنود – ابنِ أوتوماتًا نهائيًّا محدودًا يتلقّى اللغة المعرَّفة فيه.
كلّ واحد من الأوتوماتات يستقبل العدد، رقمًا بعد رقم، ابتداءً من رقم الآحاد.

أ. أعداد زوجيّة فقط.

ب. أعداد فرديّة أصغر من 1000.

جـ. أعداد تقسم على 5 بدون باقٍ، ورقم آلافها، إذا وُجد – هو زوجيّ.

د. أعداد تقسم على 1000 بدون باقٍ.

هـ. أعداد تقسم على 9 بدون باقٍ، وأصغر من 30.

/ يتبع في صفحة 24 /

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

أ. معطاة عمليّة مكتوبة بلغة Java وبلغة C# .

Java

```
boolean foo(String str) {
    int cntrA = 0;
    int cntrC = 0;
    for (int i= 0; i < str.length; i++) {
        if (str[i] == 'a') cntrA++;
        if (str[i] == 'c') cntrC++;
    }
    if ((cntrA % 2 == 0) &&
        (cntrC % 3 == 0))
        return true;
    return false;
}
```

C#

```
bool Foo(string str) {
    int cntrA = 0;
    int cntrC = 0;
    for (int i= 0; i < str.Length; i++) {
        if (str[i] == 'a') cntrA++;
        if (str[i] == 'c') cntrC++;
    }
    if ((cntrA % 2 == 0) &&
        (cntrC % 3 == 0))
        return true;
    return false;
}
```

(1) اكتب اللغة L فوق الأبجديّة {a , c} التي هي عبارة عن مجموعة كلّ الكلمات التي بالنسبة لها تُعيد العمليّة المعطاة true .

(2) ابنِ أوتوماتًا نهائيًّا محدودًا يتلقّى اللغة L .

ب. (لا علاقة له مع البند "أ" .)

ابنِ أوتوماتًا نهائيًّا غير محدود فوق الأبجديّة {a , b} يتلقّى كلّ الكلمات التي تحوي على الأقلّ ظهورًا واحدًا لأحد التسلسلات: bbb ، aaba ، ababa .

16. أمامك اللغة L فوق الأبجديّة {a , b , c} .

$$L = \{a^n b^m c^{n+m} \mid n , m > 0\}$$

أ. ابنِ أوتومات راصّة يتلقّى اللغة L .

ب. ابنِ آلة تيورينچ تتلقّى اللغة L .

/ يتبع في صفحة 25 /

برمجة موجّهة كائنات

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

17. أمامك الفئات A ، B ، C ، D ، E .

```
public class A {
  public  void doSomething(int x) {System.out.println("1");}
  public   void doSomethingElse(int x, int y) {System.out.println("2");}
  public  int calc(int x) {System.out.println("3"); return x + 1;}
}
public class B extends A {
  public void doSomethingElse(int x) {System.out.println("4");}
}
public class C extends A {
  public void doSomething(int x) {System.out.println("5");}
  public int otherCalc(int x) {System.out.println("6"); return x −1;}
}
public class D extends A {
  public void calc(int x, int y) {System.out.println(7 + y + x);}
}
public class E extends C {
  public int calc(int x) {System.out.println("8"); return x + 3;}
}
```

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 26 /

أ. بالنسبة لكلّ واحد من الأوامر التي أمامك، اكتب إذا كان سليمًا أم غير سليم.
إذا كان الأمر غير سليم – علّل لماذا. إذا كان الأمر سليمًا، اكتب ماذا يُطبَع.

```
A  x1 = new A();  x1.doSomethingElse(9,1);
A  x2 = new B();  x2.doSomething(9);
A  x3 = new B();  x3.doSomethingElse(9,1);
A  x4 = new B();  x4.doSomethingElse(9);
A  x5 = new C();  x5.doSomething(9);
A  x6  = new C();  System.out.println(x6.calc(9));
A  x7 = new D();   System.out.println(x7.calc(9));
A  x8 = new D();   System.out.println(x8.otherCalc(9));
A  x9 = new E();   x9.doSomething(9);
A  x10 = new E();   System.out.println(x10.calc(9));
A  x11 = new E();   System.out.println(x11.otherCalc(9));
```

ب. في القطعة التي أمامك معروضة أوامر فيها أجزاء ناقصة، أُشير إليها بالأرقام [1]-[5].

```
[1] x = new    [2]    ();
A y = new    [3]    ();
x.    [4]    (    [5]    );
y.doSomething(11);
```

(i) انسخ الأرقام [1]-[5] إلى دفترك، وبجانب كلّ واحد منها اكتب الناقص، كي ينتج المُخرَج:

```
99
5
```

(ii) هل يمكن كتابة الناقص في [3] بشكل يختلف عمّا كتبتَه، كي ينتج المُخرَج المطلوب؟
إذا كان ممكنًا – اكتب ذلك، إذا لم يكن ممكنًا – فسّر لماذا.

