מדינת ישראל
משרד החינוך

סוג הבחינה: בגרות
מועד הבחינה: קיץ תשע"ט, 2019
מספר השאלון: **899381**
תרגום לערבית (2)

مدعي المحسب
הוראות לנבחן

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח ההערכה:
בשאלון זה שלושה פרקים.
פרק ראשון
(10×1)+(15×1) — 25 נק'
פרק שני
(25×2) — 50 נק'
פרק שלישי
(25×1) — 25 נק'
סה"כ — 100 נק'

ג. חומר עזר מותר בשימוש: כל חומר עזר, חוץ ממחשב הניתן לתכנות.

ד. הוראות מיוחדות:
1. את כל התוכניות שאתה נדרש לכתוב בשפת מחשב בפרקים הראשון והשני, כתוב בשפה אחת בלבד — Java או C#.
2. **רשום על הכריכה החיצונית** של המחברת באיזו **שפה** אתה כותב — Java או C#.
3. **רשום על הכריכה החיצונית** של המחברת את שם ה**מסלול שלמדת**. המסלול הוא אחד מארבעת המסלולים האלה: מערכות מחשב ואסמבלי, מבוא לחקר ביצועים, מודלים חישוביים, תכנות מונחה עצמים.

הערה: בתוכניות 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 2019
رقم النّموذج: **899381**
ترجمة إلى العربيّة (2)

علم الحاسوب
تعليمات للممتحَن

أ. مدّة الامتحان: ثلاث ساعات.

ب. مبنى النّموذج وتوزيع الدّرجات:
في هذا النّموذج ثلاثة فصول.
الفصل الأوّل
(10×1)+(15×1) – 25 درجة
الفصل الثّاني
(25×2) – 50 درجة
الفصل الثّالث
(25×1) – 25 درجة
المجموع – 100 درجة

ج. موادّ مساعدة يُسمح استعمالها: كلّ مادّة مساعدة، عدا الحاسوب الذي يمكن برمجته.

د. تعليمات خاصّة:
1. اكتب جميع البرامج التي يُطلب منك كتابتها بلغة حاسوب في الفصلين الأوّل والثاني، بلغة واحدة فقط – Java أو C#.
2. **سجّل على الغلاف الخارجيّ** للدّفتر بأيّة **لغة** تكتب – Java أو C#.
3. **سجّل على الغلاف الخارجيّ** للدّفتر، اسم **المسار الذي تعلّمتَه**. المسار هو أحد المسارات الأربعة التّالية: أنظمة حاسوب وأسمبلي، مقدّمة لبحث الأداءات، موديلات حسابيّة، برمجة موجّهة كائنات.

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

اكتب في دفتر الامتحان فقط. اكتب "مسوّدة" في بداية كلّ صفحة تستعملها مسوّدة.
كتابة أيّة مسوّدة على أوراق خارج دفتر الامتحان قد تسبّب إلغاء الامتحان.

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

ב ה צ ל ח ה ! **نتمنّى لك النّجاح!**

الأسئلة

في هذا النموذج ثلاثة فصول.

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

الفصل الأوّل (25 درجة)

ملاحظة: في كلّ سؤال يُطلب فيه استقبال، لا حاجة لفحص سلامة المدخَلات.
للّذين يحلّون بلغة Java: في كلّ سؤال يُطلب فيه استقبال، افترِضْ أنّ الأمر التالي مكتوب في البرنامج:

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

أجب عن السؤال 1 – إلزاميّ (10 درجات).

1. اكتب عمليّة خارجيّة exact بلغة Java أو Exact بلغة C# ، تتلقّى مصفوفة – arr من نمط نصّ، وعددًا صحيحًا – num. تُعيد العمليّة عدد النصوص في المصفوفة arr ، التي طولها يساوي num.

أجب عن أحد السؤالين 2-3 (15 درجة).

2. في منظومة حاسوبيّة لمحلّ لمنتَجات التخييم توجد فئة باسم **Flashlight** لها صفتان:
model – اسم طراز مصباح من نمط نصّ.
price – سعر المصباح، عدد من نمط حقيقيّ.

أ. اكتب بلغة Java أو بلغة C# عنوان الفئة **Flashlight** وصفتَيْها، واكتب عمليّة بنائيّة تتلقّى قِيَمًا لطراز المصباح ولسعر المصباح.

ب. (1) استعدادًا للرحلة عليك التجهُّز بثلاثة مصابيح مختلفة. اكتب عمليّة خارجيّة threeFlashlights بلغة Java أو ThreeFlashlights بلغة C# ، تتلقّى مصفوفة – arr لكائنات مختلفة من نمط **Flashlight** ، وعددًا – total من نمط حقيقيّ.
على العمليّة أن تجد وتطبع مرّة واحدة فقط طرازات المصابيح الثلاثة التي مجموع أسعارها هو العدد total.
افترِضْ أنّه تمّ لكلّ صفة في الفئة **Flashlight** تعريف عمليّتَي get وَ set بلغة Java وعمليّتَي Get وَ Set بلغة C# .
افترِضْ أنّه توجد ثلاثيّة واحدة فقط لمصابيح مجموع أسعارها هو العدد total .

