מדינת ישראל
משרד החינוך

סוג הבחינה: א. בגרות לבתי"ס על-יסודיים
ב. בגרות לנבחנים חיצוניים

מועד הבחינה: קיץ תשע"א
מספר השאלון: **899205, 603**
תרגום לערבית (2)

מדעי המחשב ב'

2 יחידות לימוד (השלמה ל-5 יח"ל)

הוראות לנבחן

א. משך הבחינה: שלוש שעות.

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

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

ד. הוראות מיוחדות:
1. את כל התכניות שאתה נדרש לכתוב בשפת מחשב בפרק הראשון, כתוב בשפה אחת בלבד — C# או Java.
2. **רשום על הכריכה החיצונית** של המחברת באיזו **שפה** אתה כותב — C# או Java.
3. **רשום על הכריכה החיצונית** של המחברת את שם ה**מסלול** שלמדת, אחד מארבעת המסלולים: מערכות מחשב ואסמבלר, 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.

בהצלחה!

دولة إسرائيل
وزارة المعارف

نوع الامتحان: أ. بجروت للمدارس الثانوية
ب. بجروت للممتحَنين الخارجيّين

موعد الامتحان: صيف ٢٠١١
رقم النموذج: **٨٩٩٢٠٥، ٦٠٣**
ترجمة إلى العربية (٢)

علم الحاسوب "ب"

وحدتان تعليميتان (تكملة لـ ٥ وحدات تعليمية)

تعليمات للممتحَن

أ. مدّة الامتحان: ثلاث ساعات.

ب. مبنى النموذج وتوزيع الدرجات:
في هذا النموذج فصلان.
الفصل الأوّل: في هذا الفصل أربعة أسئلة، عليك الإجابة عن اثنين منها.
(٢٥×٢) — ٥٠ درجة
الفصل الثاني: في هذا الفصل أسئلة في أربعة مسارات مختلفة. أجب فقط عن سؤالين في المسار الذي تعلّمتَه، حسب التعليمات في مجموعة الأسئلة في هذا المسار.
(٢٥×٢) — ٥٠ درجة
المجموع — ١٠٠ درجة

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

د. تعليمات خاصّة:
١. اكتب جميع البرامج التي يُطلب منك كتابتها بلغة حاسوب في الفصل الأوّل، بلغة واحدة فقط – C# أو Java.
٢. **سجّل على الغلاف الخارجي** للدفتر بأيّة **لغة** تكتب – C# أو Java.
٣. **سجّل على الغلاف الخارجي** للدفتر، اسم **المسار** الذي تعلّمتَه، أحد المسارات الأربعة: أنظمة حاسوب وأسمبلر، مقدّمة لبحث الأداءات، موديلات حسابية، برمجة موجّهة كائنات.

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

التعليمات في هذا النموذج مكتوبة بصيغة المذكّر وموجّهة للممتحَنات وللممتحَنين على حدّ سواء.

نتمنّى لك النجاح!

الأسئلة

في هذا النموذج فصلان: الفصل الأوّل والفصل الثاني.
عليك الإجابة عن أسئلة من الفصلين، حسب التعليمات في كلّ فصل.

الفصل الأوّل (٥٠ درجة)

انتبه: في كلّ سؤال يُطلب فيه تطبيق، بإمكانك استعمال عمليات الفئات Node<T> , Queue<T> ,Stack<T> ,BinTreeNode<T> ، بدون تطبيقها. إذا استعملتَ عمليات إضافية، عليك تطبيقها.

أجب عن اثنين من الأسئلة ١–٤ (لكلّ سؤال –٢٥ درجة).

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

للطلّاب الذين يحلّون بلغة Java

أ. أمامك عدّة فئات. تتبّع العملية main التي في الفئة Program واكتب المُخرَج. يجب أن تشمل المتابعة قيَم المتغيّرات، وبالنسبة لكلّ كائن – قيَم صفاته.

```
public class One
{
    private double n;
    public One(double n) {   this.n = n;    }
    public double getN()  {  return this.n;}
    public void f()       {  System.out.println("f of one ");   }
    public void g()       {  System.out.println("g of one " + this.getN());    }
}
public class Two
{
    private Queue<One> q;
    public Two(int n , int m)
    {
        this.q = new Queue<One>();
        for (int i = n; i < m; i++)
            this.q.insert(new One(Math.pow(2 , i)));
    }
    public void f()
    {
        System.out.print("f of two ");
        if (!this.q.isEmpty())
                System.out.println(this.q.remove().getN());
    }
}
```

/ يتبع في صفحة 3 /

```
public class Together
{
	private double x;
	private int from , to;
	private One first;
	private Two second;

	public Together(double x , int from , int to)
	{
		this.x = x;
		this.from = from;
		this.to = to;
		this.first = new One(x);
		this.second = new Two(from , to);
	}
	public void methodA()
	{
		System.out.println("--  MethodA()  -- ");
		this.first.f();
	}
	public void methodB()
	{
		System.out.println("--  MethodB()  -- ");
		this.first.g();
		this.second.f();
	}
}

public class Program
{
	public static void main(String[] args)
	{
		Together tg = new Together(5.0 , 2 , 6);
		tg.methodA() ;
		tg.methodB() ;
	}
}
```