/ يتبع في صفحة 27 /

18. في هذا السؤال عليك تطبيق فئات. كلّ مرّة يُطلب فيها تطبيق فئات عليك أن تكتب عنوان الفئة وصفاتها، وتطبيق جميع العمليّات. افترض أنّ العمليّتين get وَ set معطاتان ولا حاجة لتطبيقهما.

في مطعم "ليأكل الجوعى" يستعملون برنامجًا حاسوبيًّا لإدارة وجبة الحلويّات.

الفئة **Dessert** تمثّل معطيات لوجبة الحلويّات. في الفئة ثلاث صفات:

name— نصّ، اسم وجبة الحلويّات.

cost — عدد صحيح، يمثّل سعر وجبة واحدة من هذه الحلويّات.

amount — عدد صحيح، يمثّل عدد المفردات من هذه الحلويّات.

في الفئة ثلاث عمليّات:

- عمليّة بنائيّة، تتلقّى ثلاثة پارامترات وتبتدئ الصفات بحسبها.
- عمليّة totalCost تحسب وتُعيد ثمن جميع مفردات الحلويّات.
  ثمن جميع مفردات الحلويّات هو: عدد المفردات × سعر وجبة الحلويّات الواحدة.
- عمليّة toString، تُعيد نصًّا مركّبًا من اسم وجبة الحلويّات وعدد المفردات الموجودة وثمن جميع مفردات الحلويّات.

أ. طبِّق بلغة Java الفئة **Dessert**.

الفئة **FrozenDessert** تَرِث الفئة **Dessert**، وتمثّل معطيات لوجبات الحلويّات المجمّدة.
للفئة صفة إضافيّة temperature — عدد صحيح، يمثّل درجة الحرارة التي يجب فيها حفظ وجبة الحلويّات.

في الفئة عمليّتان:

- عمليّة بنائيّة، تتلقّى ثلاثة پارامترات (الاسم، سعر وجبة الحلويّات، عدد المفردات)، وتبتدئ الصفات بحسبها. العمليّة تبتدئ الصفة temperature بـ (4−).
- العمليّة toString، تُعيد نصًّا مركّبًا من اسم وجبة الحلويّات وعدد المفردات وثمن جميع مفردات الحلويّات ودرجة الحرارة التي يجب فيها حفظ الحلويّات.

ب. طبِّق بلغة Java الفئة **FrozenDessert**.

(انتبه: تكملة السؤال في الصفحة التالية.) / يتبع في صفحة 28 /

الفئة **Cake** تَرِث الفئة **Dessert** ، وتمثّل معطيات الكعكة.
للفئة صفة إضافيّة numPieces – عدد صحيح، ويمثّل عدد قِطَع الكعكة.
في الفئة عمليّتان:

- عمليّة بنائيّة، تتلقّى أربعة پارامترات وتبتدئ الصفات بحسبها.
- عمليّة piecePrice، تحسب وتُعيد سعر القطعة الواحدة من الكعكة.

انتبه: الصفة cost تمثّل سعر الكعكة الكاملة.

جـ. طبِّق بلغة Java الفئة **Cake** .

د. طبِّق بلغة Java فئة إضافيّة باسم **IceCream** ، تَرِث الفئة **FrozenDessert** وتمثّل وجبة الحلويّات بوظة.
للفئة صفة إضافيّة: sprinkles – متغيّر بولياني قيمته true إذا أضافوا سكاكر إلى البوظة، خلاف ذلك – قيمته false .
في الفئة عمليّتان:

- عمليّة بنائيّة، تتلقّى أربعة پارامترات وتبتدئ الصفات بحسبها.
- عمليّة toString ، تُعيد نصًّا مركَّبًا من اسم وجبة الحلويّات وثمن جميع مفردات الحلويّات ودرجة الحرارة التي يجب فيها حفظ الحلويّات. إذا أضافوا سكاكر إلى الوجبة – يُضاف أيضًا النصّ "Sprinkles are free" .

