מדינת ישראל
משרד החינוך

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

مدعي المحسب ב'

2 יחידות לימוד (השלמה ל־5 יח"ל)

הוראות לנבחן

א. משך הבחינה: שלוש שעות.
ב. מבנה השאלון ומפתח ההערכה:
בשאלון זה שני פרקים.
פרק ראשון: בפרק זה ארבע שאלות, ומהן יש לענות על שתיים.
(25x2) — 50 נק'
פרק שני: בפרק זה שאלות בארבעה מסלולים שונים. ענה על שאלות רק במסלול שלמדת, לפי ההוראות בקבוצת השאלות במסלול זה.
(25x2) — 50 נק'
סה"כ — 100 נק'

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

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

دولة إسرائيل
وزارة التّربية والتّعليم

نوع الامتحان: أ. بجروت للمدارس الثّانويّة
ب. بجروت للممتحَنين الخارجيّين
موعد الامتحان: صيف 2016، **الموعد "ب"**
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.

ب 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)

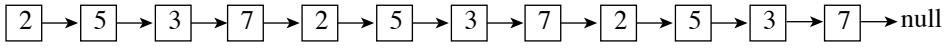
انتبه: في كلّ سؤال يُطلب فيه تطبيق، بإمكانك استعمال عمليّات الفئات قائمة ودور وراصّة وشجرة بينارية وحلقة رابطة، بدون تطبيقها.
إذا استعملتَ عمليّات إضافيّة، عليك تطبيقها.

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

1. تسمّى القائمة L **مثلّثيّة** إذا كانت تحقّق الشروط التالية:

* القائمة ليست فارغة.
* عدد الحدود فيها يقسم على 3 بدون باقٍ.
* الحدود في الثلث الأوّل من القائمة تحوي نفس القيَم التي تحويها الحدود في الثلث الثاني من القائمة ونفس القيَم التي تحويها الحدود في الثلث الثالث من القائمة. القيَم مرتَّبة بنفس الترتيب في كلّ واحد من الأثلاث.

مثلاً: القائمة L1 التي أمامك هي قائمة **مثلّثيّة** بطول 12.



اكتب بلغة C# أو بلغة Java عمليّة خارجيّة تتلقّى قائمة L حدودها من نمط صحيح.
إذا كانت L قائمة **مثلّثيّة**، تعيد العمليّة true.
خلاف ذلك – تعيد العمليّة false.

/ يتبع في صفحة 3 /

2. **انتبه:** لهذا السؤال صيغتان: الأولى بلغة Java (في الصفحتين 3-4)، والثانية بلغة C# (في الصفحتين 5-6). اعمل حسب اللغة التي تعلّمتَها.

للّذين يحلّون بلغة Java

أمامك العمليّتان sod وَ what اللتان تتلقّيان المصفوفة a التي حدودها من نمط صحيح والمرتَّبة بترتيب تصاعديّ، وتتلقّيان أيضًا عددًا صحيحًا k .

للعمليّتين نفس ادّعاء الخروج.

```
public static boolean sod(int[] a , int k)
{
    for (int i = 0; i < a.length−1; i++)
    {
        int j = i+1;
        while (j < a.length)
        {
            if (a[i] + a[j] == k)
                return true;
            j++;
        }
    }
    return false;
}

public static boolean what(int[] a , int k)
{
    int left = 0 , right = a.length−1;
    while (left < right)
    {
        if (a[left] + a[right] == k)
            return true;
        if (a[left] + a[right] < k)
            left++;
        else
            right−−;
    }
    return false;
}
```

(انتبه: تكملة السؤال في الصفحة التالية.) /يتبع في صفحة 4/

معطاة المصفوفة a :

a	2	4	7	12	18

أ. تتبّع بواسطة جدول متابعة تنفيذ العمليّة sod بالنسبة للمصفوفة المعطاة a ، والعدد k = 11 . اكتب القيمة التي تُعاد.
يجب أن يشمل جدول المتابعة أعمدة بالنسبة لِـ : i, j, a[i], a[j] ، وعمودًا إضافيًّا يُذكر فيه إذا كان الشرط الذي في الأمر if يتحقّق أم لا يتحقّق.

ب. تتبّع بواسطة جدول متابعة تنفيذ العمليّة sod بالنسبة للمصفوفة المعطاة a ، والعدد k = 10 . اكتب القيمة التي تُعاد.
يجب أن يشمل جدول المتابعة الأعمدة التي تمّ تفصيلها في البند "أ".

جـ. ما هو ادّعاء خروج العمليّة sod ؟

د. ما هي تعقيدات زمن تشغيل العمليّة sod ؟ علّل إجابتك.

هـ. تتبّع بواسطة جدول متابعة تنفيذ العمليّة what بالنسبة للمصفوفة المعطاة a ، والعدد k = 11 . اكتب القيمة التي تُعاد.
يجب أن يشمل جدول المتابعة أعمدة بالنسبة لِـ : left, right, a[left], a[right] ، وعمودًا إضافيًّا لكلّ واحد من الأمرين if ، بحيث يُذكَر في كلّ واحد من العمودين إذا كان الشرط الذي في الأمر if يتحقّق أم لا يتحقّق.

و. ما هي تعقيدات زمن تشغيل العمليّة what ؟ علّل إجابتك.

ز. أيّ من العمليّتين – sod أم what – أكثر نجاعة؟ علّل إجابتك.

حـ. تمّ تغيير ادّعاء دخول العمليّتين sod وَ what بحيث يمكن نقل مصفوفة a غير مرتَّبة إليهما.

(1) هل يتغيّر ادّعاء خروج العمليّة sod ؟ علّل إجابتك.

(2) هل يتغيّر ادّعاء خروج العمليّة what ؟ علّل إجابتك.

/ يتبع في صفحة 5 /

للّذين يحلّون بلغة C#

أمامك العمليّتان Sod وَ What اللتان تتلقّيان المصفوفة a التي حدودها من نمط صحيح والمرتَّبة بترتيب تصاعديّ، وتتلقّيان أيضًا عددًا صحيحًا k .
للعمليّتين نفس ادّعاء الخروج.

```
public static bool Sod(int[] a , int k)
{
    for (int i = 0; i < a.Length-1; i++)
    {
        int j = i+1;
        while (j < a.Length)
        {
            if (a[i] + a[j] == k)
                return true;
            j++;
        }
    }
    return false;
}

public static bool What(int[] a , int k)
{
    int left = 0 , right = a.Length-1;
    while (left < right)
    {
        if (a[left] + a[right] == k)
            return true;
        if (a[left] + a[right] < k)
            left++;
        else
            right--;
    }
    return false;
```

(انتبه: **تكملة السؤال في الصفحة التالية.**) / يتبع في صفحة 6 /

معطاة المصفوفة a :

a	2	4	7	12	18

أ. تتبّع بواسطة جدول متابعة تنفيذ العمليّة Sod بالنسبة للمصفوفة المعطاة a ، والعدد k = 11 . اكتب القيمة التي تُعاد.
يجب أن يشمل جدول المتابعة أعمدة بالنسبة لـ : i, j, a[i], a[j] ، وعمودًا إضافيًّا يُذكر فيه إذا كان الشرط الذي في الأمر if يتحقّق أم لا يتحقّق.

ب. تتبّع بواسطة جدول متابعة تنفيذ العمليّة Sod بالنسبة للمصفوفة المعطاة a ، والعدد k = 10 . اكتب القيمة التي تُعاد.
يجب أن يشمل جدول المتابعة الأعمدة التي تمّ تفصيلها في البند "أ".

جـ. ما هو ادّعاء خروج العمليّة Sod ؟

د. ما هي تعقيدات زمن تشغيل العمليّة Sod ؟ علّل إجابتك.

هـ. تتبّع بواسطة جدول متابعة تنفيذ العمليّة What بالنسبة للمصفوفة المعطاة a ، والعدد k = 11 . اكتب القيمة التي تُعاد.
يجب أن يشمل جدول المتابعة أعمدة بالنسبة لـ : left, right, a[left], a[right] ، وعمودًا إضافيًّا لكلّ واحد من الأمرين if ، بحيث يُذكَر في كلّ واحد من العمودين إذا كان الشرط الذي في الأمر if يتحقّق أم لا يتحقّق.

و. ما هي تعقيدات زمن تشغيل العمليّة What ؟ علّل إجابتك.

ز. أيّ من العمليّتين – Sod أم What – أكثر نجاعة؟ علّل إجابتك.

حـ. تمّ تغيير ادّعاء دخول العمليّتين Sod وَ What بحيث يمكن نقل مصفوفة a غير مرتَّبة إليهما.

(1) هل يتغيّر ادّعاء خروج العمليّة Sod ؟ علّل إجابتك.

(2) هل يتغيّر ادّعاء خروج العمليّة What ؟ علّل إجابتك.

/ يتبع في صفحة 7 /

3. **أ.** اكتب عمليّة خارجيّة تتلقّى شجرة بينارية t ليست فارغة، كلّ عقدة فيها تحوي راصّة لأعداد صحيحة وأكبر من 0 .

تُعيد العمليّة راصّة. بالنسبة لكلّ عقدة في الشجرة t يُدخَل حدّ إلى الراصّة المُعادة على النحو التالي:

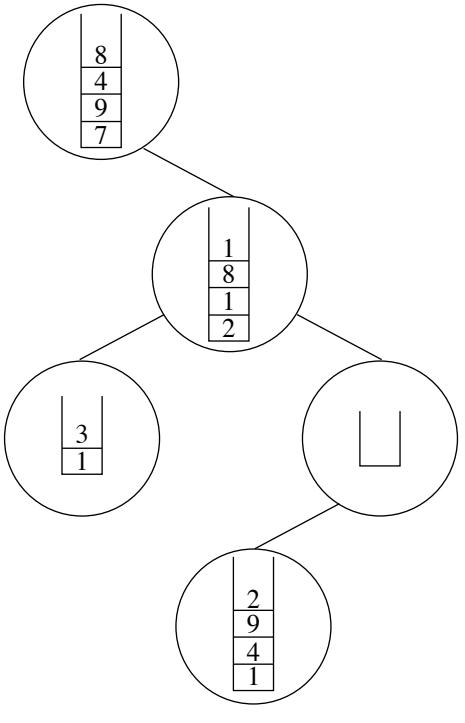
إذا كان في الراصّة التي في العقدة حدّ واحد، تُدخَل قيمته إلى الراصّة المُعادة. إذا كان في الراصّة التي في العقدة حدّان، يُدخَل مجموعهما إلى الراصّة المُعادة. إذا كانت الراصّة التي في العقدة فارغة، يُدخَل 0 إلى الراصّة المُعادة. في كلّ حالة أخرى يُدخَل إلى الراصّة المُعادة مجموع ثلاثة الحدود العليا للراصّة التي في العقدة.

