מדינת ישראל
משרד החינוך

סוג הבחינה: בגרות
מועד הבחינה: **קיץ נבצרים**, תשפ"א, 2021
מספר השאלון: **899381**
תרגום לערבית (2)

دولة إسرائيل
وزارة التّربية والتّعليم

نوع الامتحان: بجروت
موعد الامتحان: **صيف للمتعذّر عليهم**، 2021
رقم النّموذج: **899381**
ترجمة إلى العربيّة (2)

انتبه: في هذا الامتحان توجد تعليمات خاصّة.
يجب الإجابة عن الأسئلة حسب التّعليمات.

מדעי המחשב
הוראות לנבחן

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח ההערכה:
בשאלון זה שלושה פרקים.
יש לבחור בחמש שאלות מתוך שלושת הפרקים.
לכל שאלה – 20 נקודות (5×20).
סך הכול – 100 נקודות.
שים לב: אם תבחר לענות על שאלות מן הפרק השלישי, בחר בשאלות מתוך מסלול אחד בלבד.

ג. חומר עזר מותר בשימוש: כל חומר עזר, חוץ ממחשבון שיש בו אפשרות תכנות.

ד. הוראות מיוחדות:

1. **רשום על הכריכה החיצונית** של המחברת את ה**מסלול שלמדת**, אם תבחר לענות על שאלות מן הפרק השלישי, ואם לא — רשום **"ללא מסלול"**.
המסלול הוא אחד מארבעת המסלולים האלה: מערכות מחשב ואסמבלי, מבוא לחקר ביצועים, מודלים חישוביים, תכנות מונחה עצמים.

2. את כל התוכניות שאתה נדרש לכתוב בשפת מחשב בפרקים הראשון והשני, כתוב בשפה אחת בלבד — Java או C#.

הערה: לא יורדו לך נקודות אם תכתוב בתוכניות שאתה כותב אות גדולה במקום אות קטנה או להפך.

علم الحاسوب
تعليمات للممتحَن

أ. مدّة الامتحان: ثلاث ساعات.

ب. مبنى النّموذج وتوزيع الدّرجات:
في هذا النّموذج ثلاثة فصول.
عليك اختيار خمسة أسئلة من ثلاثة الفصول.
لكلّ سؤال – 20 درجة (5×20).
المجموع – 100 درجة.
انتبه: اذا اخترتَ الإجابة عن أسئلة من الفصل الثالث، اختر أسئلة من مسار واحد فقط.

جـ. موادّ مساعدة يُسمح استعمالها: كلّ مادّة مساعدة، عدا الحاسبة التي فيها إمكانيّة برمجة.

د. تعليمات خاصّة:

1. **سجّل على الغلاف الخارجيّ** للدفتر اسم **المسار الذي تعلّمته**، إذا اخترتَ الإجابة عن أسئلة من الفصل الثالث. وإذا لم تختر – سجّل "**بدون مسار**".
المسار هو أحد المسارات الأربعة التالية: أنظمة حاسوب وأسمبلي، مقدّمة لبحث الأداءات، موديلات حسابيّة، برمجة موجّهة كائنات.

2. اكتب جميع البرامج التي يُطلب منك كتابتها بلغة حاسوب في الفصلين الأوّل والثاني، بلغة واحدة فقط – Java أو C#.

ملاحظة: لن تُخصم درجات إذا كتبتَ في البرامج التي تكتبها 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 اختيار خمسة أسئلة من ثلاثة الفصول. لكلّ سؤال – 20 درجة.
سجّل على الغلاف الخارجيّ للدفتر اسم المسار الذي تعلّمته، إذا اخترتَ الإجابة عن أسئلة من الفصل الثالث. وإذا لم تختر – سجّل "بدون مسار".

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);
```

الفصل الأوّل

1. "نصّ مزدوج" هو نصّ ليس فارغًا مركَّب من نصّين فرعيّين متطابقين يظهران الواحد تلو الآخر.
أمثلة لِـ "نصّ مزدوج": "abcabc" ، "a@a@" ، "**".
أمثلة لنصوص **ليست** "نصّ مزدوج": "232323" , "ab&ab" , "abba" , "a@a" , "ab" , "%" .

اكتب عمليّة خارجيّة اسمها isDouble بلغة Java أو IsDouble بلغة C# تتلقّى نصًّا – str
وتُعيد true إذا كان str هو "نصّ مزدوج"، خلاف ذلك تُعيد false .

ملاحظة: افترِضْ أنّ النصّ str ليس فارغًا.

/ يتبع في صفحة 3 /

2. الزوج "نهاية بداية" هو زوج عددين موجبين وصحيحين، فيه الرقم الأوّل في أحد العددين مطابق للرقم الأخير في العدد الآخر (لا توجد أهمّيّة لترتيب العددين) .
عدد أصغر من 10 (أي، المكوَّن من رقم واحد فقط) يُعتبَر الرقم الأوّل والرقم الأخير أيضًا .
أمثلة لزوج "نهاية بداية" :

العدد I	العدد II	العدد I	العدد II
3090	6783	25	**5**34

معطاة الفئة **PairOfNums** – زوج عددين موجبين، ولها صفتان :