(2) ما هي تعقيدات زمن تشغيل العمليّة التي كتبتَها في البند الفرعيّ "ب (1)"؟ علّل.

/ يتبع في صفحة 3 /

3. في منظومة حاسوبيّة لشركة لبيع السيّارات المستعمَلة، عُرِّفت فئة باسم **Car** لها ثلاث صفات:

licenseNum – رقم لوحة ترخيص من نمط نصّ.

hadAccident – متغيّر من نمط بوليانيّ يتلقّى true إذا كانت السيّارة قد مرّت بحادثة طرق، خلاف ذلك – يتلقّى false .

price – سعر السيّارة، وهو عدد من نمط صحيح.

افترِضْ أنّه تمّ للفئة تعريف عمليّة بنائيّة تتلقّى قِيَمًا لكلّ واحدة من صفات الفئة، وتمّ لكلّ صفة تعريف عمليّتَي get وَ set بلغة Java وعمليّتَي Get وَ Set بلغة C#.

أ. اكتب في الفئة **Car** عمليّة داخليّة بوليانيّة باسم range بلغة Java أو Range بلغة C# ، تتلقّى عددين من نمط صحيح min وَ max . تُعيد العمليّة true إذا كان سعر السيّارة بين min وَ max (بما في ذلك min وَ max)، خلاف ذلك – تُعيد العمليّة false .

افترِضْ أنّ max أكبر من min .

ب. معطاة الفئة **AllCars** التي لها صفتان:

cars – مصفوفة أحاديّة الأبعاد من نمط **Car**.

num – عدد السيّارات الحاليّ الموجودة في الشركة، من نمط صحيح.

أمامك العمليّة البنائيّة للفئة:

```
public AllCars (int max)
{
   this.cars = new Car [max];
   this.num = 0;
}
```

(1) اكتب في الفئة **AllCars** عمليّة داخليّة بوليانيّة باسم addCar بلغة Java أو AddCar بلغة C#، تتلقّى كائنًا من نمط **Car** . تُضيف العمليّة الكائن إلى مصفوفة السيّارات في المكان الأوّل الخالي وتُعيد true. إذا كانت المصفوفة مليئة، تُعيد العمليّة false .

افترِضْ أنّه لا توجد أماكن خالية من بداية المصفوفة وحتّى العدد num (لا يشمل num).

(2) اكتب في الفئة **AllCars** عمليّة داخليّة باسم print بلغة Java أو Print بلغة C# ، تتلقّى عددين min وَ max من نمط صحيح (max أكبر من min).

تطبع العمليّة رقم لوحة الترخيص لكلّ سيّارة في المصفوفة cars ، التي لم تمرّ بحادثة طرق وسعرها بين العددين min وَ max (بما في ذلك min وَ max).

يجب عليك استعمال العمليّة التي عُرِّفت في البند "أ".

/ يتبع في صفحة 4 /

الفصل الثاني (50 درجة)

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

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

4. **انتبه:** لهذا السؤال صيغتان: بلغة Java (في الصفحتين 4-5) وبلغة C# (في الصفحتين 6-7). حلّ حسب ما تعلّمتَه.

للّذين يحلّون بلغة Java

أمامك العمليّة secret1:

```
public static boolean secret1 (int num, int digit )
    {
       if (num < 10)
            return ( num % 2 == digit % 2 );
       if ( num % 2 != digit % 2 )
            return false;
       return secret1 ( num / 10,  digit) ;
    }
```

أ. **(1)** اكتب ماذا يُعيد استدعاء العمليّة secret1 (937, 5) . عليك أن تُبيِّن متابعة.

(2) أعطِ مثالاً لعدد num مكوَّن من 3 أرقام، يُعيد بالنسبة له استدعاء العمليّة secret1(num,5) قيمة تختلف عمّا نتج في البند الفرعيّ "أ (1)". عليك أن تُبيِّن متابعة.

(3) اكتب بجملة واحدة ماذا تُنفِّذ العمليّة البوليانيّة secret1 ، أي ما هو السؤال الذي تُعيد العمليّة بالنسبة له true أو false.

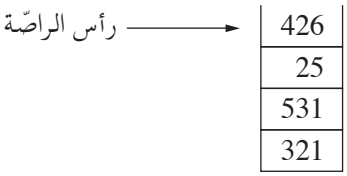
(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 5 /

أمامك العمليّة secret2 :

```
public static boolean secret2 ( Stack <Integer> s )
    {
        boolean ok;
        int x;
        if ( s.isEmpty () )
                ok = true ;
        else
        {
                 x = s.pop () ;
                 if ( ! (secret1 (x, x % 10)) )
                        ok = false;
                 else
                        ok = secret2 (s);
        }
        return ok ;
    }
```

ب. (1) بالنسبة للراصّة s التي أمامك:



اكتب ماذا تُعيد العمليّة secret2 . عليك أن تُبيِّن متابعة (لا حاجة لتبيين متابعة العمليّة secret1) .