ترتيب الحدود في الراصّة المُعادة يكون حسب المسح بترتيب ضمنيّ (inorder) لعُقَد الشجرة t .

لا حاجة لحفظ محتوى الراصّات التي في عُقَد الشجرة t .

على سبيل المثال، بالنسبة للشجرة t :

t



الراصّة المُعادة:

0
15
10
4
21

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

/ يتبع في صفحة 8 /

4. **انتبه**: لهذا السؤال صيغتان: الأولى بلغة Java (في الصفحتين 8-9)، والثانية بلغة C# (في الصفحتين 10-11). اعمل حسب اللغة التي تعلّمتَها.

للّذين يحلّون بلغة Java

اللعبة "إلعب من فضلك" هي لعبة يلعبها لاعب واحد في مرحلتين. يحصل اللاعب على 52 ورقة شدّة مخلوطة ومختلفة. يوجد على كلّ ورقة عدد بين 1 وَ 13، وإحدى صور من أربع صور. الصور ممثَّلة بواسطة الأعداد 1 حتّى 4.

في المرحلة الأولى يوزّع اللاعب الأوراق إلى أربع كومات حسب صورة الورقة، بحيث تكون في كلّ كومة أوراق عليها نفس الصورة. الأوراق موضوعة الواحدة فوق الأخرى.

في المرحلة الثانية يُجري اللاعب قرعة لعدد بين 1 وَ 4 يمثّل صورة ورقة. يتوجّه اللاعب إلى الكومة التي على جميع أوراقها تتواجد الصورة الممثَّلة بواسطة العدد الذي خرج في القرعة. يأخذ اللاعب الورقة التي في رأس الكومة، وينقلها إلى كومة خامسة.

يتمّ تنفيذ هذه المرحلة حتّى تنتهي اللعبة.

تنتهي اللعبة عندما يخرج في القرعة عدد يمثّل كومة لا توجد فيها أوراق.

الفوز هو حالة يكون فيها مجموع الأعداد التي على الأوراق في الكومة الخامسة يقسم على 100 بدون باقٍ.

بعد أن تنتهي اللعبة يفحصون إذا تحقّق فوز.

لغرض تطبيق اللعبة عُرّفت الفئات: **Card** التي تمثّل ورقة واحدة، وَ **Deck** التي تمثّل الكومات الخمس اللازمة في اللعبة، وَ **Test** التي تُدير اللعبة.

كما وكُتبت في الفئة **Test** العمليّة public static boolean game(Card[] cards)، التي تتلقّى مصفوفة من 52 ورقة مختلفة مرتَّبة عشوائيًّا. العمليّة تُدير اللعبة حتّى نهايتها. تُعيد العمليّة true إذا انتهت اللعبة بفوز، وخلاف ذلك – تُعيد false.

أمامك الفئة **Card** التي تمثّل ورقة عليها عدد (value) بين 1 وَ 13 وصورة (shape) بين 1 وَ 4.

```
public class Card
{
    private int value;
    private int shape;

    public Card (int value, int shape)
    {
        this.value = value;
        this.shape = shape;
    }
    public int getValue () {   return this.value;    }
    public int getShape () {   return this.shape;    }
}
```

/ يتبع في صفحة 9 /

أ. i اكتب بلغة Java عنوان الفئة **Deck** وصفاتها.

ii اكتب بلغة Java في الفئة **Deck** عناوين العمليّات التالية:

- عمليّة بنائيّة تعرّف بأن تكون الكومات الخمس فارغة.
- عمليّة insert تتلقّى ورقة وتُضيفها إلى الكومة الصحيحة حسب المرحلة الأولى في اللعبة.
- عمليّة move تُجري قرعة لعدد يمثّل صورة. إذا كانت كومة الأوراق التي خرجت صورتها في القرعة فارغة، تُعيد العمليّة false. خلاف ذلك – تنقل العمليّة إلى الكومة الخامسة الورقة التي في رأس الكومة التي خرج عددها في القرعة وتُعيد true.
- عملية sum تُعيد المجموع الحاليّ للأعداد التي على الأوراق في الكومة الخامسة.

انتبه: في هذا البند الفرعيّ، لا حاجة لتطبيق العمليّات.

ب. طبِّق بلغة Java العمليّة البنائيّة في الفئة **Deck** .

جـ. طبِّق بلغة Java العمليّة sum في الفئة **Deck** .

د. طبِّق بلغة Java العمليّة game في الفئة **Test**.

بإمكانك استعمال العمليّتين insert وَ move للفئة Deck، بدون تطبيقهما.
إذا استعملتَ عمليّات أخرى، عليك تطبيقها.

/ يتبع في صفحة 10 /

للّذين يحلّون بلغة C#

اللعبة "إلعب من فضلك" هي لعبة يلعبها لاعب واحد في مرحلتين. يحصل اللاعب على 52 ورقة شدّة مخلوطة ومختلفة. يوجد على كلّ ورقة عدد بين 1 وَ 13 ، وإحدى صور من أربع صور.
الصور ممثَّلة بواسطة الأعداد 1 حتّى 4 .

في المرحلة الأولى يوزّع اللاعب الأوراق إلى أربع كومات حسب صورة الورقة، بحيث تكون في كلّ كومة أوراق عليها نفس الصورة. الأوراق موضوعة الواحدة فوق الأخرى.

في المرحلة الثانية يُجري اللاعب قرعة لعدد بين 1 وَ 4 يمثّل صورة ورقة. يتوجّه اللاعب إلى الكومة التي على جميع أوراقها تتواجد الصورة الممثَّلة بواسطة العدد الذي خرج في القرعة.
يأخذ اللاعب الورقة التي في رأس الكومة، وينقلها إلى كومة خامسة.
يتمّ تنفيذ هذه المرحلة حتّى تنتهي اللعبة.

تنتهي اللعبة عندما يخرج في القرعة عدد يمثّل كومة لا توجد فيها أوراق.

الفوز هو حالة يكون فيها مجموع الأعداد التي على الأوراق في الكومة الخامسة يقسم على 100 بدون باقٍ.

بعد أن تنتهي اللعبة يفحصون إذا تحقّق فوز.

لغرض تطبيق اللعبة عُرّفت الفئات: **Card** التي تمثّل ورقة واحدة، وَ **Deck** التي تمثّل الكومات الخمس اللازمة في اللعبة، وَ **Test** التي تُدير اللعبة.

كما وكُتبت في الفئة **Test** العمليّة public static bool Game(Card[] cards) ، التي تتلقّى مصفوفة من 52 ورقة مختلفة مرتَّبة عشوائيًّا. العمليّة تُدير اللعبة حتّى نهايتها. تُعيد العمليّة true إذا انتهت اللعبة بفوز، وخلاف ذلك – تُعيد false.

أمامك الفئة **Card** التي تمثّل ورقة عليها عدد (value) بين 1 وَ 13 وصورة (shape) بين 1 وَ 4.

```
public class Card
{
    private int value;
    private int shape;

    public Card (int value, int shape)
    {
        this.value = value;
        this.shape = shape;
    }
    public int GetValue () {   return this.value;    }
    public int GetShape () {   return this.shape;    }
}
```

/ يتبع في صفحة 11 /

أ. i اكتب بلغة C# عنوان الفئة **Deck** وصفاتها.

ii اكتب بلغة C# في الفئة **Deck** عناوين العمليّات التالية:

- عمليّة بنائيّة تعرّف بأن تكون الكومات الخمس فارغة.
- عمليّة Insert تتلقّى ورقة وتُضيفها إلى الكومة الصحيحة حسب المرحلة الأولى في اللعبة.
- عمليّة Move تُجري قرعة لعدد يمثّل صورة. إذا كانت كومة الأوراق التي خرجت صورتها في القرعة فارغة، تُعيد العمليّة false. خلاف ذلك – تنقل العمليّة إلى الكومة الخامسة الورقة التي في رأس الكومة التي خرج عددها في القرعة وتُعيد true.
- عمليّة Sum تُعيد المجموع الحاليّ للأعداد التي على الأوراق في الكومة الخامسة.

انتبه: في هذا البند الفرعيّ، لا حاجة لتطبيق العمليّات.

ب. طبِّق بلغة C# العمليّة البنائيّة في الفئة **Deck**.

جـ. طبِّق بلغة C# العمليّة Sum في الفئة **Deck**.

د. طبِّق بلغة C# العمليّة Game في الفئة **Test**.

بإمكانك استعمال العمليّتين Insert وَ Move للفئة Deck، بدون تطبيقهما.
إذا استعملتَ عمليّات أخرى، عليك تطبيقها.

/ يتبع في صفحة 12 /

الفصل الثاني (50 درجة)

في هذا الفصل أسئلة في أربعة مسارات:
أنظمة حاسوب وأسمبلي، ص 12-16.
مقدّمة لبحث الأداءات، ص 17-27.
موديلات حسابيّة، ص 28-29.
برمجة موجّهة كائنات بلغة Java، ص 30-40؛ برمجة موجّهة كائنات بلغة #C ، ص 41-51.

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 طريقة لحساب تربيع عدد صحيح وأكبر من 0 :

$1^2 = 1$
$2^2 = 1 + 3$
$3^2 = 1 + 3 + 5$
$4^2 = 1 + 3 + 5 + 7$
$5^2 = 1 + 3 + 5 + 7 + 9$
وهكذا.