```
public class PairOfNums {
    private int num1 ;
    private int num2 ;
}
```

افترِضْ أنّ العمليّتين get وَ set بلغة Java والعمليّتين Get وَ Set بلغة #C قائمتان بالنسبة لصفتَي الفئة .
أمامك واجهة تطبيق جزئيّة للفئة **PairOfNums** بلغة Java وبلغة #C .
بإمكانك استعمال عمليّات واجهة التطبيق بدون الحاجة لتطبيقها .

وصف العمليّة	عنوان العمليّة
عمليّة بنائيّة تتلقّى قيَمًا بالنسبة لصفتَي الفئة .	public PairOfNums (int num1, int num2)
عمليّة تُعيد true إذا كان الكائن هو زوج "نهاية بداية" ، خلاف ذلك، تُعيد false .	بلغة **Java** public boolean endStart () بلغة **C#** public bool EndStart ()

أمامك عنوان عمليّة خارجيّة بلغة Java :

```
public static PairOfNums[] generate (int n)
```

أمامك عنوان عمليّة خارجيّة بلغة #C :

```
public static PairOfNums[] Generate (int n)
```

العمليّة generate / Generate تبني مصفوفة فارغة لكائنات من نمط PairOfNums بكبَر n . بالنسبة لكلّ خليّة في المصفوفة، تقترع العمليّة عددين بين 1 وَ 1,000 (بما في ذلك 1 وَ 1,000)، وتفحص إذا كانا زوج "نهاية بداية" . إذا كانا كذلك، يُدخَل إلى الخليّة كائن قيمَتا صِفَتَيْه هما زوج العدين اللذين اقتُرِعا . خلاف ذلك، تواصل العمليّة اقتراع عددين حتّى ينتج زوج "نهاية بداية" ، وعندها يُدخَل إلى الخليّة كائن قيمَتا صِفَتَيْه هما زوج العددين اللذين اقتُرِعا .
بعد أن تمتلئ المصفوفة، تُعيد العمليّة المصفوفة .
طبِّق العمليّة .

ملاحظة : افتَرِضْ أنّ n أكبر من صفر .

/ يتبع في صفحة 4 /

3. مصفوفة ثنائيّة الأبعاد لأعداد صحيحة تُسمّى "مصفوفة ركنيّة"، إذا تحقّق فيها الشرطان التاليان:

- عدد الأعمدة يساوي عدد الأسطر، وأكبر من 0 .
- جميع القيَم في السطر الأوّل وفي العمود الأوّل هي 1 .

في السطر الثاني وفي العمود الثاني ابتداءً من الحدّ الثاني، جميع القيَم هي 2 .

في السطر الثالث وفي العمود الثالث ابتداءً من الحدّ الثالث، جميع القيَم هي 3 ، وهكذا حتّى نهاية المصفوفة.

أمامك مثال لِـ "مصفوفة ركنيّة" بكِبَر 5×5 .

	0	1	2	3	4
0	1	1	1	1	1
1	1	2	2	2	2
2	1	2	3	3	3
3	1	2	3	4	4
4	1	2	3	4	5

اكتب عمليّة خارجيّة اسمها isCorner بلغة Java أو IsCorner بلغة C# تتلقّى مصفوفة ثنائيّة الأبعاد– matrix لأعداد صحيحة وتُعيد true إذا كانت "مصفوفة ركنيّة"، خلاف ذلك تُعيد false .

/ يتبع في صفحة 5 /

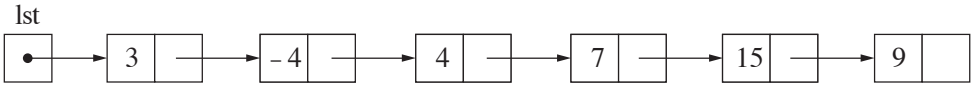
الفصل الثاني

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

4. "سلسلة منظَّمة" هي سلسلة حلقات من نمط صحيح، يتحقّق فيها الشرطان التاليان:

- عدد الحلقات في السلسلة هو زوجيّ.
- جميع الأعداد في النصف الأوّل من السلسلة أصغر من جميع الأعداد في النصف الثاني من السلسلة.

مثال لِـ "سلسلة منظَّمة":



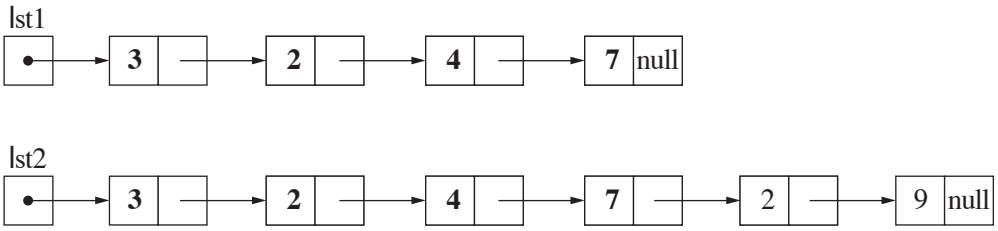
أ. اكتب عمليّة خارجيّة اسمها isArranged بلغة Java أو IsArranged بلغة C# تتلقّى سلسلة حلقات lst من نمط صحيح، ليست null، وتُعيد true إذا كانت "سلسلة منظَّمة"، خلاف ذلك تُعيد false.