(2) اكتب بجملة واحدة ماذا تُنفِّذ العمليّة البوليانيّة secret2 ، أي ما هو السؤال الذي تُعيد العمليّة بالنسبة له true أو false .

/ يتبع في صفحة 6 /

للّذين يحلّون بلغة C#

أمامك العمليّة Secret1

```
public static bool Secret1 (int num, int digit )
    {
        if ( num < 10 )
            return ( num % 2 == digit % 2 );
        if ( num % 2 != digit % 2 )
            return false;
        return Secret1 ( num / 10, digit) ;
    }
```

أ. **(1)** اكتب ماذا يُعيد استدعاء العمليّة Secret1 (937,5) . عليك أن تُبيِّن متابعة.

(2) أعطِ مثالاً لعدد num مكوَّن من 3 أرقام، يُعيد بالنسبة له استدعاء العمليّة Secret1 (num,5) قيمة تختلف عمّا نتج في البند الفرعيّ "أ (1)". عليك أن تُبيِّن متابعة.

(3) اكتب بجملة واحدة ماذا تُنفِّذ العمليّة البوليانيّة Secret1 ، أي ما هو السؤال الذي تُعيد العمليّة بالنسبة له true أو false .

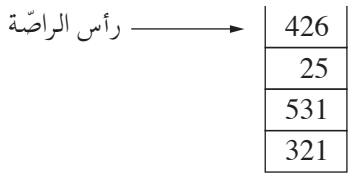
(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 7 /

أمامك العمليّة Secret2:

```
public static bool Secret2 ( Stack <int> s )
    {
        bool ok;
        int x;
        if ( s.IsEmpty () )
                ok = true ;
        else
        {
                x = s.Pop () ;
                if ( ! (Secret1 (x, x % 10)) )
                        ok = false;
                else
                        ok = Secret2 (s);
        }
        return ok ;
    }
```

ب. (1) بالنسبة للراصّة s التي أمامك:



اكتب ماذا تُعيد العمليّة Secret2 . عليك أن تُبيِّن متابعة (لا حاجة لتبيين متابعة العمليّة Secret1) .

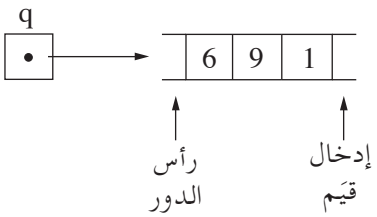
(2) اكتب بجملة واحدة ماذا تُنفِّذ العمليّة البوليانيّة Secret2 ، أي ما هو السؤال الذي تُعيد العمليّة بالنسبة له true أو false .

/ يتبع في صفحة 8 /

5. أ. "دور عدد" هو دور لأرقام بين 1 وَ 9 (بما في ذلك 1 وَ 9)، يمثّل عددًا صحيحًا – الحدّ الأوّل (رأس الدور) هو رقم الآحاد، والحدّ الثاني هو رقم العشرات وهكذا.

افترِضْ أنّ عدد الأرقام الممكن في الدور ليس أكبر من عدد الأرقام التي يستطيع أن يحويها متغيّر من نمط int .

مثال: الدور الذي أمامك يمثّل العدد 196.

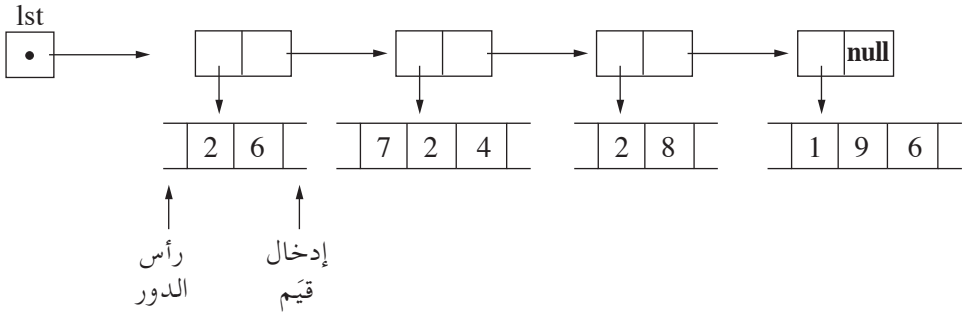


اكتب عمليّة باسم toNumber بلغة Java أو ToNumber بلغة C# ، تتلقّى "دور عدد" – q ، وتُعيد العدد المُمثَّل في الدور.

ملاحظة: المحافظة على مبنى الدور ليس إجباريًّا.

ب. معطاة سلسلة حلقات، كلّ حلقة فيها تحوي "دور عدد".

مثال: سلسلة الحلقات التي أمامك تمثّل الأعداد 691 , 82 , 427 , 62 .



اكتب عمليّة باسم bigNumber بلغة Java أو BigNumber بلغة C# ، تتلقّى توجيهًا lst إلى سلسلة الحلقات، وتُعيد العدد الأكبر المُمثَّل في سلسلة الحلقات.