في AL مخزون عدد صحيح، أكبر من 0 وأصغر من 16 .
أمامك قطعة برنامج بلغة أسمبلي، تحسب تربيع العدد المخزون في AL حسب الطريقة الموصوفة، وتخزن النتيجة في AX .
حُذفت من قطعة البرنامج أسطر كاملة أو أجزاء أسطر في الأماكن المشار إليها بالأرقام i-iv.
انسخ أرقام الأسطر المشار إليها إلى دفترك، واكتب بجانب كلّ واحد من الأرقام سطر الأمر بما في ذلك الأجزاء الناقصة، كي تنفّذ قطعة البرنامج المطلوب.

```
                    MOV     BX , 0
i                   ------  ------
                    MOV     DX , 1
                    MOV     CX , AX
                    DEC     CX
ii                  ------  ------
            A1:     ADD     BX , DX
iii                 ADD     DX , -------
                    LOOP    A1
            A2:     ADD     BX , DX
iv                  MOV     AX , -------
            A3:     NOP
```

/ يتبع في صفحة 13 /

ب. (لا علاقة له مع البند "أ".)

عُرِّفت المعطيات في مقطع المعطيات على النحو التالي:

```
T    DB    55 , 90 , 110 , 1
```

انتبه: الأعداد هي أعداد عشريّة.

أمامك قطعة برنامج بلغة أسمبلي هدفها حساب مجموع الحدود في المصفوفة T.

```
        MOV     CX , 4
        XOR     AX , AX
        LEA     BX , T
AGAIN:  ADD     AL , [BX]
        INC     BX
        LOOP    AGAIN
```

قطعة البرنامج لا تنفّذ المطلوب.

(1) تتبّع بواسطة جدول متابعة تنفيذ قطعة البرنامج، واكتب ماذا يكون محتوى AL، وماذا يكون محتوى عَلَم وضع المرحَّل (דגל הנשא) وعَلَم الفيض الزائد (דגל הגלישה) في نهاية تنفيذ قطعة البرنامج.

(2) غيِّر قطعة البرنامج بحيث تحسب مجموع الحدود في المصفوفة المعطاة T.
انسخ قطعة البرنامج بعد التغيير إلى دفترك.

/ يتبع في صفحة 14 /

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

أ. في مقطع المعطيات عُرِّفت المصفوفة:

```
ARR  DB  5  DUP (?)
```

أمامك 4 قطع iv-i بلغة أسمبلي هدفها ابتداء جميع خلايا المصفوفة ARR بِـ 0.
تتبّع بواسطة جدول متابعة تنفيذ كلّ واحدة من القطع iv-i التي أمامك، وحدّد إذا كانت تنفّذ المطلوب أم لا تنفّذ المطلوب.

i

```
        MOV     SI , 0
        MOV     CX , 4
A1:     MOV     ARR[SI] , 0
        INC     SI
        LOOP    A1
```

---

ii

```
        MOV     CX , 5
        LEA     BX , ARR
        MOV     AL , 0
A1:     MOV     [BX] , AL
        INC     BX
        LOOP    A1
```

---

iii

```
        MOV     BX , 5
        DEC     BX
A1:     MOV     ARR[BX] , 0
        DEC     BX
        JNZ     A1
```

---

iv

```
        MOV     DI , 0
A1:     MOV     ARR[DI] , 0
        INC     DI
        CMP     DI , 5
        JC      A1
```

/ يتبع في صفحة 15 /

ب. (لا علاقة له مع البند "أ" .)

أمامك قطعة برنامج بلغة أسمبلي :

انتبه : المعطيات هي بأساس سداسيّ عشريّ .

```
START:  MOV     AX , C83BH
        MOV     BX , A89CH
        SHL     AX , 1
        OR      AL , 33H
        NOT     BL
        ADD     AX , BX
```

تتبّع بواسطة جدول متابعة تنفيذ قطعة البرنامج .

فصّل في جدول المتابعة محتوى CF ، SF ، ZF ، BX ، AX في كلّ مرحلة .

/ يتبع في صفحة 16 /

7. عُرِّفت مصفوفتان في مقطع المعطيات.
المصفوفة NUMBERS بِكِبَر 100 بايت، التي تحوي أعدادًا سداسيّة عشريّة بين 0 وَ 0FF،
والمصفوفة APP ابتداءً من العنوان 2500H، بِكِبَر 256 بايت، تحوي أصفارًا.
اكتب بلغة أسمبلي قطعة برنامج، تمرّ مرّة واحدة فقط على جميع حدود المصفوفة NUMBERS،
وتسجّل في المكان الـ i في المصفوفة APP عدد المرّات التي يظهر فيها العدد i في
المصفوفة NUMBERS.

8. خُزِّن في الخازن AX عدد بينارِيّ. اكتب قطعة برنامج بلغة أسمبلي، تُعوِّض في الخازن BL
عدد المرّات التي يظهر فيها التسلسل 1011 في العدد البينارِيّ الذي في الخازن AX.
مثال: إذا كان في الخازن AX مخزونًا العدد البينارِيّ: 0110110010110110 ،
يُعوَّض في الخازن BL العدد 3.

/ يتبع في صفحة 17 /

مقدّمة لبحث الأداءات

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

9. معطاة مسألة تخطيط خطّيّ:

$$\text{Max}\{z = 5x_1 - x_2\}$$

تخضع للاضطرارات التالية:

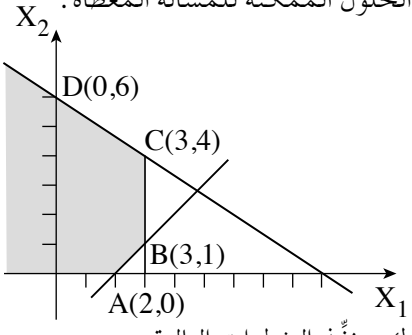
$$2x_1 + 3x_2 \le 18$$
$$x_1 - x_2 \le 2$$
$$x_1 \le 3$$
$$x_2 \ge 0$$

x_1 ليس مقيّدًا بإشارة.

معطى أيضًا أنّ الحلّ $(3,1)$ هو حلّ أمثل للمسألة.

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

أ. أمامك رسم لمجال الحلول الممكنة للمسألة المعطاة.



انسخ الرسم إلى دفترك، ونفِّذ الخطوات التالية:

الخطوة 1: أضِف إلى الرسم الذي في دفترك مسقط ارتفاع دالّة الهدف بالنسبة لـ $z = 5$.
احسب إحداثيّات نقاط تقاطع هذا المسقط مع المحورين x_1 وَ x_2، وأشر إليها في الرسم.

الخطوة 2: أضِف إلى الرسم الذي في دفترك مسقط ارتفاع دالّة الهدف بالنسبة لـ $z = 15$.
احسب إحداثيّات نقاط تقاطع هذا المسقط مع المحورين x_1 وَ x_2، وأشِر إليها في الرسم.

الخطوة 3: بواسطة سهم، أشِر في الرسم الذي في دفترك، إلى اتّجاه تصاعد دالّة الهدف.

هل تُبيِّن الخطوات 1-3 أنّ الحلّ $(1,3)$ هو الحلّ الأمثل الوحيد؟ علّل إجابتك.

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

ب. نضيف إلى المسألة المعطاة في بداية السؤال الاضطرار:

$$x_1 - x_2 \geq 2$$

ارسم في دفترك هيئة محاور جديدة، وأشِر فيها فقط إلى مجال الحلول الممكنة الذي ينتج بعد إضافة هذا الاضطرار.

جـ. نضيف إلى المسألة المعطاة في بداية السؤال اضطرارًا جديدًا معيّنًا، بحيث يكون مجال الحلول الممكنة بعد إضافته هو القطعة التي بين النقطتين (4 , 3) وَ (1 , 3) .

هل يتغيّر الحلّ الأمثل المعطى (1 , 3) ؟ علّل إجابتك.

د. نضيف إلى المسألة المعطاة في بداية السؤال اضطرارًا جديدًا:

$$3x_1 + 2x_2 \leq 14$$

ادّعى أحد الطلّاب أنّ الحلّ الأمثل المعطى (1 , 3) لن يتغيّر.

هل هذا الطالب على حقّ؟ علّل إجابتك بدون أن ترسم من جديد مجال الحلول الممكنة بعد إضافة هذا الاضطرار.

هـ. نغيّر فقط دالّة هدف المسألة التي في بداية السؤال إلى $z = 3x_1$.

أمامك الأقوال i-iv ، واحد منها فقط صحيح. انسخ القول الصحيح إلى دفترك، وعلّل اختيارك.

i الحلّ الأمثل الوحيد سيكون (1 , 3).

ii الحلّ الأمثل الوحيد سيكون (4 , 3).

iii الحلّ الأمثل الجديد سيكون:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \lambda\begin{pmatrix} 3 \\ 1 \end{pmatrix} + (1-\lambda)\begin{pmatrix} 3 \\ 4 \end{pmatrix}$$

عندما $0 \leq \lambda \leq 1$

iv الحلّ الأمثل لن يكون محصورًا.

/ يتبع في صفحة 19 /

10. في هذا السؤال ستّة بنود، "أ – و"، لا علاقة بينها. أجب عن جميع البنود.

أ. أمامك مسألة نقل فيها مصدران وَ 4 غايات. تكاليف الوحدة من كلّ مصدر إلى كلّ غاية معطاة في الجدول الذي أمامك.

المصادر	الغايات				العرض
	1	2	3	4	
1	8	9	4	6	100
2	14	12	13	8	80
الطلب	10	40	30	70	

حسب الجدول المعطى لا يمكن تفعيل طريقة الزاوية الشماليّة – الغربيّة لإيجاد حلّ ممكن لمسألة النقل. فسّر لماذا.

ب. في الجدول الذي أمامك معطى جزء من حلّ أساسيّ ممكن لمسألة نقل:

$x_{12} = 2$ ، $x_{11} = 10$.

المصادر	الغايات			العرض
	1	2	3	
1	4 (10)	5 (2)	10	12
2	6	3	6	12
3	6	2	6	6
الطلب	10	10	10	

انسخ الجدول إلى دفترك، وأكمله حسب طريقة الزاوية الشماليّة – الغربيّة.

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

جـ. في الجدول الذي أمامك معطى حلّ أساسيّ ممكن لمسألة نقل. القيمة التي حُدِّدت لِـ u_1 هي 0. انسخ الجدول إلى دفترك، واحسب قيمة v_1 ، v_2 ، v_3 ، u_2 ، u_3 ، واكتب القيَم التي حسبتَها في الأماكن الملائمة في الجدول.