ملاحظة: يجب الحفاظ على السلسلة lst.

ب. ما هي تعقيدات زمن تشغيل هذه العمليّة؟ علّل.

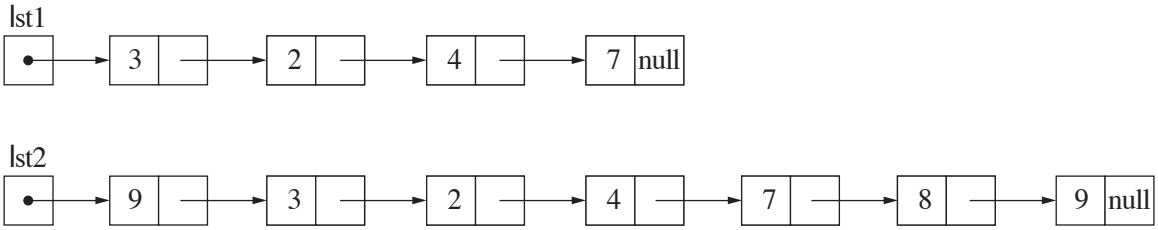
/ يتبع في صفحة 6 /

5. أ. سلسلة الحلقات lst1 هي "سلسلة فرعيّة ابتدائيّة" لسلسلة الحلقات lst2 إذا كانت جميع قيَم lst1 تظهر بنفس التسلسل من بداية lst2 (يمكن أن تكون قيَم إضافيّة في lst2 بعد هذا التسلسل) .

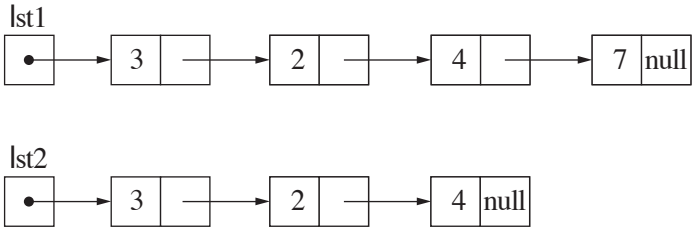
مثال لِـ lst1 التي هي "سلسلة فرعيّة ابتدائيّة" لِـ lst2 :



مثال لِـ lst1 **ليست** "سلسلة فرعيّة ابتدائيّة" لِـ lst2 :



مثال إضافيّ لِـ lst1 **ليست** "سلسلة فرعيّة ابتدائيّة" لِـ lst2 :

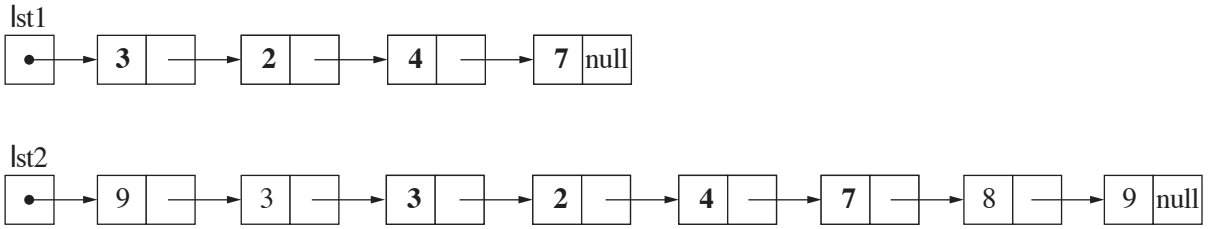


اكتب عمليّة خارجيّة اسمها isPrefix بلغة Java أو IsPrefix بلغة C# تتلقّى سلسلتَي حلقات من نمط صحيح lst1 وَ lst2 ليستا null .
تُعيد العمليّة true إذا كانت lst1 "سلسلة فرعيّة ابتدائيّة" لِـ lst2 ، خلاف ذلك تُعيد false .
ملاحظة: يجب الحفاظ على السلسلتين lst1 وَ lst2 .

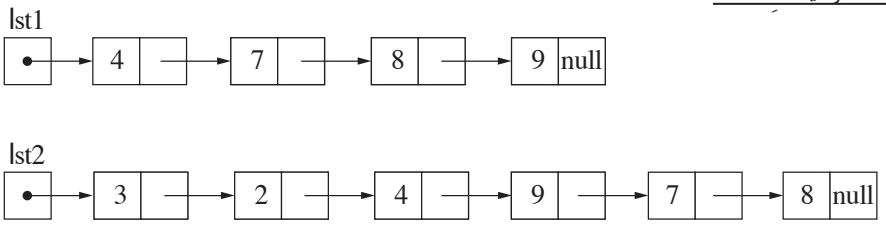
(انتبه : تكملة السؤال في الصفحة التالية .)
/ يتبع في صفحة 7 /

ב. سلسلة الحلقات lst1 هي "سلسلة فرعيّة" لسلسلة الحلقات lst2 إذا كانت جميع قيَم lst1 تظهر في مكان ما في lst2 بنفس التسلسل.

مثال لِـ lst1 التي هي "سلسلة فرعيّة" لِـ lst2:



مثال لِـ lst1 **ليست** "سلسلة فرعيّة" لِـ lst2:



اكتب عمليّة خارجيّة اسمها isSubChain بلغة Java أو IsSubChain بلغة #C تتلقّى سلسلتَي حلقات من نمط صحيح lst1 وَ lst2 ليستا null .
تُعيد العمليّة true إذا كانت lst1 "سلسلة فرعيّة" لِـ lst2 ، خلاف ذلك تُعيد false .
ملاحظة: يجب استعمال العمليّة التي كتبتَها في البند "أ".

/ يتبع في صفحة 8 /

6. معطاة الفئة **CovidTest** ، التي تمثّل شخصًا فُحص بفحص كورونا، ولها 4 صفات:

- name – اسم المفحوص من نمط نصّ.
- id – رقم هويّة من نمط نصّ.
- cityCode – كود لمدينة السكن، من نمط صحيح (مثلاً: 1030 بالنسبة لأشدود، 23 بالنسبة لعكّا).
- sick – متغيّر من نمط بولياني، يتلقّى true إذا كان المفحوص مريضًا، خلاف ذلك يتلقّى false .

افترِضْ أنّ العمليّتين get وَ set بلغة Java والعمليّتين Get وَ Set بلغة #C قائمتان بالنسبة لصفات الفئة.

اكتب عمليّة خارجيّة mostSick بلغة Java أو MostSick بلغة #C تتلقّى دورًا – q ليس فارغًا من نمط **CovidTest** . تُعيد العمليّة كود المدينة التي كمّيّة المرضى فيها هي الأكبر.

ملاحظات: – موقع المفحوصين في الدور ليس حسب ترتيب ما.

– كلّ مفحوص يظهر فقط مرّة واحدة في الدور.

– كود المدينة لا يتعلّق بِكبَر الدور (مثال: يمكن أن يكون عدد الحدود في الدور 1,000، ويوجد كود مدينة رقمه 5000).

– لا حاجة لحفظ الدور.

افترِضْ أنّه توجد فقط مدينة واحدة فيها كمّيّة المرضى هي الأكبر.

/ يتبع في صفحة 9 /

7. انتبه ! لهذا السؤال صيغتان : بلغة Java في صفحة 9 وبلغة #C في صفحة 10 .

بلغة Java :

أ. أمامك العمليّة sod1 :