هـ. اكتب بلغة Java عمليّة خارجيّة printDesserts تتلقّى كپارامتر مصفوفة من نمط **Dessert** تحوي داخلها وجبات حلويّات مختلفة. بالنسبة لكلّ واحدة من وجبات الحلويّات التي في المصفوفة، على العمليّة أن تطبع سعرها.
بالنسبة لحلويّات الكعكة يجب طباعة سعر الكعكة الكاملة.

/ يتبع في صفحة 29 /

19. أمامك واجهات التطبيق IOne وَ ITwo وَ IThree وَ IFour والفئات Alpha وَ Beta وَ Gamma وَ Program.

```
public interface IOne {
  public boolean firstA(Object x);
  public void firstB(int num);
}

public interface ITwo {
  public int second ();
}

public interface IThree {
  public int third();
}
public  interface IFour {
  public int fourth();
}

public class Alpha implements IOne {}
public class Beta implements ITwo, IFour {}
public class Gamma implements ITwo, IThree {}
public class Program  {
  public static void main(String[] args) {}
}
```

أ. اكتب أسماء العمليّات التي يجب تطبيقها في كلّ واحدة من الفئات Alpha وَ Beta وَ Gamma، وفسّر لماذا.

(انتبه: تكملة السؤال في الصفحة التالية.) / يتبع في صفحة 30 /

افترض أنّ العمليّات التي كتبتَها في البند "أ" قد طُبّقت في كلّ واحدة من الفئات.

ب. هل التعريف public class Omega implements IFour extends Beta { } سليم؟
إذا كان غير سليم – فسّر لماذا.

جـ. بالنسبة لكلّ واحدة من القِطَع i -iv التي أمامك، لو كتبناها في العمليّة main – حدِّد إذا كانت العمليّة main التي تنتج سليمة أم غير سليمة. علّل تحديداتك.

i `ITwo x1 = new ITwo();`

ii `Beta b = new Beta();`

iii `Alpha a = new Alpha();`
`IOne x2 = a;`

iv `Gamma c = new Gamma ();`
`IOne x3 = c ;`

د. بالنسبة لكلّ واحد من المتطلّبَيْن i-ii اللذين أمامك، حدِّد إذا كان يمكن تنفيذه بواسطة كتابة أمر أو أوامر في العمليّة main. إذا كان ممكنًا – اكتب الأمر الملائم أو الأوامر الملائمة. إذا لم يكن ممكنًا – فسّر لماذا.

i تفعيل العمليّة second على كائن من نمط Beta.

ii تحويل كائن من نمط Gamma بحيث يكون كائنًا من نمط Alpha .

/ يتبع في صفحة 31 /

20. في هذا السؤال أربعة بنود، "أ–د". لا توجد علاقة بين البنود. أجب عن جميع البنود.

أ. معطاة الفئة A والفئة B التي تَرِث من A .

الأمر A a1 = new B(); يمرّ بتجميع (مسح أوامر) ويعمل بصورة سليمة.

بالنسبة لكلّ واحدة من الجمل i-iii التي أمامك، حدِّد إذا كانت صحيحة أم غير صحيحة. فسّر تحديداتك.

i لا يمكن كتابة الأمر Object obj = a1; ، لأنّ هناك حاجة لتحويل واضح (casting).

ii لا يمكن كتابة الأمر A a2 = a1; ، لأنّ هناك حاجة لتحويل واضح (casting).

iii لا يمكن كتابة الأمر B b1 = a1; ، لأنّ هناك حاجة لتحويل واضح (casting).

ب. معطاة العمليّة main في الفئة Program :

```
public static void  main(String[] args) {
  C c = new A();
  B bl = (B) (new A());
  B  b2 = new D();
  A  a = new D();
}
```

حدِّد أيّة إمكانيّة من الإمكانيّات i-v التي أمامك تصف علاقة بين الفئات، حسب العمليّة main المعطاة.

i C تَرِث من A ، B تَرِث من A ، D تَرِث من A .

ii A تَرِث من C ، B تَرِث من A ، D تَرِث من A .

iii C تَرِث من A ، B تَرِث من A ، D تَرِث من B .

iv A تَرِث من C ، C تَرِث من B ، D تَرِث من A .

v A تَرِث من C ، B تَرِث من A ، D تَرِث من B ، C تَرِث من B .