المصادر	**الغايات**			**العرض**	u_i
	1	2	3		
1	4 (10)	5 (2)	10	12	0
2	6	3 (2)	6 (10)	12	
3	6	2 (6)	6	6	
الطلب	10	10	10		
v_j					

/ يتبع في صفحة 21 /

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

المصادر	**الغايات**			**العرض**	u_i
	1	2	3		
1	4 (20)	5 (4)	10	24	0
2	2	6 (16)	3 (4)	20	1
3	6	2	6 (8)	8	4
الطلب	20	20	12		
v_j	4	5	2		

هل الحلّ هو أمثل؟ علّل إجابتك.

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

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

المصادر	الغايات			العرض	u_i
	1	2	3		
1	10 (20)	25	30	20	10
2	10 (30)	22	14 (20)	50	10
3	16	20 (40)	20 (20)	60	16
الطلب	50	40	40		
v_j	0	4	4		

أمامك الأقوال i-iv ، واحد منها فقط صحيح. انسخ القول الصحيح إلى دفترك، وعلّل اختيارك.

i الحلّ المعطى ليس حلّاً ممكنًا.

ii الحلّ المعطى هو حلّ أساسيّ ممكن لكنّه ليس أمثل.

iii الحلّ المعطى هو حلّ أمثل وحيد.

iv الحلّ المعطى هو حلّ أمثل لكنّه ليس حلّاً أمثل وحيدًا.

/ يتبع في صفحة 23 /

و. في الجدول الذي أمامك معطى جزء من حلّ أساسيّ ممكن لمسألة نقل تمّ حسابه حسب طريقة الزاوية الشماليّة – الغربيّة، ومعطاة قيَم v_1 ، v_2 ، v_3 ، u_1 ، u_2 ، u_3 التي تمّ حسابها حسب هذا الحلّ.

المصادر	**الغايات**			**العرض**	u_i
	1	2	3		
1	12 (20)	15	17	20	2
2	10	18 (10)	14	10	0
3	20	10 (5)	18 (10)	15	−8
الطلب	20	15	10		
v_j	10	18	26		

ما الذي يجب إضافته إلى الجدول للحصول على حلّ أساسيّ ممكن يلائم جميع المعطيات التي في الجدول؟

انسخ الإجابة الصحيحة إلى دفترك من بين الإمكانيّات i-iv، وعلّل إجابتك.

i $X_{12} = 0$.

ii $X_{21} = 0$.

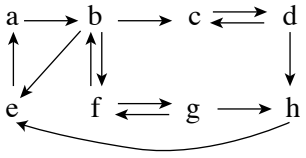
iii $X_{13} = 0$.

iv $X_{31} = 0$.

/ يتبع في صفحة 24 /

11. في هذا السؤال ثلاثة بنود، "أ–جـ"، لا تتعلّق ببعضها البعض. أجب عن جميع البنود.

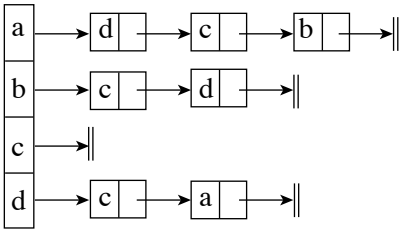
أ. أمامك رسم بياني موجَّه G .



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

ii هل الرسم البيانيّ المعطى هو رسم بيانيّ قابل للربط موجَّه بقوّة (قابل للربط جيّدًا)؟
علّل إجابتك.

ب. $G = (V , E)$ هو رسم بياني موجَّه، عندما V هي مجموعة الرؤوس في الرسم البيانيّ وَ E هي مجموعة الأقواس في الرسم البيانيّ. G ممثَّل بواسطة قائمة المُجاوَرات التي أمامك:



i فَعِّل ألغوريثمًا لبحث عميق (بطريقة DFS) على الرسم البيانيّ المعطى ابتداءً من الرأس a.
ارسم في دفترك الشجرة الامتداديّة (DFS) / الغابة الامتداديّة (DFS) التي نتجت.

ii فَعِّل ألغوريثمًا لبحث عميق (بطريقة DFS) على الرسم البيانيّ المعطى ابتداءً من الرأس c.
ارسم في دفترك الشجرة الامتداديّة (DFS) / الغابة الامتداديّة (DFS) التي نتجت.

iii فَعِّل ألغوريثمًا لبحث عرضيّ (بطريقة BFS) على الرسم البيانيّ المعطى ابتداءً من الرأس d.
ارسم في دفترك الشجرة الامتداديّة (BFS) / الغابة الامتداديّة (BFS) التي نتجت.

/ يتبع في صفحة 25 /

جـ. $G = (V , E)$ هو رسم بياني موجَّه، عندما V هي مجموعة الرؤوس في الرسم البيانيّ وَ E هي مجموعة الأقواس في الرسم البيانيّ. G ممثَّل بواسطة قائمة مُجاوَرات.
أمامك وصف لألغوريثم يفحص إذا كانت توجد دائرة في الرسم البيانيّ G ، ويطبع بلاغًا ملائمًا.

ينقص في الألغوريثم تعبيران مشار إليهما بـ **(1)** وَ **(2)**.

وصف الألغوريثم

الخطوة 1: فعِّل الألغوريثم لإيجاد مركِّبات ربط قويّ (SCC) على الرسم البيانيّ G .
(الألغوريثم يُنتِج مركِّبات الربط القويّ الموجودة في الرسم البيانيّ.)

الخطوة 2: إذا كان **(1)** **(2)** ، عندها
المخرَج هو: في الرسم البيانيّ G لا توجد دائرة.

الخطوة 3: خلاف ذلك
المخرَج هو: في الرسم البيانيّ G توجد دائرة.

أمامك ثلاث إمكانيّات لإكمال التعبير **(1)** وثلاث إمكانيّات لإكمال التعبير **(2)**.
الإمكانيّات:

(1) هي إحدى التالية:
- لا يوجد مركِّب ربط قويّ يحوي
- مركِّب ربط قويّ واحد على الأقلّ يحوي
- كلّ واحد من مركِّبات الربط القويّ يحوي

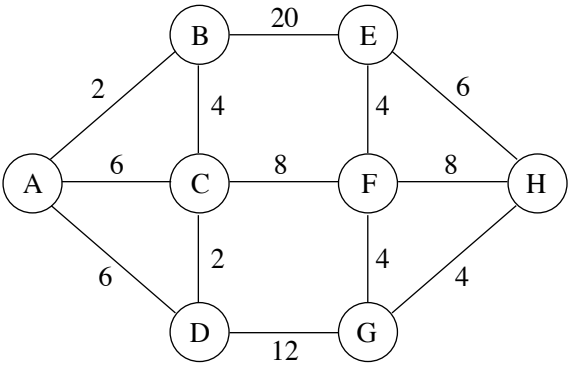
(2) هي إحدى التالية:
- رأسًا واحدًا فقط
- رأسين على الأقلّ
- جميع رؤوس الرسم البيانيّ

اختَر إحدى الإمكانيّات لإكمال التعبير **(1)** وإحدى الإمكانيّات لإكمال التعبير **(2)** ، بحيث ينفّذ الألغوريثم المطلوب.
اكتب اختيارَيْك في دفترك.

/ يتبع في صفحة 26 /

12. في هذا السؤال بندان، "أ–ب"، لا يتعلّقان ببعضهما البعض. أجب عن البندين.

أ. أمامك شبكة:



اكتب في دفترك أقصر خمسة مسارات من الرأس A إلى الرأس H في الشبكة المعطاة.
صف كلّ مسار كهذا على حِدة.

ب. الرسم البيانيّ G هو قابل للربط وليس موجَّهًا، ومعرَّف بواسطة $G = (V , E)$ ، عندما V هي مجموعة الرؤوس في الرسم البيانيّ وَ E هي مجموعة الأقواس في الرسم البيانيّ.
دالّة الوزن $W:E\rightarrow R^{+}$ تحدّد وزنًا (عددًا) لكلّ قوس في الرسم البيانيّ G .

لتكن X وَ Y وَ Z رؤوسًا في الرسم البيانيّ G.
نفترض أنّ كلّ قوس في الرسم البيانيّ ملوَّن بلون أزرق أو بلون أحمر.
أمامك وصف لألغوريثم، يفحص إذا كان من بين المسارات المكوَّنة من أقواس زرقاء فقط، جميع المسارات الأقصر من X إلى Y تمرّ عبر Z.
إذا كانت تمرّ – يعيد الألغوريثم القيمة "صدق" (TRUE).
خلاف ذلك – يعيد القيمة "كذب" (FALSE).
تنقص في الألغوريثم 5 تعابير مشار إليها بِـ **(1)-(5)**.

/ يتبع في صفحة 27 /

وصف الألغوريثم

الخطوة 1:

نُزيل من الرسم البيانيّ جميع الأقواس **(1)** ونحصل على رسم بيانيّ جديد $G_1 = (V , E_1)$.

الخطوة 2:

نحسب، بمساعدة ألغوريثم ديكسترا، طول أقصر مسار من X إلى Y في الرسم البيانيّ **(2)**.
نرمز إلى طول هذا المسار بِـ M_1 .

الخطوة 3:

نُزيل من الرسم البيانيّ G_1 الرأس **(3)** وجميع الأقواس الموصولة به، ونحصل على رسم بيانيّ جديد $G_2 = (V_2 , E_2)$.

الخطوة 4:

نحسب، بمساعدة ألغوريثم ديكسترا، طول أقصر مسار من X إلى Y في الرسم البيانيّ **(4)**.
نرمز إلى طول هذا المسار بِـ M_2 .

الخطوة 5:

إذا كان **(5)** عندها أعِد القيمة "صدق".
خلاف ذلك – أعِد القيَمة "كذب".

انسخ الجدول الذي أمامك إلى دفترك، واكتب فيه التعابير الناقصة.

التعبير **(1)**	
التعبير **(2)**	
التعبير **(3)**	
التعبير **(4)**	
التعبير **(5)**	

/ يتبع في صفحة 28 /

موديلات حسابيّة

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

13. نعرِّف اللغة L_1 فوق الأبجديّة $\{a, c\}$:

$$L_1 = \{c^n a^{n+2} \mid n > 0\}$$

أمامك اللغة L فوق الأبجديّة $\{a, b, c\}$:

$$L = \{w_1 w_2 w_3 ... w_k b^k \mid k > 0 \ , \ w_i \in L_1\}$$

مثال: الكلمة caaaccaaaacaaabbb هي كلمة في اللغة L، عندما $k = 3$، وكذلك:

$$w_1 = caaa$$
$$w_2 = ccaaaa$$
$$w_3 = caaa$$

ابنِ أوتومات راصّة يتلقّى اللغة L.