```
public static int sod1(int x) {
    if (x < 10)
        return x;
    return sod1(x / 10);
}
```

(1) اكتب القيمة التي تُعاد عند استدعاء العمليّة sod1(123) . عليك تبيين متابعة .

(2) بالنسبة لِـ x أكبر من 9 ، ما هو هدف العمليّة sod1 ؟

ب. أمامك العمليّة sod2 :

```
public static int sod2(int x) {
    if (x < 10)
        return 0;
    return sod2(x / 10) * 10 + (x % 10);
}
```

(1) اكتب القيمة التي تُعاد عند استدعاء العمليّة sod2(123) . عليك تبيين متابعة .

(2) بالنسبة لِـ x أكبر من 9 ، ما هو هدف العمليّة sod2 ؟
افترِضْ أنّ الرقم 0 لا يظهر في العدد x .

جـ. أمامك العمليّة sod3 :

```
public static int sod3(int x, int y) {
    if (y == 0)
        return x;
    int tmp1 = (x * 10) + sod1(y);
    int tmp2 = sod2(y);
    return sod3(tmp1, tmp2);
}
```

(1) اكتب القيمة التي تُعاد عند استدعاء العمليّة sod3(123, 68) . عليك تبيين متابعة .
في هذا البند الفرعيّ، لا حاجة لتنفيذ متابعة للعمليّتين sod1 وَ sod2 .

(2) اكتب القيمة التي تُعاد عند استدعاء العمليّة sod3(35, 792) (لا حاجة لتنفيذ متابعة) .

(3) بالنسبة لِـ x وَ y أكبر من 0 ، ما هو هدف العمليّة sod3 ؟
افترِضْ أنّ الرقم 0 لا يظهر في العدد y .

/ يتبع في صفحة 10 /

بلغة #C

أ. أمامك العمليّة Sod1 :

```
public static int Sod1(int x) {
    if (x < 10)
        return x;
    return Sod1(x / 10);
}
```

(1) اكتب القيمة التي تُعاد عند استدعاء العمليّة Sod1(123) . عليك تبيين متابعة.

(2) بالنسبة لِـ x أكبر من 9 ، ما هو هدف العمليّة Sod1 ؟

ب. أمامك العمليّة Sod2 :

```
public static int Sod2(int x) {
    if (x < 10)
        return 0;
    return Sod2(x / 10) * 10 + (x % 10);
}
```

(1) اكتب القيمة التي تُعاد عند استدعاء العمليّة Sod2(123) . عليك تبيين متابعة.

(2) بالنسبة لِـ x أكبر من 9 ، ما هو هدف العمليّة Sod2 ؟

افترِضْ أنّ الرقم 0 لا يظهر في العدد x .

جـ. أمامك العمليّة Sod3 :

```
public static int Sod3(int x, int y) {
    if (y == 0)
        return x;
    int tmp1 = (x * 10) + Sod1(y);
    int tmp2 = Sod2(y);
    return Sod3(tmp1, tmp2);
}
```

(1) اكتب القيمة التي تُعاد عند استدعاء العمليّة Sod3(123, 68) . عليك تبيين متابعة.

في هذا البند الفرعيّ، لا حاجة لتنفيذ متابعة للعمليّتين Sod1 وَ Sod2 .

(2) اكتب القيمة التي تُعاد عند استدعاء العمليّة Sod3(35, 792) (لا حاجة لتنفيذ متابعة) .

(3) بالنسبة لِـ x وَ y أكبر من 0 ، ما هو هدف العمليّة Sod3 ؟

افترِضْ أنّ الرقم 0 لا يظهر في العدد y .

/ يتبع في صفحة 11 /

الفصل الثالث

في هذا الفصل أسئلة في أربعة مسارات:
أنظمة حاسوب وأسمبلي، في الصفحتين 11-12.
مقدّمة لبحث الأداءات، في الصفحتين 13-14.
موديلات حسابيّة، في الصفحة 15.
برمجة موجّهة كائنات بلغة Java ، في الصفحات 16-19؛ برمجة موجّهة كائنات بلغة #C، في الصفحات 20-23.

إذا اخترتَ الإجابة عن أسئلة من هذا الفصل، اختر أسئلة من مسار واحد فقط.

أنظمة حاسوب وأسمبلي

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

أ. أمامك قطعة برنامج مكتوبة باللغتين Java وَ #C.