بالنسبة لسلسلة الحلقات التي وُصفت في المثال، تُعيد العمليّة العدد 691 .

يجب عليك استعمال العمليّة التي عُرِّفت في البند "أ".

/ يتبع في صفحة 9 /

6. معطاة الفئة **Range** التي لها صفتان:

low – عدد من نمط صحيح.

high – عدد من نمط صحيح.

high أكبر من low .

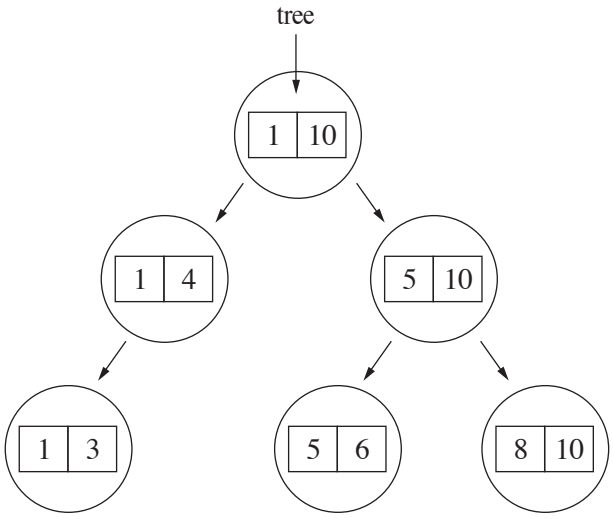
افترِضْ أنّه تمّ لكلّ صفة تعريف عمليّتَي get وَ set بلغة Java وعمليّتَي Get وَ Set بلغة C#.

شجرة مجالات هي شجرة حدودها من نمط **Range** .

شجرة مجالات مرتَّبة هي شجرة فارغة أو شجرة مجالات تَتَحَقَّق لكلّ عقدة فيها الشروط التالية:

- إذا تواجد ابن أيسر، فإنّ low العقدة يساوي low الابن الأيسر، وَ high العقدة أكبر أو يساوي high الابن الأيسر.
- إذا تواجد ابن أيمن، فإنّ high العقدة يساوي high الابن الأيمن، وَ low العقدة أصغر أو يساوي low الابن الأيمن.
- إذا تواجد ابنان، فإنّ high الابن الأيسر أصغر من low الابن الأيمن.

مثال **لشجرة مجالات مرتَّبة**:



اكتب عمليّة خارجيّة بوليانيّة باسم order بلغة Java أو Order بلغة C# ، تتلقّى **شجرة مجالات** أو **شجرة فارغة** وتُعيد true إذا كانت الشجرة **شجرة مجالات مرتَّبة**، خلاف ذلك – تُعيد العمليّة false .

/ يتبع في صفحة 10 /

الفصل الثالث (25 درجة)

في هذا الفصل أسئلة في أربعة مسارات:
أنظمة حاسوب وأسمبلي، في الصفحتين 10-11.
مقدّمة لبحث الأداءات، في الصفحات 12-15.
موديلات حسابيّة، في الصفحة 16.
برمجة موجّهة كائنات بلغة Java، في الصفحات 17-20؛ برمجة موجّهة كائنات بلغة C#، في الصفحات 21-24.

أجب عن سؤال واحد في المسار الذي تعلّمتَه.

أنظمة حاسوب وأسمبلي

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

7. في هذا السؤال أربعة بنود "أ – د". لا توجد علاقة بين البنود. عليك الإجابة عن جميعها.

أ. أمامك قطعة برنامج بلغة أسمبلي.

```
        MOV BX, 100H
        MOV SI, 2
        MOV CX, 3
L1:
        MOV AL, CL
        MOV [BX], AL
        INC BX
        LOOP L1
L2:
        DEC AL
        MOV [BX], AL
        INC BX
        DEC SI
        JNZ L2
        NOP
```

تتبّع بواسطة جدول متابعة تنفيذ قطعة البرنامج. يجب أن تشمل في جدول المتابعة عمودًا لكلّ واحد من الخوازن المعروضة في القطعة. بالإضافة إلى ذلك يجب أن ترسم خارطة ذاكرة ملائمة.

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 11 /

ب. أمامك قطعة برنامج مكتوبة بلغة Java وبلغة #C . اكتب قطعة ملائمة بلغة أسمبلي.

```
if (a > 0 || b > 0) c++;
```

افترِضْ أنّ المتغيّرات a ، b ، c مخزونة في الخوازن AX وَ BX وَ CX بالتلاؤم.
افترِضْ أنّ الأعداد موجَّهة (Signed) .

جـ. معطاة القطعتان – القطعة 1 والقطعة 2 .
هل في نهاية تشغيل كلّ واحدة من القطعتين تكون القيَم في الخازن CX متساوية؟ علّل.