(انتبه : تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 32 /

جـ. معطى تعريفان للفئتين School وَ City:

```
public class City{___(1)___int students;  }
public  class  School extends City {
  void poll() { System.out.println("There are "+ students + " in the school ") ; }
}
```

كي لا يقع خطأ في التجميع (مسح أوامر) يمكن إكمال جزء السطر المُشار إليه بـِ (1) بواسطة إحدى الإمكانيّات التي أمامك. اختر الإمكانيّة الملائمة وانسخها إلى دفترك.

- كتابة public أو private أو protected .
- كتابة public أو private فقط.
- كتابة public أو protected فقط.
- كتابة protected فقط.
- كتابة public فقط.
- كتابة private فقط.

د. أمامك تعريف للفئة Quest :

```
public class Quest {
   private int  num;
   private static int count = 0;

   public Quest()
   {
      count++;
      num = count;
   }
   public void printNow()
    {
      System.out.println (num + "" + count);
     }
}
```

حدِّد كم كائنًا يجب تكوينها من الفئة Quest ، وعلى أيّ كائن منها يجب تفعيل العمليّة printNow كي يكون المُخرَج 35 .

/ يتبع في صفحة 33 /

برمجة موجّهة كائنات

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

21. أمامك الفئات A ، B ، C ، D ، E .

```
public class A
 {
    public virtual void DoSomething(int x) {Console.WriteLine("1"); }
    public void DoSomethingElse(int x, int y) {Console.WriteLine("2"); }
    public virtual int Calc(int x) { Console.WriteLine("3"); return x + 1; }
 }
 public class B : A
 {
    public void DoSomethingElse(int x) {Console.WriteLine("4"); }
 }
 public class C : A
 {
    public override void DoSomething(int x) {Console.WriteLine("5"); }
    public int OtherCalc(int x) { Console.WriteLine("6"); return x −1; }
 }
 public class D : A
 {
    public void Calc(int x, int y) {Console.WriteLine(7 + y + x); }
 }
 public class E : C
 {
    public override int Calc(int x) {Console.WriteLine("8"); return x + 3; }
 }
```

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 34 /

أ. بالنسبة لكلّ واحد من الأوامر التي أمامك، اكتب إذا كان سليمًا أم غير سليم.
إذا كان الأمر غير سليم – علّل لماذا. إذا كان الأمر سليمًا اكتب ماذا يُطبَع.

```
A x1 = new A(); x1.DoSomethingElse(9, 1);
A x2 = new B(); x2.DoSomething(9);
A x3 = new B(); x3.DoSomethingElse(9, 1);
A x4 = new B(); x4.DoSomethingElse(9);
A x5 = new C(); x5.DoSomething(9);
A x6 = new C(); Console.WriteLine(x6.Calc(9));
A x7 = new D(); Console.WriteLine(x7.Calc(9));
A x8 = new D(); Console.WriteLine(x8.OtherCalc(9));
A x9 = new E(); x9.DoSomething(9);
A x10 = new E(); Console.WriteLine(x10.Calc(9));
A x11 = new E(); Console.WriteLine(x11.OtherCalc(9));
```

ب. في القطعة التي أمامك معروضة أوامر فيها أجزاء ناقصة، أُشير إليها بالأرقام [1]-[5].

```
[1] x = new    [2]    ();
A y = new    [3]    ();
x.    [4]    (    [5]    );
y.DoSomething(11);
```

(i) انسخ الأرقام [1]-[5] إلى دفترك، وبجانب كلّ واحد منها اكتب الناقص، كي ينتج المُخرَج:

```
99
5
```

(ii) هل يمكن كتابة الناقص في [3] بشكل يختلف عمّا كتبتَه، كي ينتج المُخرَج المطلوب؟
إذا كان ممكنًا – اكتب ذلك، إذا لم يكن ممكنًا – فسّر لماذا.

/ يتبع في صفحة 35 /

22. في هذا السؤال عليك تطبيق فئات. كلّ مرّة يُطلب فيها تطبيق فئات عليك أن تكتب عنوان الفئة وصفاتها، وتطبيق جميع العمليّات. افترض أنّ العمليّتين Get وَ Set معطاتان ولا حاجة لتطبيقهما.

في مطعم "ليأكل الجوعى" يستعملون برنامجًا حاسوبيًّا لإدارة وجبة الحلويّات.