```
while ( x < 10 && y > 0 )
{
    if ( x % 2 == 1 )
        x = x * 2;
    else
        y = y − 2;
}
```

اكتب قطعة برنامج ملائمة بلغة أسمبلي.
افترِضْ أنّ المتغيّرين x وَ y موجَّهان (signed) ومخزونان في الخازنين BX وَ CX بالتلاؤم.

ب. معطى أنّه في الراصّة، في خليّتين متجاورتين – في رأس الراصّة وفي العنوان الأعلى منه (في SP وفي SP+1) – مخزون عددان غير موجَّهين (unsigned) بين 0 وَ 255 (بما في ذلك 0 وَ 255). العدد الأقلّ من بينهما موجود في رأس الراصّة.
اكتب برنامجًا فرعيًّا اسمه TEST يفحص إذا كان العددان متعاقبين (مثل 1-2 ، 15-16 ، 201-202 وما شابه).
إذا كانا متعاقبين، يخزن البرنامج الفرعيّ 1 في الخازن AL ، خلاف ذلك يخزن 0 .

/ يتبع في صفحة 12 /

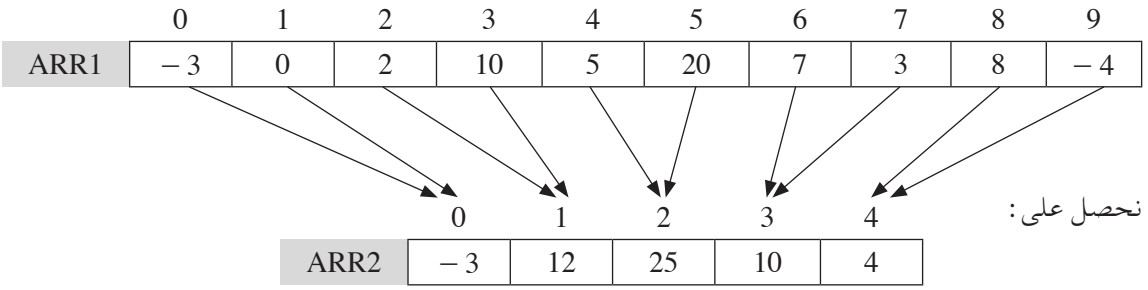
9. في هذا السؤال بندان، "أ – ب"، لا توجد علاقة بينهما. أجب عن البندين.

أ. في مقطع المعطيات الذي أمامك عُرِّفت المعطيات التالية:

```
ARR1  DW 10 DUP(?)
ARR2  DW 5 DUP(?)
```

اكتب قطعة برنامج تُدخِل إلى خلايا في المصفوفة – ARR2 مجموع كلّ زوج حدّين متعاقبين في المصفوفة – ARR1 (في المؤشّرات: 0-1 ، 2-3 ، 4-5 ، وهكذا) حسب ترتيب ظهورها.

مثال: بالنسبة للمصفوفة – ARR1 التي أمامك:



ب. أمامك قطعة كود:

```
     MOV   CL , DL
     MOV   AL , DL
     MOV   DL , 0
A1:  ADD   DL , AL
     LOOP  A1
```

(1) بافتراض أنّه في الخازن DL مخزونة القيمة 5 ، ماذا ستكون القيمة المخزونة في الخازن DL في نهاية تنفيذ قطعة الكود؟

(2) ماذا يجب أن تكون قيمة DL كي تكون قيمته 100 في نهاية تنفيذ قطعة الكود؟

(3) اشرح بجملة واحدة ماذا تنفّذ قطعة الكود.

/ يتبع في صفحة 13 /

مقدّمة لبحث الأداءات

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

أ. في الجدول الذي أمامك معطاة مسألة نقل.

العرض	الغايات				المصادر
	4	3	2	1	
200	3	5	7	6	1
400	2	4	7	15	2
200	7	6	13	17	3
	180	270	250	100	**الطلب**

(1) انسخ الجدول إلى دفترك، واقترِحْ حلّاً أساسيًّا ممكنًا حسب طريقة الزاوية الشماليّة – الغربيّة.

(2) إذا تغيّر سعر النقل من المصدر 3 إلى الغاية 2 إلى 15، هل يتغيّر الحلّ الأساسيّ؟ علّل.

(3) بافتراض أنّ $u_1 = 0$:

– أكمِل القِيَم v_4 ، v_3 ، v_2 ، v_1 ، u_3 ، u_2 .

– برهن أنّ الحلّ الذي وجدتَه في البند الفرعيّ "أ (1)" ليس أمثل.

– نفِّذ تكرارًا للتحسين، وبيّن أنّ تكلفة النقل قد قلّت.

ب. الرسم البيانيّ $G = (V,E)$ ليس موجَّهًا، وفيه n عُقَد مرقَّمة من 1 حتّى n ، هو رسم بيانيّ "دورانيّ" إذا تحقّقت فيه الشروط التالية:

– $n \geq 4$.

– يوجد قوس بين العقدة V_1 وبين كلّ واحدة من العُقَد الأخرى.

– في كلّ عقدة V_i تُحقِّق $i \neq 1$ وَ $i \neq n$ يوجد قوس بينها وبين العقدة V_{i+1} .

– يوجد قوس بين العقدتين V_n وَ V_2 .

– لا توجد أقواس إضافيّة باستثناء المذكورة أعلاه.

(1) ارسم رسمًا بيانيًّا "دورانيًّا" بالنسبة لـ $n = 6$.

(2) معطى الرسم البيانيّ G_n "الدورانيّ" الذي فيه n عُقَد. نريد إضافة k عُقَد إضافيّة إلى الرسم البيانيّ، مرقَّمة من V_{n+1} وحتّى V_{n+k} .
أيّ قوس يجب إزالته، وما هو عدد الأقواس التي يجب إضافتها كي يبقى الرسم البيانيّ رسمًا بيانيًّا "دورانيًّا" (كدالّة لـ k)؟ علّل إجابتك.

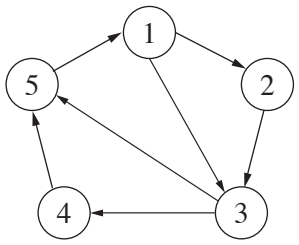
(3) ما هو عدد الأقواس في الرسم البيانيّ G_n "الدورانيّ" الذي فيه n عُقَد (كدالّة لـ n)؟ علّل إجابتك.

/ يتبع في صفحة 14 /

11. الرسم البيانيّ $G = (V, E)$ الموجَّه، والذي فيه n عُقَد مرقَّمة من 1 حتّى n ، هو رسم بيانيّ "مميّز" إذا تحقّقت فيه الشروط التالية:

- n هو فرديّ.
- من كلّ عقدة V_i تُحقِّق $i \% 2 = 0$ يخرج قوس إلى العقدة V_{i+1} .
- من كلّ عقدة V_i تُحقِّق $i \% 2 = 1$ وَ $i \neq n$ يخرج قوس إلى العقدة V_{i+1} وقوس إلى العقدة V_{i+2} .
- من العقدة V_n يخرج قوس إلى العقدة V_1 .
- لا توجد أقواس إضافيّة باستثناء المذكورة أعلاه.

مثال: الرسم البيانيّ G_5 الذي أمامك هو رسم بيانيّ "مميّز":



أ. ارسم رسمًا بيانيًّا "مميّزًا" بالنسبة للرسم البيانيّ G_1 ، الذي فيه عقدة واحدة، وبالنسبة للرسم البيانيّ G_7 ، الذي فيه 7 عُقَد.

ب. (1) ماذا ستكون درجة دخول ودرجة خروج العقدة V_n في الرسم البيانيّ "المميّز" G_1 ، الذي فيه عقدة واحدة؟ علّل إجابتك.

(2) ماذا ستكون درجة دخول ودرجة خروج العقدة V_n في الرسم البيانيّ "المميّز" G_n ، الذي فيه n عُقَد، وَ $n > 1$ ؟" علّل إجابتك.

جـ. ما هو عدد الأقواس في الرسم البيانيّ "المميّز" G_n ، الذي فيه n عُقَد (كدالّة لـ n)؟ علّل إجابتك.

د. بالنسبة للرسم البيانيّ G_5 الذي في المثال، اعرض شجرة امتداديّة صغرى عرضيّة (BFS).

/ يتبع في صفحة 15 /

موديلات حسابيّة

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

أ. معطاة اللغات L_1 وَ L_2 وَ L_3 فوق الأبجديّة $\{a, b\}$:

$L_1 = \{$ a الحرف فيها يظهر التي المرّات عدد من أكبر هو b الحرف فيها يظهر التي المرّات عدد فيها التي الكلمات جميع$\}$

$L_2 = \{$3 مرّات على الأقلّ b الحرف فيها يظهر التي الكلمات جميع$\}$

$L_3 = L_1 \cap \overline{L_2}$

فصّل جميع الكلمات التي تنتج في اللغة L_3 .

ب. ابنِ أوتوماتًا نهائيًّا محدودًا كاملاً فوق الأبجديّة $\{a, b, c\}$ يتلقّى فقط كلمات فيها أحد التسلسلين aa أو bb .
الأوتومات لا يتلقّى كلمات لا يظهر فيها أيّ واحد من التسلسلين، أو كلمات يظهر فيها كلا التسلسلين.

أمثلة لكلمات يتلقّاها الأوتومات:

caac, bbb, aacaa, bba

أمثلة لكلمات لا يتلقّاها الأوتومات:

cac, bab, aabb, bbcaa

13. معطاة عمليّة مكتوبة بلغة Java وبلغة C# .
تتلقّى العمليّة عددًا صحيحًا موجبًا – x .

C#	**Java**

public static int Foo (int x)
{
return (x / 2) + (x % 2);
}
	public static int foo (int x)
	{
	return (x / 2) + (x % 2);
	}

بلغة **Java**

أمامك اللغة L فوق الأبجديّة $\{a,b\}$: $L = \{a^n b^k \mid n > 0 \, , \, k = foo(n)\}$

بلغة **C#**

أمامك اللغة L فوق الأبجديّة $\{a,b\}$: $L = \{a^n b^k \mid n > 0 \, , \, k = Foo(n)\}$

مثال: عندما تكون قيمة n هي 3 ، وقيمة k هي 2 : $k = 3 / 2 + 3 \% 2$;
لذلك الكلمة التي تتبع لِلُّغة هي aaabb .

أ. **(1)** اكتب الكلمة التي تتبع لِلُّغة بالنسبة لـ $n = 2$.

(2) اكتب الكلمة التي تتبع لِلُّغة بالنسبة لـ $n = 5$.

ب. ابنِ أوتومات راصّة يتلقّى اللغة L .

/ يتبع في صفحة 16 /

برمجة موجّهة كائنات

السؤالان 14-15 معدّان للّذين يكتبون بلغة Java .

14. أمامك الفئات A ، B ، C ، D .