القطعة 1	القطعة 2
MOV AX, 10	MOV CX, 10
MOV CX, AX	SHL CX, 1
MUL CX	MOV DX, CX
MOV CX, AX	SHL CX, 2
	ADD CX, DX

د. في الخازن AX يوجد عدد ما. اكتب أوامر تُبَدِّل بين البايْت السفليّ وبين البايْت العلويّ للخازن.
مثال: إذا كانت القيمة الابتدائيّة لـ AX هي 5A9Ch ، تكون قيمته بعد التبديل 9C5Ah .
لا تَستعْمِل الأمرين ROR , ROL .

8. "منوال الأعداد" في المصفوفة هو العدد الذي يظهر أكبر عدد من المرّات في المصفوفة.
مثلاً في المصفوفة التي أمامك:

index:	0	1	2	3	4	5	6	7	8	9
valve	−3	6	0	11	8	−9	5	6	33	6

منوال الأعداد هو 6 ، لأنّ عدد المرّات التي يظهر فيها في المصفوفة هو 3 ، ولا يوجد عدد آخر يظهر في المصفوفة أكثر من 3 مرّات.

أ. معطاة المصفوفة ARR بِكبَر 100 تحوي أعدادًا من نمط بايْت.
اكتب برنامجًا فرعيًّا يتلقّى عددًا صحيحًا عن طريق الراصّة، ويخزن في الخازن AL عدد المرّات التي يظهر فيها العدد في المصفوفة ARR.

ب. اكتب برنامجًا يجد منوال الأعداد في المصفوفة ARR ، ويخزنه في المتغيّر VAL .
يجب عليك استعمال البرنامج الفرعيّ الذي كتبتَه في البند "أ".
افترِضْ أنّه يوجد في المصفوفة منوال أعداد وحيد.

/ يتبع في صفحة 12 /

مقدّمة لبحث الأداءات

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

9. معطاة مسألة تخطيط خطّيّ :

$$\min\{z = 3x_1 + 5x_2\}$$

تخضع للاضطرارات التالية :

(1) $x_1 + x_2 \geq 5$

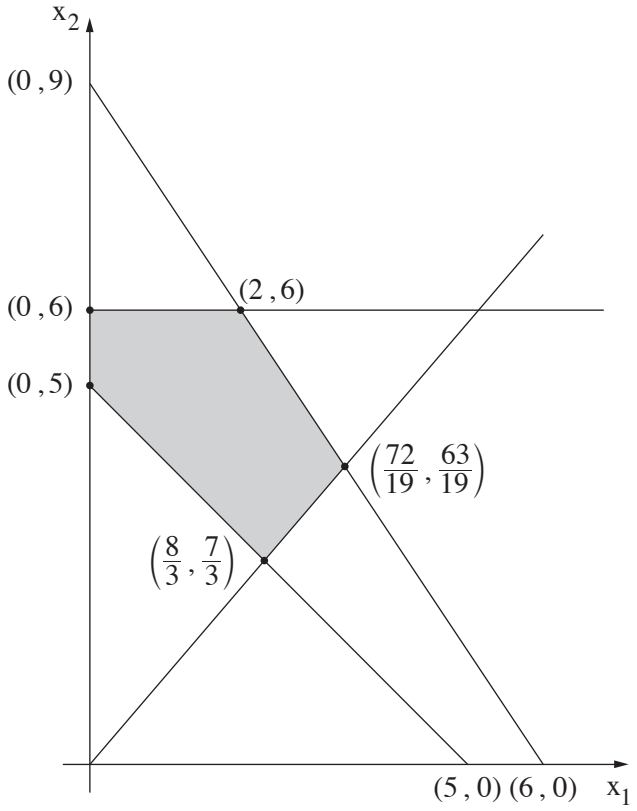
(2) $x_1 \geq 0$

(3) $0 \leq x_2 \leq 6$

(4) $7x_1 \leq 8x_2$

(5) $3x_1 + 2x_2 \leq 18$

أمامك رسم لمجال الحلول الممكنة للمسألة المعطاة .



(انتبه : تكملة السؤال في الصفحة التالية .)

/ يتبع في صفحة 13 /

كلّ واحد من البنود "أ"–"د" التي أمامك يتطرّق إلى مسألة التخطيط الخطّيّ المعطاة.
البنود "أ"–"د" لا تتعلّق ببعضها البعض. أجب عن جميع البنود.
معطاة أربعة أقوال i-iv. لكلّ واحد من البنود "أ"–"د" التي أمامك يوجد قول واحد صحيح فقط.

i يوجد فقط حلّ أمثل وحيد.
ii توجد لانهاية من الحلول المثلى.
iii الحلّ الأمثل ليس محصورًا.
iv لا يوجد حلّ أمثل.

بالنسبة لكلّ واحد من البنود "أ"–"د" التي أمامك، حدِّد أيّ قول من الأقوال i-iv هو الصحيح. اكتب في دفترك الحرف الذي يدلّ على البند ("أ"–"د")، وانسخ القول الصحيح 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. نُغيّر فقط دالّة هدف المسألة المعطاة في بداية السؤال إلى $\max\{z = 3x_1 + 5x_2\}$.
أيّ قول هو الصحيح بعد التغيير؟ علّل إجابتك.