الفئة **Dessert** تمثّل معطيات لوجبة الحلويّات. في الفئة ثلاث صفات:

name— نصّ، اسم وجبة الحلويّات.

cost — عدد صحيح، يمثّل سعر وجبة واحدة من هذه الحلويّات.

amount — عدد صحيح، يمثّل عدد المفردات من هذه الحلويّات.

في الفئة ثلاث عمليّات:

- عمليّة بنائيّة، تتلقّى ثلاثة بارامترات وتبتدئ الصفات بحسبها.
- عمليّة TotalCost تحسب وتُعيد ثمن جميع مفردات الحلويّات.
  ثمن جميع مفردات الحلويّات هو: عدد المفردات × سعر وجبة الحلويّات الواحدة.
- عمليّة ToString، تُعيد نصًّا مركّبًا من اسم وجبة الحلويّات وعدد المفردات الموجودة وثمن جميع مفردات الحلويّات.

أ. طبِّق بلغة C# الفئة **Dessert**.

الفئة **FrozenDessert** تَرِث الفئة **Dessert**، وتمثّل معطيات لوجبات الحلويّات المجمّدة.

للفئة صفة إضافيّة temperature — عدد صحيح، يمثّل درجة الحرارة التي يجب فيها حفظ وجبة الحلويّات.

في الفئة عمليّتان:

- عمليّة بنائيّة، تتلقّى ثلاثة بارامترات (الاسم، سعر وجبة الحلويّات، عدد المفردات)، وتبتدئ الصفات بحسبها. العمليّة تبتدئ الصفة temperature بِـ (4−).
- العمليّة ToString، تُعيد نصًّا مركّبًا من اسم وجبة الحلويّات وعدد المفردات وثمن جميع مفردات الحلويّات ودرجة الحرارة التي يجب فيها حفظ الحلويّات.

ب. طبِّق بلغة C# الفئة **FrozenDessert**.

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

الفئة **Cake** تَرِث الفئة **Dessert** ، وتمثّل معطيات الكعكة.
للفئة صفة إضافيّة numPieces – عدد صحيح، ويمثّل عدد قِطَع الكعكة.
في الفئة عمليّتان:

- عمليّة بنائيّة، تتلقّى أربعة پارامترات وتبتدئ الصفات بحسبها.
- عمليّة PiecePrice، تحسب وتُعيد سعر القطعة الواحدة من الكعكة.

انتبه: الصفة cost تمثّل سعر الكعكة الكاملة.

جـ. طبِّق بلغة #C الفئة **Cake** .

د. طبِّق بلغة #C فئة إضافيّة باسم **IceCream** ، تَرِث الفئة **FrozenDessert** وتمثّل وجبة الحلويّات بوظة.
للفئة صفة إضافيّة: sprinkles – متغيّر بوليانيّ قيمته true إذا أضافوا سكاكر إلى البوظة، خلاف ذلك – قيمته false .
في الفئة عمليّتان:

- عمليّة بنائيّة، تتلقّى أربعة پارامترات وتبتدئ الصفات بحسبها.
- عمليّة ToString ، تُعيد نصًّا مركَّبًا من اسم وجبة الحلويّات وثمن جميع مفردات الحلويّات ودرجة الحرارة التي يجب فيها حفظ الحلويّات. إذا أضافوا سكاكر إلى الوجبة – يُضاف أيضًا النصّ "Sprinkles are free" .

هـ. اكتب بلغة #C عمليّة خارجيّة PrintDesserts تتلقّى كپارامتر مصفوفة من نمط **Dessert** تحوي داخلها وجبات حلويّات مختلفة. بالنسبة لكلّ واحدة من وجبات الحلويّات التي في المصفوفة، على العمليّة أن تطبع سعرها.
بالنسبة لحلويّات الكعكة يجب طباعة سعر الكعكة الكاملة.

/ يتبع في صفحة 37 /

23. أمامك واجهات التطبيق IOne وَ ITwo وَ IThree وَ IFour والفئات Alpha وَ Beta وَ Gamma وَ Program.

```
interface IOne {
  bool FirstA(Object x);
  void FirstB(int num);
}

interface ITwo {
  int Second();
}

interface IThree {
  int Third();
}
interface IFour{
  int Forth();
}

public class Alpha : IOne {}
public class Beta : ITwo, IFour {}
public class Gamma : ITwo, IThree {}
public class Program  {
  public static void Main() {}
}
```

أ. اكتب أسماء العمليّات التي يجب تطبيقها في كلّ واحدة من الفئات Alpha وَ Beta وَ Gamma، وفسّر لماذا.