14. أمامك اللغة L فوق الأبجديّة $\{\$, 0\}$:

$$L = \left\{ 0^3 \$ 0^{i_1} \$ 0^{i_2} \$ ... 0^{i_k} \$ \;\middle|\; \begin{array}{r} k \geq 1 \ , \\ \text{لكلّ } m \text{ بين } 1 \text{ وَ } k : i_m \geq 0 \\ i_m \text{ وَ } \text{يقسم على } 3 \text{ بدون باقٍ} \end{array} \right\}$$

أ. اكتب أقصر كلمة في اللغة L.

ب. ابنِ أوتوماتًا نهائيًّا محدودًا يتلقّى اللغة L.

15. أمامك اللغة L فوق الأبجديّة $\{0, 1, 2\}$:

$$L = \{0^n 1^k 2 \mid n > k \geq 0\}$$

أ. برهن أنّ اللغة L غير نظاميّة.

ب. معطاة اللغة L_1 فوق الأبجديّة $\{1, 2\}$:

$$L_1 = \{1^n 2 \mid n \geq 0\}$$

ما هي اللغة $L \cap L_1$؟ علّل.

/ يتبع في صفحة 29 /

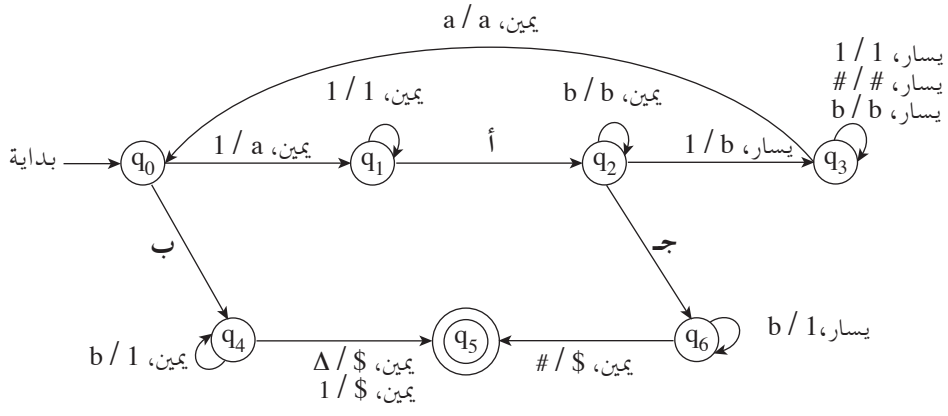
16. بُنيت آلة تيورينچ تحسب الدالّة $f(m,n)$. تجد هذه الآلة أصغر عدد من العددين m وَ n .
تتلقّى الآلة كمُدخَل عددين m وَ n صحيحين وأكبر من 0 . العددان مسجّلان على الشريط كعددين أونارّيين (العدد الأونارّي m هو عدد ممثَّل بواسطة m رموز من 1) وبين العددين مسجَّلة الإشارة # .
مُخرَج الآلة هو أصغر عدد من العددين، ويُسجَّل على الشريط كعدد أونارّي بين إشارتَي $.
الإشارة $ اليسرى تُسجَّل بدلاً من الإشارة # ، والعدد يُسجَّل عن يمينها.
خلال العمليّة تستطيع الآلة الاستعانة بالإشارتين a ، b .
مثال: بالنسبة للمُدخَلين $m=3$ وَ $n=5$ ، قبل الحساب يبدو شريط الذاكرة على النحو التالي:

⊢	1	1	1	#	1	1	1	1	1	Δ	Δ	Δ	…

بعد إنهاء الحساب يبدو الشريط على النحو التالي:

⊢	…	…	…	$	1	1	1	$	…

ملاحظة: الخلايا المشار إليها بـ… هي خلايا لا أهمّيّة لمحتواها.
أمامك رسم جزئيَ للآلة.



أ. توجد في الشريط ثلاثة انتقالات أُشير إليها بالأحرف أ–جـ.
تنقص في هذه الانتقالات إشارات الإدخال والأوامر.
انسخ الرسم إلى دفترك، وأكمل فيه الانتقالات الثلاثة الناقصة بحيث تحسب الآلة الدالّة $f(m,n)$.

ب. بيّن عمليّة الحساب في الآلة بالنسبة للمُدخَلين $m=1$ ، $n=1$.
في كلّ مرحلة، اكتب حالة الشريط، وأشر أين يتواجد رأس الآلة، واكتب في أيّ من الحالات q_6-q_0 تتواجد الآلة.

/ يتبع في صفحة 30 /

برمجة موجّهة كائنات

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

17. أمامك جزء من مشروع يتناول وسائل مواصلات ويشمل الفئات التالية:

الفئة Vehicle	تمثّل وسيلة مواصلات
الفئة Train	تمثّل قطارًا الذي هو وسيلة مواصلات
الفئة Boat	تمثّل قاربًا الذي هو وسيلة مواصلات
الفئة Airplane	تمثّل طائرة التي هي وسيلة مواصلات
الفئة TransportationCompany	تمثّل شركة لديها وسائل مواصلات من أنواع مختلفة

```
public class Vehicle
{
    private String type;          //  الموقع (أرض / مياه / جوّ)
    private String way;           //  نوع الطريق (شارع / سكّة حديد / نهر/ ...)
    private int maxSpeed;         //  السرعة القصوى

    public Vehicle(String type, String way, int maxSpeed)
    {
        this.type = type;
        this.way = way;
        this.maxSpeed = maxSpeed;
    }
}

public class Train extends Vehicle
{
    private int numOfCarriages;          //  عدد عربات القطار
    public Train(int maxSpeed, int numOfCarriages)
    {
        super("land", "tracks", maxSpeed);
        this.numOfCarriages = numOfCarriages;
    }
    public void incNumOfCarriages(int n)  //  تُكبِّر بـِ n عدد العربات في القطار
    {
        this.numOfCarriages = this.numOfCarriages + n;
    }
}
```

/ يتبع في صفحة 31 /

```
public class Boat extends Vehicle
{
    public Boat(String way, int maxSpeed)
    {
        super("water", way, maxSpeed);
    }
}

public class Airplane extends Vehicle
{
    private int maxHeight;                          // ارتفاع الطيران الأقصى
    public Airplane(int maxSpeed, int maxHeight)
    {
        super("sky" , "air" , maxSpeed);
        this.maxHeight = maxHeight;
    }
}

public class TransportationCompany
{
    private Vehicle[] vehicles = new Vehicle[50];   // مصفوفة وسائل المواصلات في الشركة
    private int counter = 0;                        // عدد وسائل المواصلات الموجودة فعليًّا
    public TransportationCompany()
    {
    }

    public void addVehicle (Vehicle v)              // تُضيف وسيلة مواصلات إلى مصفوفة وسائل
    {                                               // المواصلات في الشركة. افترض أنّه يوجد مكان
                                                    // لإضافة وسيلة مواصلات.
        this.vehicles[counter] = v;
        this.counter++;
    }
}
```

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 32 /

أ. طبِّق بلغة Java فئة رئيسيّة Program فيها عمليّة رئيسيّة، تنفّذ المهمّتين التاليتين:

i بناء كائن من نمط شركة وسائل مواصلات — **TransportationCompany** يُسمّى company1 .

ii إضافة قارب واحد وقطار واحد إلى الشركة company1 .

اختر قيَمًا للصفات كما تشاء.

ب. في الفئة **TransportationCompany** عُرِّفت العمليّة:

```
public void display()
{
    for (int i=0; i<this.counter; i++)
    {
        System.out.println((i+1) + ":" + this.vehicles[i]);
    }
}
```

طبِّق بلغة Java عمليّات تُمكّن تنفيذًا سليمًا للعمليّة display() ، بحيث تُطبَع لكلّ وسيلة مواصلات جميع صفاتها. عرِّف العمليّات بالطريقة الأكثر ملاءمة لمبادئ البرمجة الموجّهة كائنات (تعليب — encapsulation ، توريث — inheritance ، تعدّد الأشكال — polymorphism) .

بالنسبة لكلّ عملية تطبّقها، اكتب إلى أيّ فئة تتبع.

لا يمكن تغيير العمليّة display() .

جـ. طبِّق بلغة Java عمليّة، تتلقّى عددًا صحيحًا n وتُضيف n عربات إلى كلّ القطارات التي تتبع للشركة التي لديها وسائل مواصلات من أنواع مختلفة. وثِّق العمليّة، واكتب في أيّ فئة يجب تعريفها. لا يمكن تغيير العمليّات الموجودة في المشروع.

/ يتبع في صفحة 33 /

18. أمامك الفئتان **AA** وَ **BB** :

```
public class AA
{
    private String st;

    public AA()                 {  this.st = "excellent";  }
    public AA(String st)        {  this.st = st;           }
    public String getSt()       {  return this.st;         }
    public void setSt(String st){  this.st = st;           }
    public String toString()    {  return "st = " + this.st; }
}

public class BB extends AA
{
    private int num;

    public BB()                 {  super();  this.num = 1;  }
    public BB(int num , String st) {  super(st); this.num = Math.abs(num);  }
    public int getNum()         {  return this.num;        }
    public void setNum(int num) {  this.num = num;}
    public String toString()    {  return super.toString() + " num = " + this.num;}
}
```

أ. عرِّف في الفئة **AA** عمليّة بوليانيّة باسم isLike(Object obj)، تتلقّى كائنًا obj من نمط Object. إذا كان الكائن obj من نمط **AA** وكذلك محتوى النصّ st التابع لِـ obj مطابقًا لِمحتوى النصّ st التابع للكائن الحاليّ – تُعيد العمليّة true، خلاف ذلك – تُعيد false.

ب. عرِّف في الفئة **BB** عمليّة تدهس العمليّة التي عرّفتَها في البند "أ".
إذا كان الكائن obj من نمط **BB** وكذلك قيمة صفته num مساوية لقيمة الصفة num التابعة للكائن الحاليّ – تُعيد العمليّة true ، خلاف ذلك – تُعيد false.

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 34 /

جـ. أمامك قطعة من عمليّة رئيسيّة:

```
AA  a = new AA("excellent");
BB  b = new BB();
a = b;
if (a.isLike(b))  System.out.println(a);
```

هل قطعة البرنامج سليمة؟

إذا كانت سليمة – ماذا يكون مُخرَج القطعة؟ اكتب أيّة صيغة للعمليّة isLike تُفعَّل – الصيغة التابعة لِـ AA أم الصيغة التابعة لِـ BB.