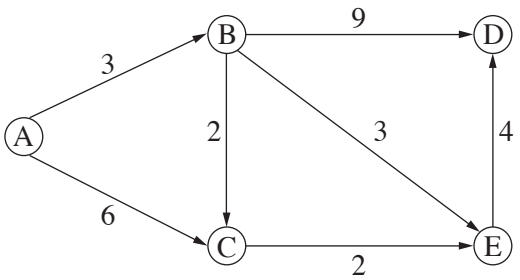
جـ. نُغيّر فقط دالّة هدف المسألة المعطاة في بداية السؤال إلى $\min\{z = 5x_1 + 5x_2\}$.
أيّ قول هو الصحيح بعد التغيير؟ علّل إجابتك.

د. نُضيف اضطرارًا إضافيًّا إلى المسألة المعطاة في بداية السؤال، وهو: $x_1 > 2x_2$.
أيّ قول هو الصحيح بعد إضافة الاضطرار؟ علّل إجابتك.

/ يتبع في صفحة 14 /

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

أ. $G=(V,E)$ هو **رسم بيانيّ موجَّه**:



(1) اعرض الرسم البيانيّ بواسطة مصفوفة مجاوَرات.

(2) جد أقصر مسارات من الرأس A إلى كلّ واحدة من العُقَد التي في الرسم البيانيّ.
صف كلّ واحد من المسارات بصورة تخطيطيّة.

ب. (1) في الجدول الذي أمامك معطى حلّ أساسيّ ممكن نَتَجَ بعد k تكرارات بالنسبة لمسألة نقل، ومعطاة قيمة u_2.

المصادر	الغايات				العرض	u_i
	A	B	C	D		
1	8 (30)	7	3	2	30	
2	4 (10)	9 (70)	4 (10)	4	90	0
3	3	2	7 (40)	8 (40)	80	
الطلب	40	70	50	40		
v_j						

i. انسخ الجدول إلى دفترك، وأكمل فيه القيَم $u_1, u_3, v_1, v_2, v_3, v_4$.

ii. اشرح لماذا الحلّ المعطى ليس أمثل.

iii. عليك تنفيذ تكرار إضافيّ، أي التكرار $k+1$. ما هو المتغيّر الذي يخرج من القاعدة في هذا التكرار؟

iv. ارسم في دفترك جدولاً جديدًا، واكتب فيه الحلّ الذي يَنتُج بعد هذا التكرار.

v. هل انخفضت التكلفة الكلّيّة لمسألة النقل؟

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

(2) الجدول الذي أمامك يمثّل مسألة نقل معيّنة.

المصادر	**الغايات**			**العرض**
	A	B	C	
1	10	5	7	10
2	12	8	4	18
3	8	2	5	10
الطلب	16	12	10	

i. نُغيّر طلب الغاية A من 16 إلى 10 ، بدون تغيير باقي الطلبات والعروض التي في الجدول.
ارسم في دفترك جدولاً جديدًا يشمل جميع المعطيات التي في الجدول المعطى، ويُمكِّن إيجاد حلّ أساسيّ ممكن، بحسب طريقة الزاوية الشماليّة – الغربيّة. يجب أن يشمل الجدول الأسعار إلى كلّ غاية.

ii. جد الحلّ الأساسيّ الممكن بحسب طريقة الزاوية الشماليّة – الغربيّة.

/ يتبع في صفحة 16 /

موديلات حسابيّة

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

11. أمامك اللغتان L_1 وَ L_2 فوق الأبجديّة $\{0,1\}$:

$L_1 = \{0^i 1^{2n} \mid i \geq 1,\ n = i \bmod 3\}$

$L_2 = \{0^i 1^{n+i} \mid i \geq 1,\ n = i \bmod 3\}$

أ. اكتب كلمة بطول 6 تتبع للّغة L_1 وكلمة بطول 6 تتبع للّغة L_2 .

ب. إذا كانت اللّغة L_1 نظاميّة – ابنِ أوتوماتًا نهائيًّا محدودًا ليس كاملاً يتلقّى اللغة، وإذا كانت اللّغة غير نظاميّة – ابنِ أوتومات راصّة يتلقّى اللّغة.

جـ. إذا كانت اللّغة L_2 نظاميّة – ابنِ أوتوماتًا نهائيًّا محدودًا ليس كاملاً يتلقّى اللغة، وإذا كانت اللّغة غير نظاميّة – ابنِ أوتومات راصّة يتلقّى اللّغة.

12. ابنِ آلة تيورينچ تتلقّى كمُدخَل كلمات فوق الأبجديّة $\{0,1\}$.

تُعيد الآلة الكلمة المُتلقَّاة بدون الأصفار، أي أنّ الكلمة التي تُعاد تحوي فقط أرقام الـ 1، بين إشارتَي $ في مكان ما على الشريط.

مثلاً، إذا كان الشريط يبدو على النحو التالي:

⊢	1	1	0	0	1	0	1	△	△	....

المُخرَج يمكن أن يكون مثلاً:

....	△	$	1	1	1	1	$	△	.....