```
public class A {

    public static int count = 0;
    protected int num;

    public A() {
        this.num = 3;
        count ++;
    }

    public int getNum() {
        return this.num;

    }

    public void setNum(int num) {
        this.num = num;
    }

    public void foo(int num) {
        this.num = num;
        num++;
    }

    public boolean isEqual(A other) {
        return (this.num == other.num);
    }

    public void func() {
        System.out.println("I am A");
    }
}
```

```
public class B extends A {

    public void func() {
        System.out.println("I am B");
    }

    public boolean isEqual(B other) {
        return ( this == other);
    }
}

-----------------------------------------------------------

public class C extends A { }

-----------------------------------------------------------

public class D {
    private A a;

    public D(A a) {
        this.a = a;
    }

    public void func() {
        (this.a).func ();
    }

    public void isB() {
        System.out.println((this.a) instanceof B);
    }
}
```

/ يتبع في صفحة 17 /

أ. ارسم الخارطة الهرميّة بين الفئات A ، B ، C ، D .

يجب الإشارة إلى توريث بواسطة السهم ◁—— وإلى احتواء بواسطة الإشارة ——◆ .

ب. ارسم الكائنات التي تكوّنت في العمليّة main في الفئة Tester التي أمامك، واكتب المخرَج.

```
public class Tester {
    public static void main(String[] args) {
        A y1 = new A();
        A y2 = new A() ;
        y1.foo(y1.getNum()) ;
        System.out.println(y1.isEqual(y2)) ;
        B y3 = new B();
        A y4 = new B();
        System.out.println(y3.isEqual(y4));
        System.out.println(y4.isEqual(y3));
        B y5 = y3;
        y5.setNum(0);
        System.out.println(y3.isEqual(y5));
        C y6 = new C();
        System.out.println(((B)y4).isEqual(y6));
        D d1 = new D(y4);
        D d2 = new D(y6);
        System.out.println(B.count);
        d1.func();
        d2.func();
        d1.isB();
        d2.isB();
    }
}
```

/ يتبع في صفحة 18 /

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

أ. معطاة الفئات التالية:

```
public class C {
    public void dolt(D d) { System.out.println("cd") ; }
}

public class D extends C {
    public void dolt(D d) { System.out.println("dd") ; }
    public void dolt(C c) { System.out.println("dc") ; }
}

public class CDTest  {
    public void main (String[] args) {
        C cd = new D();
        D dd = (D) cd;
        ****
    }
}
```

(1) بالنسبة لكلّ واحد من الأوامر 1-3 التي أمامك، اذكر إذا كان الأمر سليمًا إذا بَدَّلَ السطر **** في العمليّة main. إذا كان سليمًا، اكتب المخرَج، وإذا لم يكن سليمًا، علّل لماذا الأمر ليس سليمًا.

1. dd.dolt (cd);
2. cd.dolt (dd);
3. cd.dolt (cd);

(2) للفئة C أُضيفت العمليّة التي أمامك:

```
public void dolt (Object o) {
    System.out.println("co") ; }
```

بالنسبة لكلّ واحد من الأوامر 1-3 التي أمامك، اذكر إذا كان الأمر سليمًا إذا بَدَّلَ السطر **** في العمليّة main. إذا كان سليمًا، اكتب المخرَج، وإذا لم يكن سليمًا، علّل لماذا الأمر ليس سليمًا.

1. dd.dolt (cd);
2. cd.dolt (dd);
3. cd.dolt (cd);

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 19 /

ب. معطاة الفئتان AA ، BB :

```
public class AA {
    public AA() { System.out.println ("In AA"); }
    public AA(int n) {
        this();
        System.out.println ("In AA");
        System.out.println (n * 2);
    }
}
```

```
public class BB extends AA {
    public BB() { System.out.println ("In BB"); }
    public BB(double k) {
        System.out.println ("In BB" + (int)k);
    }
    public BB(int k) {
        super(k * 2);
        for (int i = 1; i < k; i ++)
            System.out.println ("In BB");
    }
}
```

أمامك الفئة Run :

```
public class Run {
    public static void main (String[] args) {
        ****
    }
}
```

في الفئة Run في العمليّة الرئيسيّة main المكان المُشار إليه بِـ**** يمثّل سطر كود ما.
أمامك ثلاثة مخرَجات لا علاقة بينها. اذكر بالنسبة لكلّ مخرَج، ما هو سطر الكود الذي يُنتِجه.

(1)
```
In AA
In BB
```

(2)
```
In AA
In BB7
```

(3)
```
In AA
In AA
12
In BB
In BB
```

ملاحظة: عليك الاعتماد على العمليّات البنائيّة المعرَّفة في الفئتين AA وَ BB . لا تُضِف عمليّات وطباعات.

/ يتبع في صفحة 20 /

برمجة موجّهة كائنات

السؤالان 16-17 معدّان للّذين يكتبون بلغة C# .

16. أمامك الفئات A ، B ، C ، D .