إذا كانت غير سليمة – اشرح ما هو الخطأ ومتى سيظهر: أثناء التجميع والتحويل للغة الآلة أم أثناء التشغيل.

د. أمامك قطعة من عمليّة رئيسيّة:

```
AA  aa = new AA();
BB  bb = new BB(2 , "excellent");
bb = aa;
if (bb.isLike(aa))  System.out.println(bb);
```

هل قطعة البرنامج سليمة؟

إذا كانت سليمة – ماذا يكون مُخرَج القطعة؟ اكتب أيّة صيغة للعمليّة isLike تُفعَّل – الصيغة التابعة لِـ AA أم الصيغة التابعة لِـ BB.

إذا كانت غير سليمة – اشرح ما هو الخطأ ومتى سيظهر: أثناء التجميع والتحويل للغة الآلة أم أثناء التشغيل.

هـ. اكتب عمليّة خارجيّة باسم longString تتلقّى مصفوفة لكائنات من نمط Object .
تُعيد العمليّة نصًّا مركَّبًا من دمج الصفة st لكائنات من نمط AA في المصفوفة، على النحو التالي:

- إذا كانت للكائن الصفة st فقط، النصّ الذي في الصفة st يُدمَج مرّة واحدة.
- إذا كانت للكائن الصفة num أيضًا، النصّ الذي في الصفة st يُدمَج num مرّات.
- إذا لم يكن في المصفوفة أيّ كائن من نمط AA ، يُعاد نصّ فارغ.

/ يتبع في صفحة 35 /

19. طوّرت شركة لتوزيع برامج الحاسوب برنامج حاسوب يتناول متواليات لأعداد صحيحة.
تمّ تطوير المنظومة على مراحل.
بالنسبة لكلّ متوالية أعداد، يتطرّقون إلى:
(1) الحدّ الأوّل في المتوالية الذي رقمه التسلسليّ هو 1.
(2) الحدّ الذي رقمه التسلسليّ في المتوالية هو n.
(3) طباعة n الحدود الأولى في المتوالية.

طُوّرت **في المرحلة الأولى** فئتان:
متوالية حسابيّة (**ASeq**) – متوالية الفرق فيها بين كلّ حدّ والحدّ الذي قبله هو قيمة ثابتة.
متوالية هندسيّة (**GSeq**) – متوالية حاصل القسمة (الأساس) فيها بين كلّ حدّ والحدّ الذي قبله هو قيمة ثابتة.

فيما يلي كود الفئتين اللتين طُوِّرتا في المرحلة الأولى:

```
public class ASeq
{
    private int first;
    private int difference;

    public ASeq(int first, int difference)
    {
        this.first = first;
        this.difference = difference;
    }

    public int theNElement(int n)
    {
        return this.first + (n−1) ∗ this.difference;
    }
```

(انتبه: تكملة السؤال في الصفحات التالية.)

/ يتبع في صفحة 36 /

```
    public void displayNElements(int n)
    {
        System.out.print("The sequence elements: ");
        for (int i = 0; i < n−1; i++)
            System.out.print(this.theNElement(i+1) + " , ");
        System.out.println(this.theNElement(n));
    }
}

public class GSeq
{
    private int first;
    private int product;

    public GSeq(int first , int product)
    {
        this.first = first;
        this.product = product;
    }

    public int theNElement(int n)
    {
        return this.first * (int)Math.pow(this.product , n−1);
    }

    public void displayNElements(int n)
    {
        System.out.print("The sequence elements: ");
        for (int i = 0 ; i < n−1; i++)
            System.out.print(this.theNElement(i+1) + " , ");
      System.out.println(this.theNElement(n));
    }
}
```

/ يتبع في صفحة 37 /

أ. تتبّع قطعة البرنامج التي أمامك. اعرض في المتابعة الكائن الذي يُبنى وصفاته والمُخرَج.

```
ASeq  aSeq = new ASeq(2 , 3);
System.out.println(aSeq.theNElement(4));
aSeq.displayNElements(5);
```

تَقَرَّرَ **في المرحلة الثانية** من التطوير أنّه من المناسب تطوير فئة جديدة تصف متوالية ثابتة (**Sequence**) بحيث تَرِث الفئتان **ASeq** وَ **GSeq** من الفئة الجديدة. في المتوالية الثابتة معرَّفة قيمة الحدّ الأوّل، وجميع سائر الحدود مساوية للحدّ الأوّل.

ب. أكمِل تطوير المرحلة الثانية بالطريقة الأكثر ملاءمة لمبادئ البرمجة موجّهة الكائنات، وحسب التعليمتين (i)-(ii):

(i) طبّق بشكل كامل الفئة العليا **Sequence**. يجب أن تتطرّق الفئة إلى:
(1) الحدّ الأوّل في المتوالية الذي رقمه التسلسليّ هو 1.
(2) الحدّ الذي رقمه التسلسليّ في المتوالية هو n .
(3) طباعة n الحدود الأولى في المتوالية.

(ii) طبِّق الفئة **ASeq** من جديد بحيث تَرِث من الفئة **Sequence** .

تَقَرَّرَ **في المرحلة الثالثة** من التطوير توسيع المشروع الذي يشمل ثلاث الفئات التي تمّ تطويرها في المرحلة الثانية (**Sequence** , **ASeq** , **GSeq**)، بحيث يكون بالإمكان بالنسبة لكلّ متوالية، تفعيل عمليّة تحسب وتُعيد مجموع n الحدود الأولى في المتوالية.
افترض أنّ الفئة **GSeq** قد تمّ تطبيقها من جديد بحيث تَرِث من الفئة **Sequence**.

جـ. بالنسبة لكلّ واحدة من الفئات **Sequence**, **ASeq** , **GSeq** ، اكتب هل يجب إجراء تغييرات فيها بحيث يفي المشروع بمتطلّبات التطوير في المرحلة الثالثة بالطريقة الأكثر ملاءمة لمبادئ البرمجة الموجَّهة كائنات.
إذا كان يجب إجراء تغييرات — فصِّلها وطبِّقها.

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 38 /

تَقَرَّرَ **في المرحلة الرابعة** من التطوير، تطوير عمليّة ستاتيّة check تتلقّى قيمة صحيحة n ، وكائنين لمتواليات: أحدهما من نمط **ASeq** والآخر من نمط **GSeq**.

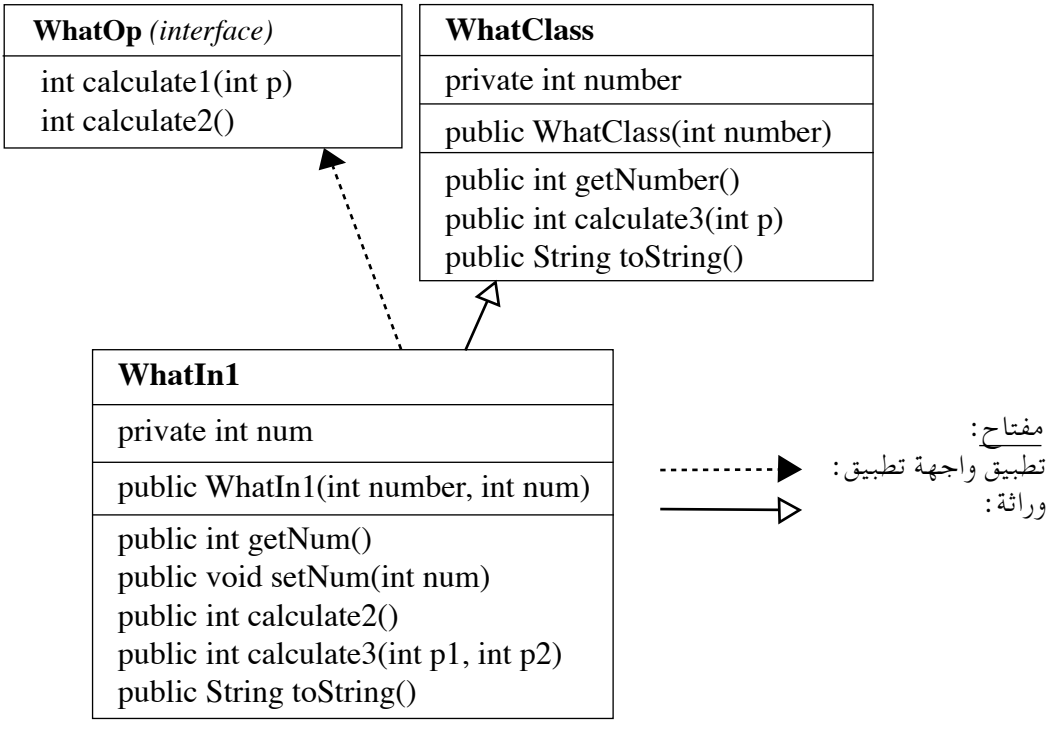
تحسب العمليّة مجموع n الحدود الأولى في كلّ واحدة من المتواليتين وتُعيد:

- الرمز 'A' — إذا كان مجموع n الحدود الأولى في المتوالية من نمط **ASeq** هو الأكبر من بين المجموعين.
- الرمز 'G' — إذا كان مجموع n الحدود الأولى في المتوالية من نمط **GSeq** هو الأكبر من بين المجموعين.
- الرمز 'E' — إذا كان مجموع n الحدود الأولى في المتواليتين متساويًا.

د. طبِّق العمليّة الستاتيّة check حسب المتطلّبات التي عُرِّفت في المرحلة الرابعة من التطوير.

/ يتبع في صفحة 39 /

20. أمامك UML جزئيّ:



أ. هل تطبيق العمليّة public int calculate1(int p) في الفئة **WhatClass**، يفي بمتطلّب تطبيقها في الفئة **WhatIn1** ؟ علّل إجابتك.

ب. تطبيق العمليّة البنائيّة في الفئة **WhatIn1** هو:

```
public WhatIn1(int number, int num)
{
    this.number = number;
    this.num = num;
}
```

هل العمليّة سليمة؟ إذا كانت سليمة – صف تنفيذ العمليّة؛ إذا لم تكن سليمة – صحِّح العمليّة (لا تغيِّر شيئًا باستثناء العمليّة نفسها). اكتب العمليّة المصحَّحة في دفترك.