/ يتبع في صفحة 17 /

برمجة موجّهة كائنات بلغة Java

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

13. معطاة الفئات : A , B , Driver

```
public class A {
    private int val ;
    public A () {
        this.val = 1 ;
    }

    public A (int val) {
        this.val = val ;
    }

    public int getVal () {
        return this.val ;
    }

    public boolean equals (Object other) {
        System.out.println ("AObject") ;
        if (other instanceof A)
            return (this.val == ((A) other).val) ;
        return false;
    }
}
```

```
public class B extends A {
    private String st ;
    public B () {
        this.st = "B" ;
    }

    public B (String st, int val) {
        super (val) ;
        this.st = st ;
    }

    public String getSt () {
        return this.st ;
    }

    public boolean equals (Object other) {
        System.out.println ("BObject") ;
        if (other instanceof B)
            return (this.st.equals (((B) other).st)
            && this.getVal () == ((B) other).getVal ()) ;
        return false;
    }

    public boolean equals(A other) {
        System.out.println ("BA") ;
        if (other instanceof B)
            return (this.st.equals (((B) other).st)
            && this.getVal () == ((B) other).getVal ()) ;
        return false;
    }

    public boolean equals (B other) {
        System.out.println ("BB") ;
        return (this.st.equals (other.st)
        && this.getVal () == other.getVal ()) ;
    }
}
```

(انتبه : تكملة السؤال في الصفحة التالية .)

/ يتبع في صفحة 18 /

```
public class Driver {
    public static void main (String[] args) {
        A a1 = new A () ;
        A a2 = new A (5) ;
        A ab = new B () ;
        B b1 = new B ("B", 1) ;
        B b2 = new B ("B", 5) ;
        //  *** //
    }
}
```

أ. ارسم كائنات تكوّنت في العمليّة main . بالنسبة لكلّ كائن عليك ذكر قيَم صفاته.

ب. أمامك ثمانية أوامر، 1-8 .

- عوِّض (ضع) كلّ واحد من الأوامر في المكان //***// في العمليّة main للفئة Driver .
- اكتب ماذا يكون المُخرَج في أعقاب تعويض كلّ واحد من الأوامر.

```
1. if (a1.equals (b1)) System.out.println (1) ;
2. if (b1.equals (a1)) System.out.println (2) ;
3. if (a1.equals (ab)) System.out.println (3) ;
4. if (ab.equals (a1)) System.out.println (4) ;
5. if (b1.equals (ab)) System.out.println (5) ;
6. if (ab.equals (b1)) System.out.println (6) ;
7. if (a1.equals (a2)) System.out.println (7) ;
8. if (b1.equals (b2)) System.out.println (8) ;
```

/ يتبع في صفحة 19 /

14. معطاة الفئات: Driver , First , Second , Third .

```
public class First {
    public static int count = 0 ;
    private String str ;

    public First (String str)  {
        count ++ ;
        this.str = str  ;
    }

    public First (First g)  {
        count ++ ;
        this.str = "Copy" + g.str ;
    }

    public void setStr (String str)   {
        this.str = str ;
    }

    public void print ()   {
        System.out.println ("First" + this.str) ;
    }
}
```

```
public class Second extends First  {
    private First f;
    public Second (First fx , First fy)   {
        super (fx) ;
        this.f = fy  ;
    }

    public void setStr (String str)   {
        super.setStr (str) ;
        this.f = new First (str) ;
    }

    public void print ()   {
        System.out.println ("Second") ;
        super.print () ;
        this.f.print () ;
    }
}
```

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 20 /

```
public class Third  {
    private int curr ;
    private First[] arr ;
    public Third (int count )  {
        curr = 0 ;
        arr = new First [count] ;
    }

    public void print ()  {
        for ( int i = 0 ; i < curr ; i++ )
            arr [i].print () ;
    }

    public void add ( First s )  {
        if ( curr < arr.length ) {
            arr [curr] = s ;
            curr ++ ;  }
    }
}
```

```
public class Driver  {
    public static void main (String[] args) {
        First f1 = new First ("One") ;
        First f2 = new First ("Two") ;
        First f3 = new First (f2) ;
        Second s1 = new Second (f1, f3) ;
        Third t = new Third (First.count) ;
        t.add (f1) ;
        t.add (f2) ;
        t.add (s1) ;
        System.out.println (First.count) ;
        t.print () ;
        System.out.println (" ----------") ;
        f1.setStr ("Five") ;
        System.out.println (First.count) ;
        t.print () ;
    }
}
```

أ. ارسم مخطّطًا هرميًّا بين الفئات First , Second , Third . يجب الإشارة إلى توريث بواسطة السهم ◁——— ، وإلى احتواء بواسطة الإشارة ◆——— .

ب. تتبّع تنفيذ العمليّة main للفئة Driver :

(1) اعرض محتوى جميع الكائنات التي تكوّنت.

(2) اكتب المُخرَج الذي يَنتُج.