```
public class A {

    public static int count = 0;
    protected int num;

    public A() {
        this.num = 3;
        count ++;
    }

    public int GetNum() {
        return this.num;

    }

    public void SetNum(int num) {
        this.num = num;
    }

    public void Foo(int num) {
        this.num = num;
        num ++;
    }

    public bool IsEqual(A other) {
        return (this.num == other.num);
    }

    public virtual void Func() {
        Console.WriteLine("I am A");
    }
}
```

```
public class B : A {

    public override void Func() {
        Console.WriteLine("I am B");
    }

    public bool IsEqual(B other) {
        return (this == other);
    }
}

-----------------------------------------------------------

public class C : A { }

-----------------------------------------------------------

public class D {
    private A a;

    public D(A a) {
        this.a = a;
    }

    public void Func() {
        (this.a).Func();
    }

    public void IsB() {
        Console.WriteLine((this.a) is B);
    }
}
```

/ يتبع في صفحة 21 /

أ. ارسم الخارطة الهرميّة بين الفئات A ، B ، C ، D .

يجب الإشارة إلى توريث بواسطة السهم ◁—— وإلى احتواء بواسطة الإشارة ◆—— .

ب. ارسم الكائنات التي تكوّنت في العمليّة Main في الفئة Tester التي أمامك، واكتب المخرَج.

```
public class Tester {
    public static void Main(string[] args) {
        A y1 = new A();
        A y2 = new A();
        y1.Foo(y1.GetNum());
        Console.WriteLine(y1.IsEqual(y2));
        B y3 = new B();
        A y4 = new B();
        Console.WriteLine(y3.IsEqual(y4));
        Console.WriteLine(y4.IsEqual(y3));
        B y5 = y3;
        y5.SetNum(0);
        Console.WriteLine(y3.IsEqual(y5));
        C y6 = new C();
        Console.WriteLine(((B)y4).IsEqual(y6));
        D d1 = new D(y4);
        D d2 = new D(y6);
        Console.WriteLine(B.count);
        d1.Func();
        d2.Func();
        d1.IsB();
        d2.IsB();
    }
}
```

/ يتبع في صفحة 22 /

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

أ. معطاة الفئات التالية:

```
public class C {
    public virtual void Dolt(D d) { Console.WriteLine("a"); }
}

public class D : C {
    public override void Dolt(D d) { Console.WriteLine("b"); }
    public void Dolt(C c) { Console.WriteLine("x"); }
}

public class CDTest  {
    public void Main (string[] args) {
        C cd = new D();
        D dd = (D) cd;
        ****
    }
}
```

(1) بالنسبة لكلّ واحد من الأوامر 1-3 التي أمامك، اذكر إذا كان الأمر سليمًا إذا بَدَّلَ السطر **** في العمليّة Main. إذا كان سليمًا، اكتب المخرَج، وإذا لم يكن سليمًا، علّل لماذا الأمر ليس سليمًا.

1. dd.Dolt (cd);
2. cd.Dolt (dd);
3. cd.Dolt (cd);

(2) للفئة C أُضيفت العمليّة التي أمامك:

```
public void Dolt (Object o) {
    Console.WriteLine("o"); }
```

بالنسبة لكلّ واحد من الأوامر 1-3 التي أمامك، اذكر إذا كان الأمر سليمًا إذا بَدَّلَ السطر **** في العمليّة Main. إذا كان سليمًا، اكتب المخرَج، وإذا لم يكن سليمًا، علّل لماذا الأمر ليس سليمًا.

1. dd.Dolt (cd);
2. cd.Dolt (dd);
3. cd.Dolt (cd);

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 23 /

ب. معطاة الفئتان AA ، BB :

```
public class AA {
    public AA() { Console.WriteLine ("In AA"); }
    public AA(int n) : this() {
        Console.WriteLine ("In AA");
        Console.WriteLine (n * 2);
    }
}
```

```
public class BB : AA {
    public BB() { Console.WriteLine ("In BB"); }
    public BB(double k) {
        Console.WriteLine ("In BB" + (int)k);
    }
    public BB(int k) : base(k * 2){
        for (int i = 1; i < k; i++)
            Console.WriteLine ("In BB");
    }
}
```

أمامك الفئة Run :

```
public class Run {
    public static void Main (string[] args) {
        ****
    }
}
```

في الفئة Run في العمليّة الرئيسيّة Main المكان المُشار إليه بِـ**** يمثّل سطر كود ما.
أمامك ثلاثة مخرَجات لا علاقة بينها. اذكر بالنسبة لكلّ مخرَج، ما هو سطر الكود الذي يُنتِجه.

(1)
```
In AA
In BB
```

(2)
```
In AA
In BB7
```

(3)
```
In AA
In AA
12
In BB
In BB
```

ملاحظة: عليك الاعتماد على العمليّات البنائيّة المعرَّفة في الفئتين AA وَ BB . لا تُضِف عمليّات وطباعات.

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.