(انتبه: تكملة السؤال في الصفحة التالية.) / يتبع في صفحة 38 /

افترض أنّ العمليّات التي كتبتَها في البند "أ" قد طُبّقت في كلّ واحدة من الفئات.

ب. هل التعريف public class Omega : IFour, Beta { } سليم؟

إذا كان غير سليم – فسّر لماذا.

جـ. بالنسبة لكلّ واحدة من القِطَع i -iv التي أمامك، لو كتبناها في العمليّة Main – حدِّد إذا كانت العمليّة Main التي تنتج سليمة أم غير سليمة. علّل تحديداتك.

i ITwo x1 = new ITwo();

ii Beta b = new Beta();

iii Alpha a = new Alpha();

IOne x2 = a;

iv Gamma c = new Gamma ();

IOne x3 = c ;

د. بالنسبة لكلّ واحد من المتطلّبَيْن i-ii اللذين أمامك، حدِّد إذا كان يمكن تنفيذه بواسطة كتابة أمر أو أوامر في العمليّة Main. إذا كان ممكنًا – اكتب الأمر الملائم أو الأوامر الملائمة. إذا لم يكن ممكنًا – فسّر لماذا.

i تفعيل العمليّة Second على كائن من نمط Beta.

ii تحويل كائن من نمط Gamma بحيث يكون كائنًا من نمط Alpha.

/ يتبع في صفحة 39 /

24. في هذا السؤال أربعة بنود، "أ–د". لا توجد علاقة بين البنود. أجب عن جميع البنود.

أ. معطاة الفئة A والفئة B التي تَرِث من A .

الأمر A a1 = new B(); يمرّ بتجميع (مسح أوامر) ويعمل بصورة سليمة.

بالنسبة لكلّ واحدة من الجمل i-iii التي أمامك، حدِّد إذا كانت صحيحة أم غير صحيحة.

فسّر تحديداتك.

i لا يمكن كتابة الأمر Object obj = a1; ، لأنّ هناك حاجة لتحويل واضح (casting).

ii لا يمكن كتابة الأمر A a2 = a1; ، لأنّ هناك حاجة لتحويل واضح (casting).

iii لا يمكن كتابة الأمر B b1 = a1; ، لأنّ هناك حاجة لتحويل واضح (casting).

ب. معطاة العمليّة Main في الفئة Program :

```
public static void  Main(string[] args) {
  C c = new A();
  B bl = (B) (new A());
  B  b2 = new D();
  A  a = new D();
}
```

حدِّد أيّة إمكانيّة من الإمكانيّات i-v التي أمامك تصف علاقة بين الفئات، حسب العمليّة Main المعطاة.

i C تَرِث من A ، B تَرِث من A ، D تَرِث من A .

ii A تَرِث من C ، B تَرِث من A ، D تَرِث من A .

iii C تَرِث من A ، B تَرِث من A ، D تَرِث من B .

iv A تَرِث من C ، C تَرِث من B ، D تَرِث من A .

v A تَرِث من C ، B تَرِث من A ، D تَرِث من B ، C تَرِث من B .

(انتبه : تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 40 /

جـ. معطى تعريفان للفئتين School وَ City:

```
public class City{____(1)____int students;  }
public  class  School : City {
  void Poll() { Console.WriteLine("There are "+ students + " in the school ") ; }
}
```

كي لا يقع خطأ في التجميع (مسح أوامر) يمكن إكمال جزء السطر المُشار إليه بِـ (1) بواسطة إحدى الإمكانيّات التي أمامك. اختر الإمكانيّة الملائمة وانسخها إلى دفترك.

- كتابة public أو private أو protected .
- كتابة public أو private فقط.
- كتابة public أو protected فقط.
- كتابة protected فقط.
- كتابة public فقط.
- كتابة private فقط.

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 41 /

د. أمامك تعريف للفئة Quest :

```
public class Quest  {
    private int num;
    private static int count = 0;

    public Quest()
    {
        count++;
        num = count;
    }
    public void PrintNow()
    {
        Console.WriteLine(num+""+count);
    }
}
```

حدِّد كم كائنًا يجب تكوينها من الفئة Quest ، وعلى أيّ كائن منها يجب تفعيل العمليّة PrintNow كي يكون المُخرَج 35 .

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.