/ يتبع في صفحة 21 /

برمجة موجّهة كائنات بلغة C#

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

15. معطاة الفئات : A, B, Driver

```
public class A {
    private int val ;

    public A () {
        this.val = 1 ;
    }

    public A (int val) {
        this.val = val ;
    }

    public int GetVal () {
        return this.val ;
    }

    public virtual bool Equals (Object other) {
        Console.WriteLine ("AObject") ;
        if (other is A)
            return (this.val == ((A) other).val) ;
        return false ;
    }
}
```

```
public class B : A {
    private string st ;

    public B () {
        this.st = "B" ;
    }
    public B (string st, int val)
        : base (val) {
        this.st = st ;
    }
    public string GetSt () {
        return this.st ;
    }
    public override bool Equals (Object other) {
        Console.WriteLine ("BObject ") ;
        if (other is B)
            return (this.st.Equals (((B) other).st)
            && this.GetVal () == ((B) other).GetVal ()) ;
        return false ;
    }
    public bool Equals (A other) {
        Console.WriteLine ("BA");
        if (other is B)
            return (this.st.Equals (((B) other). st)
            && this.GetVal () == ((B) other).GetVal ()) ;
        return false;
    }
    public bool Equals (B other) {
        Console.WriteLine("BB") ;
        return (this.st.Equals (other. st)
        && this.GetVal () == other.GetVal ()) ;
    }
}
```

(انتبه : تكملة السؤال في الصفحة التالية .)
/ يتبع في صفحة 22 /

```
public class Driver {
public static void Main (string[] args) {
        A a1 = new A () ;
        A a2 = new A (5) ;
        A ab = new B () ;
        B b1 = new B ("B", 1) ;
        B b2 = new B ("B", 5) ;
        //  *** //
}
}
```

أ. ارسم كائنات تكوّنت في العمليّة Main . بالنسبة لكلّ كائن عليك ذكر قيَم صفاته.

ب. أمامك ثمانية أوامر، 1-8 .

- عوِّض (ضع) كلّ واحد من الأوامر في المكان //***// في العمليّة Main للفئة Driver .
- اكتب ماذا يكون المُخرَج في أعقاب تعويض كلّ واحد من الأوامر.

1. if (a1.Equals (b1)) Console.WriteLine (1) ;
2. if (b1.Equals (a1)) Console.WriteLine (2) ;
3. if (a1.Equals (ab)) Console.WriteLine (3) ;
4. if (ab.Equals (a1)) Console.WriteLine (4) ;
5. if (b1.Equals (ab)) Console.WriteLine (5) ;
6. if (ab.Equals (b1)) Console.WriteLine (6) ;
7. if (a1.Equals (a2)) Console.WriteLine (7) ;
8. if (b1.Equals (b2)) Console.WriteLine (8) ;

/ يتبع في صفحة 23 /

16. معطاة الفئات : Driver, First, Second, Third .

```
public class First
{
    public static int count=0 ;
    private string str ;

    public First (string str)  {
        count ++  ;
        this.str = str ;
    }

    public First (First g)  {
        count ++ ;
        this.str = "Copy" + g.str ;
    }

    public virtual void SetStr (string str)   {
        this.str = str ;
    }

    public virtual void Print ()    {
        Console.WriteLine ("First" + this.str ) ;
    }
}
```

```
public class Second : First  {
    private First f;

    public Second (First fx, First fy)
    : base (fx)   {
        this.f = fy ;
    }

    public override void SetStr (string str)   {
        base.SetStr (str) ;
        this.f = new First (str) ;
    }

    public override void Print ()    {
        Console.WriteLine ("Second") ;
        base.Print () ;
        this.f.Print ();
    }
}
```

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 24 /

```
public class Third  {
      private int curr ;
      private First[] arr ;

    public Third (int count)  {
          curr = 0 ;
          arr = new First [count] ;
    }

    public void Print ()  {
          for (int i = 0 ; i < curr ; i++)
                arr [i].Print () ;
      }

    public void Add (First s)  {
          if (curr < arr.Length) {
                arr [curr] = s ;
                curr++ ;  }
    }
}
```

```
public class Driver
{
    public static void Main (string[] args) {
          First f1 = new First ("One") ;
          First f2 = new First ("Two") ;
          First f3 = new First (f2) ;
          Second s1 = new Second (f1, f3) ;
          Third t = new Third (First.count) ;
          t.Add (f1) ;
          t.Add (f2) ;
          t.Add (s1) ;
          Console.WriteLine (First.count) ;
          t.Print() ;
          Console.WriteLine (" ----------") ;
          f1.SetStr ("Five") ;
          Console.WriteLine (First.count) ;
          t.Print () ;
    }
}
```

أ. ارسم مخطّطًا هرميًّا بين الفئات First , Second , Third . يجب الإشارة إلى توريث بواسطة السهم ◁——— ، وإلى احتواء بواسطة الإشارة ◆——— .

ب. تتبّع تنفيذ العمليّة Main للفئة Driver :

(1) اعرض محتوى جميع الكائنات التي تكوّنت.

(2) اكتب المُخرَج الذي يُنتَج.

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.