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 40 /

جـ. في الفئة **WhatIn1** طُبِّقت العمليّة calculate2() ، التي تُعيد القيمة الصحيحة لمعدّل صفات الكائن:

```
public int calculate2()
{    return (int)((this.number + this.num)/2);                }
```

هل العمليّة سليمة؟ إذا كانت سليمة – صف تنفيذ العمليّة؛ إذا لم تكن سليمة – صحِّح العمليّة (لا تغيِّر شيئًا باستثناء العمليّة نفسها) . اكتب العمليّة المصحَّحة في دفترك .

د. أمامك وصف جزئيّ للفئة **WhatIn2** :

class **WhatIn2** extends **WhatIn1**
private int sum
public WhatIn2(int number, int num, int sum)
public int calculate3(int p1, int p2, int p3)

(1) هل يمكن الاعتماد على العمليّة البنائيّة الاختيار الافتراضيّ (ברירת מחדל) بدلاً من تعريف عمليّة بنائيّة في الفئة **WhatIn2**؟ علّل إجابتك .

(2) العمليّة calculate3 طُبِّقت في الفئات الثلاث **WhatClass** وَ **WhatIn1** وَ **WhatIn2** على النحو التالي:

class **WhatClass**	public int calculate3(int p) { return this.number * p; }
class **WhatIn1**	public int calculate3(int p1, int p2) { return this.calculate3(p1) + this.num * p2 * p2; }
class **WhatIn2**	public int calculate3(int p1, int p2, int p3) { return this.calculate3(p1, p2) + this.sum * p3 * p3 * p3;}

افترض أنّ الكائن obj هو من نمط **WhatIn2** ، وقيَم صفاته هي:

number = 1

num = 2

sum = 3

أمامك أمر مكتوب في العمليّة الرئيسيّة:

```
System.out.println(obj.calculate3(1000, 100, 10));
```

بيّن متابعة تنفيذ الأمر. تطرّق في المتابعة إلى استدعاء العمليّات وإلى قيَم صفات الكائن. اكتب المُخرَج الذي ينتج.

/ يتبع في صفحة 41 /

برمجة موجّهة كائنات

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

21. أمامك جزء من مشروع يتناول وسائل مواصلات ويشمل الفئات التالية:

الفئة Vehicle	تمثّل وسيلة مواصلات
الفئة Train	تمثّل قطارًا الذي هو وسيلة مواصلات
الفئة Boat	تمثّل قاربًا الذي هو وسيلة مواصلات
الفئة Airplane	تمثّل طائرة التي هي وسيلة مواصلات
الفئة TransportationCompany	تمثّل شركة لديها وسائل مواصلات من أنواع مختلفة

```
public class Vehicle
{
    private string type;          //  الموقع (أرض / مياه / جوّ)
    private string way;           //  نوع الطريق (شارع / سكّة حديد / نهر / ...)
    private int maxSpeed;         //  السرعة القصوى

    public Vehicle(string type, string way, int maxSpeed)
    {
        this.type = type;
        this.way = way;
        this.maxSpeed = maxSpeed;
    }
}

public class Train : Vehicle
{
    private int numOfCarriages;          //  عدد عربات القطار
    public Train(int maxSpeed, int numOfCarriages) : base("land", "tracks",maxSpeed)
    {
    this.numOfCarriages = numOfCarriages;
    }
    public void incNumOfCarriages(int n)  //  تُكبِّر بـ n عدد العربات في القطار
    {
        this.numOfCarriages = this.numOfCarriages + n;
    }
}
```

(**انتبه: تكملة السؤال في الصفحة التالية.**) / يتبع في صفحة 42 /

```
public class Boat : Vehicle
{
    public Boat(string way, int maxSpeed) : base("water", way, maxSpeed)
    {

    }
}

public class Airplane : Vehicle
{
    private int maxHeight;                          // ارتفاع الطيران الأقصى
    public Airplane(int maxSpeed, int maxHeight) : base("sky", "air", maxSpeed)
    {
        this.maxHeight = maxHeight;
    }
}

public class TransportationCompany
{
    private Vehicle[] vehicles = new Vehicle[50];   // مصفوفة وسائل المواصلات في الشركة
    private int counter = 0;                        // عدد وسائل المواصلات الموجودة فعليًا
    public TransportationCompany()
    {
    }

    public void AddVehicle (Vehicle v)              // تُضيف وسيلة مواصلات إلى مصفوفة وسائل
    {                                               المواصلات في الشركة. افترض أنّه يوجد مكان
                                                    لإضافة وسيلة مواصلات //
        this.vehicles[counter] = v;
        this.counter++;
    }
}
```

/ يتبع في صفحة 43 /

أ. طبِّق بلغة C# فئة رئيسيّة Program فيها عمليّة رئيسيّة، تنفّذ المهمّتين التاليتين:

i بناء كائن من نمط شركة وسائل مواصلات – **TransportationCompany** يُسمّى company1.

ii إضافة قارب واحد وقطار واحد إلى الشركة company1.

اختر قيَمًا للصفات كما تشاء.

ب. في الفئة **TransportationCompany** عُرِّفت العمليّة:

```
public void Display()
{
    for (int i=0; i<this.counter; i++)
    {
        Console.WriteLine((i+1) + ":" + this.vehicles[i]);
    }
}
```

طبِّق بلغة C# عمليّات تُمكّن تنفيذًا سليمًا للعمليّة Display() ، بحيث تُطبَع لكلّ وسيلة مواصلات جميع صفاتها. عرِّف العمليّات بالطريقة الأكثر ملاءمة لمبادئ البرمجة الموجّهة كائنات (تعليب – encapsulation ، توريث – inheritance ، تعدّد الأشكال – polymorphism).

بالنسبة لكلّ عمليّة تطبّقها، اكتب إلى أيّ فئة تتبع.

لا يمكن تغيير العمليّة Display().

جـ. طبِّق بلغة C# عمليّة، تتلقّى عددًا صحيحًا n وتُضيف n عربات إلى كلّ القطارات التي تتبع للشركة التي لديها وسائل مواصلات من أنواع مختلفة. وثِّق العمليّة، واكتب في أيّ فئة يجب تعريفها. لا يمكن تغيير العمليّات الموجودة في المشروع.

/ يتبع في صفحة 44 /

22. أمامك الفئتان **AA** وَ **BB** :

```
public class AA
{
    private string st;
    public AA()                          { this.st = "excellent";    }
    public AA(string st)                 { this.st = st;             }
    public string GetSt()                { return this.st;           }
    public void SetSt (string st)        { this.st = st;             }
    public override string ToString()    { return "st = " + this.st; }
}

public class BB : AA
{
    private int num;
    public BB() : base()                          { this.num = 1;    }
    public BB(int num, string st) : base(st)      { this.num = Math.Abs(num);      }
    public int GetNum()                           { return this.num; }
    public void SetNum(int num)                   { this.num = num;}
    public override string ToString()             { return base.ToString() + " num = "
                                                    + this.num;}
}
```

أ. عرِّف في الفئة **AA** عمليّة بوليانيّة باسم IsLike (Object obj)، تتلقّى كائنًا obj من نمط Object. إذا كان الكائن obj من نمط **AA** وكذلك محتوى النصّ st التابع لِـ obj مطابقًا لِمحتوى النصّ st التابع للكائن الحاليّ – تُعيد العمليّة true، خلاف ذلك – تُعيد false.

ب. عرِّف في الفئة **BB** عمليّة تدهس العمليّة التي عرّفتَها في البند "أ".
إذا كان الكائن obj من نمط **BB** وكذلك قيمة صفته num مساوية لقيمة الصفة num التابعة للكائن الحاليّ – تُعيد العمليّة true ، خلاف ذلك – تُعيد false.

/ يتبع في صفحة 45 /

جـ. أمامك قطعة من عمليّة رئيسيّة:

```
AA a = new AA("excellent");
BB b = new BB();
a = b;
if (a.IsLike(b)) Console.WriteLine(a);
```

هل قطعة البرنامج سليمة؟

إذا كانت سليمة – ماذا يكون مُخرَج القطعة؟ اكتب أيّة صيغة للعمليّة IsLike تُفعَّل – الصيغة التابعة لِـ AA أم الصيغة التابعة لِـ BB.

إذا كانت غير سليمة – اشرح ما هو الخطأ ومتى سيظهر: أثناء التجميع والتحويل للغة الآلة أم أثناء التشغيل.

د. أمامك قطعة من عمليّة رئيسيّة:

```
AA  aa = new AA();
BB  bb = new BB(2 , "excellent");
bb = aa;
if (bb.IsLike(aa)) Console.WriteLine(bb);
```

هل قطعة البرنامج سليمة؟

إذا كانت سليمة – ماذا يكون مُخرَج القطعة؟ اكتب أيّة صيغة للعمليّة IsLike تُفعَّل – الصيغة التابعة لِـ AA أم الصيغة التابعة لِـ BB.

إذا كانت غير سليمة – اشرح ما هو الخطأ ومتى سيظهر: أثناء التجميع والتحويل للغة الآلة أم أثناء التشغيل.

هـ. اكتب عمليّة خارجيّة باسم LongString تتلقّى مصفوفة لكائنات من نمط Object.

تُعيد العمليّة نصًّا مركَّبًا من دمج الصفة st لكائنات من نمط AA في المصفوفة، على النحو التالي:

- إذا كانت للكائن الصفة st فقط، النصّ الذي في الصفة st يُدمَج مرّة واحدة.
- إذا كانت للكائن الصفة num أيضًا، النصّ الذي في الصفة st يُدمَج num مرّات.
- إذا لم يكن في المصفوفة أيّ كائن من نمط AA، يُعاد نصّ فارغ.

/ يتبع في صفحة 46 /

23. طوّرت شركة لتوزيع برامج الحاسوب برنامج حاسوب يتناول متواليات لأعداد صحيحة.
تمّ تطوير المنظومة على مراحل.
بالنسبة لكلّ متوالية أعداد، يتطرّقون إلى:
(1) الحدّ الأوّل في المتوالية الذي رقمه التسلسليّ هو 1.
(2) الحدّ الذي رقمه التسلسليّ في المتوالية هو n.
(3) طباعة n الحدود الأولى في المتوالية.

طُوِّرت **في المرحلة الأولى** فئتان:
متوالية حسابيّة (**ASeq**) – متوالية الفرق فيها بين كلّ حدّ والحدّ الذي قبله هو قيمة ثابتة.
متوالية هندسيّة (**GSeq**) – متوالية حاصل القسمة (الأساس) فيها بين كلّ حدّ والحدّ الذي قبله هو قيمة ثابتة.

فيما يلي كود الفئتين اللتين طُوِّرتا في المرحلة الأولى:

```
public class ASeq
{
    private int first;
    private int difference;

    public ASeq(int first, int difference)
    {
        this.first = first;
        this.difference = difference;
    }

    public int TheNElement(int n)
    {
        return this.first + (n−1) ∗ this.difference;
    }
```

/ يتبع في صفحة 47 /

```
    public void DisplayNElements(int n)
    {
        Console.Write("The sequence elements: ");
        for (int i = 0; i < n−1; i++)
                Console.Write(this.TheNElement(i+1) + " , ");
        Console.WriteLine(this.TheNElement(n));
    }
}
public class GSeq
{
    private int first;
    private int product;

    public GSeq(int first, int product)
    {
        this.first = first;
        this.product = product;
    }
    public int TheNElement(int n)
    {
        return this.first * (int)Math.Pow(this.product , n−1);
    }
    public void DisplayNElements(int n)
    {
        Console.Write("The sequence elements: ");
            for (int i = 0; i < n−1; i++)
                Console.Write(this.TheNElement(i+1) + " , ");
            Console.WriteLine (this.TheNElement(n));
    }
}
```

(انتبه: تكملة السؤال في الصفحة التالية.)
/ يتبع في صفحة 48 /

أ. تتبّع قطعة البرنامج التي أمامك. اعرض في المتابعة الكائن الذي يُبنى وصفاته والمُخرَج.

```
ASeq aSeq = new ASeq(2, 3);
Console.WriteLine(aSeq.TheNElement(4));
aSeq.DisplayNElements(5);
```

تَقَرَّرَ **في المرحلة الثانية** من التطوير أنّه من المناسب تطوير فئة جديدة تصف متوالية ثابتة (**Sequence**) بحيث تَرِث الفئتان **ASeq** وَ **GSeq** من الفئة الجديدة. في المتوالية الثابتة معرَّفة قيمة الحدّ الأوّل، وجميع سائر الحدود مساوية للحدّ الأوّل.

ب. أكمِل تطوير المرحلة الثانية بالطريقة الأكثر ملاءمة لمبادئ البرمجة موجّهة الكائنات، وحسب التعليمتين (i)-(ii):

(i) طبّق بشكل كامل الفئة العليا **Sequence**. يجب أن تتطرّق الفئة إلى:

(1) الحدّ الأوّل في المتوالية الذي رقمه التسلسليّ هو 1.

(2) الحدّ الذي رقمه التسلسليّ في المتوالية هو n .

(3) طباعة n الحدود الأولى في المتوالية.

(ii) طبِّق الفئة **ASeq** من جديد بحيث تَرِث من الفئة **Sequence** .

تَقَرَّرَ **في المرحلة الثالثة** من التطوير توسيع المشروع الذي يشمل ثلاث الفئات التي تمّ تطويرها في المرحلة الثانية (**Sequence** , **ASeq** , **GSeq**)، بحيث يكون بالإمكان بالنسبة لكلّ متوالية، تفعيل عمليّة تحسب وتُعيد مجموع n الحدود الأولى في المتوالية.

افترض أنّ الفئة **GSeq** قد تمّ تطبيقها من جديد بحيث تَرِث من الفئة **Sequence** .

جـ. بالنسبة لكلّ واحدة من الفئات **Sequence**, **ASeq** , **GSeq** ، اكتب هل يجب إجراء تغييرات فيها بحيث يفي المشروع بمتطلّبات التطوير في المرحلة الثالثة بالطريقة الأكثر ملاءمة لمبادئ البرمجة الموجَّهة كائنات.

إذا كان يجب إجراء تغييرات — فصِّلها وطبِّقها.

/ يتبع في صفحة 49 /

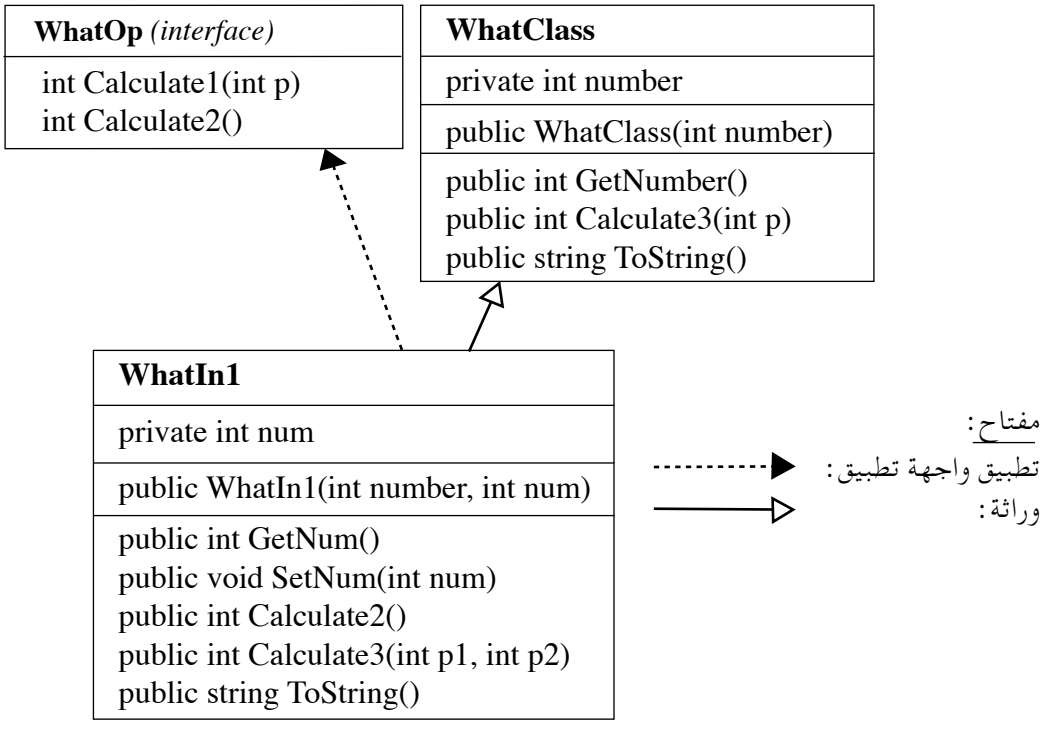
تَقَرَّرَ **في المرحلة الرابعة** من التطوير، تطوير عمليّة ستاتيّة Check تتلقّى قيمة صحيحة n ، وكائنين لمتواليات: أحدهما من نمط **ASeq** والآخر من نمط **GSeq**. تحسب العمليّة مجموع n الحدود الأولى في كلّ واحدة من المتواليتين وتُعيد:

- الرمز 'A' — إذا كان مجموع n الحدود الأولى في المتوالية من نمط **ASeq** هو الأكبر من بين المجموعين.
- الرمز 'G' — إذا كان مجموع n الحدود الأولى في المتوالية من نمط **GSeq** هو الأكبر من بين المجموعين.
- الرمز 'E' — إذا كان مجموع n الحدود الأولى في المتواليتين متساويًا.

د. طبِّق العمليّة الستاتيّة Check حسب المتطلّبات التي عُرِّفت في المرحلة الرابعة من التطوير.

/ يتبع في صفحة 50 /

24. أمامك UML جزئيّ :



أ. هل تطبيق العمليّة public int Calculate1(int p) في الفئة **WhatClass** ، يفي بمتطلّب تطبيقها في الفئة **WhatIn1** ؟ علّل إجابتك.

ب. تطبيق العمليّة البنائيّة في الفئة **WhatIn1** هو :

```
public WhatIn1(int number, int num)
{
    this.number = number;
    this.num = num;
}
```

هل العمليّة سليمة؟ إذا كانت سليمة – صف تنفيذ العمليّة؛ إذا لم تكن سليمة – صحِّح العمليّة (لا تغيِّر شيئًا باستثناء العمليّة نفسها) . اكتب العمليّة المصحَّحة في دفترك.

/ يتبع في صفحة 51 /

جـ. في الفئة **WhatIn1** طُبِّقت العمليّة Calculate2() ، التي تُعيد القيمة الصحيحة لمعدّل صفات الكائن:

```
public int Calculate2()
{    return (int)((this.number + this.num)/2);    }
```

هل العمليّة سليمة؟ إذا كانت سليمة – صف تنفيذ العمليّة؛ إذا لم تكن سليمة – صحِّح العمليّة (لا تغيِّر شيئًا باستثناء العمليّة نفسها) . اكتب العمليّة المصحَّحة في دفترك.

د. أمامك وصف جزئيّ للفئة **WhatIn2** :

class **WhatIn2**: **WhatIn1**
private int sum
public WhatIn2(int number, int num, int sum)
public int Calculate3(int p1, int p2, int p3)

(1) هل يمكن الاعتماد على العمليّة البنائيّة الاختيار الافتراضيّ (בְּרֵרת מחדל) بدلاً من تعريف عمليّة بنائيّة في الفئة **WhatIn2** ؟ علّل إجابتك.

(2) العمليّة Calculate3 طُبِّقت في الفئات الثلاث **WhatClass** وَ **WhatIn1** وَ **WhatIn2** على النحو التالي:

class **WhatClass**	public int Calculate3(int p) { return this.number * p; }
class **WhatIn1**	public int Calculate3 (int p1, int p2) { return this.Calculate3(p1) + this.num * p2 * p2; }
class **WhatIn2**	public int Calculate3 (int p1, int p2, int p3) { return this.Calculate3(p1, p2) + this.sum * p3 * p3 * p3;}

افترض أنّ الكائن obj هو من نمط **WhatIn2** ، وقيَم صفاته هي:

number = 1

num = 2

sum = 3

أمامك أمر مكتوب في العمليّة الرئيسيّة:

```
Console.WriteLine(obj.Calculate3(1000, 100, 10));
```

بيّن متابعة تنفيذ الأمر. تطرّق في المتابعة إلى استدعاء العمليّات وإلى قيَم صفات الكائن. اكتب المُخرَج الذي ينتج.

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.