ב. بافتراض أنّ العملية البنائية في الفئة Together تتلقّى أعداداً أكبر من 0 فقط –
ما الذي تنفّذه العملية f() في الفئة Two ؟

/ يتبع في صفحة 4 /

للطلّاب الذين يحلّون بلغة C#

أ. أمامك عدّة فئات. تتبّع العملية Main التي في الفئة Program واكتب المُخرَج.
يجب أن تشمل المتابعة قيَم المتغيّرات، وبالنسبة لكلّ كائن – قيَم صفاته.

```
public class One
{
    private double n;
    public One(double n) {   this.n = n;    }
    public double GetN()  {   return this.n;}
    public void F()       {   Console.WriteLine("F of one ");  }
    public void G()       {   Console.WriteLine("G of one " + this.GetN());  }
}

public class Two
{
    private Queue<One> q;
    public Two(int n , int m)
    {
        this.q = new Queue<One>();
        for (int i = n; i < m; i++)
            this.q.Insert(new One(Math.Pow(2 , i)));
    }
    public void F()
    {
        Console.Write("F of two ");
        if (!this.q.IsEmpty())
                Console.WriteLine(this.q.Remove().GetN());
    }
}
```

/ يتبع في صفحة 5 /

```
public class Together
{
    private double x;
    private int from , to;
    private One first;
    private Two second;

    public Together(double x , int from , int to)
    {
        this.x = x;
        this.from = from;
        this.to = to;
        this.first = new One(x);
        this.second = new Two(from , to);
    }
    public void MethodA()
    {
        Console.WriteLine("--  MethodA()  -- ");
        this.first.F();
    }
    public void MethodB()
    {
        Console.WriteLine("--  MethodB()  -- ");
        this.first.G();
        this.second.F();
    }
}

public class Program
{
    public static void Main(string[] args)
    {
        Together tg = new Together(5.0 , 2 , 6);
        tg.MethodA();
        tg.MethodB();
    }
}
```

ב. بافتراض أنّ العملية البنائية في الفئة Together تتلقّى أعداداً أكبر من 0 فقط –
ما الذي تنفّذه العملية F() في الفئة Two ؟

/ يتبع في صفحة 6 /

٢. **انتبه**: لهذا السؤال صيغتان: الأولى بلغة Java (في الصفحتين 6-7)، والثانية بلغة C# (في الصفحتين 8-9). اعمل حسب اللغة التي تعلّمتَها.

للطلّاب الذين يحلّون بلغة Java

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

للعمليتين 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;
}
```

/ يتبع في صفحة 7 /

معطاة المصفوفة a :

a	2	4	7	12	18

أ. تتبّع بواسطة جدول متابعة تنفيذ العملية 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 إضافيًا يُذكر فيه إذا كان الشرط الذي في الأمر if يتحقّق أم لا يتحقّق.

ب. تتبّع بواسطة جدول متابعة تنفيذ العملية 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 العملية sod ؟

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

هـ. تتبّع بواسطة جدول متابعة تنفيذ العملية 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 إضافيًا 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 العملية what ؟ علّل إجابتك.

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

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

(١) هل يتغيّر ادّعاء خروج العملية sod ؟ علّل إجابتك.

(٢) هل يتغيّر ادّعاء خروج العملية what ؟ علّل إجابتك.

/ يتبع في صفحة 8 /

للطلّاب الذين يحلّون بلغة C#

أمامك العمليتان Sod وَ What اللتان تتلقّيان المصفوفة a التي حدودها من نمط صحيح والمرتَّبة بترتيب تصاعدي، وتتلقّيان أيضًا عددًا صحيحًا k .

للعمليتين 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;
}
```

/ يتبع في صفحة 9 /

معطاة المصفوفة a :

a	2	4	7	12	18

أ. تتبّع بواسطة جدول متابعة تنفيذ العملية 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 إضافيًا يُذكر فيه إذا كان الشرط الذي في الأمر if يتحقّق أم لا يتحقّق.

ب. تتبّع بواسطة جدول متابعة تنفيذ العملية 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 العملية Sod ؟

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

هـ. تتبّع بواسطة جدول متابعة تنفيذ العملية 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 إضافيًا 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 العملية What ؟ علّل إجابتك.

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

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

(١) هل يتغيّر ادّعاء خروج العملية Sod ؟ علّل إجابتك.

(٢) هل يتغيّر ادّعاء خروج العملية What ؟ علّل إجابتك.

/ يتبع في صفحة 10 /

٣. **انتبه:** لهذا السؤال صيغتان: الأولى بلغة Java (في الصفحتين 10-11)، والثانية بلغة C# (في الصفحتين 12-13). اعمل حسب اللغة التي تعلّمتَها.

للطلّاب الذين يحلّون بلغة Java

أمامك واجهة تطبيق الفئة "**صفّ-إلغاء**" (**UndoQueue**) :

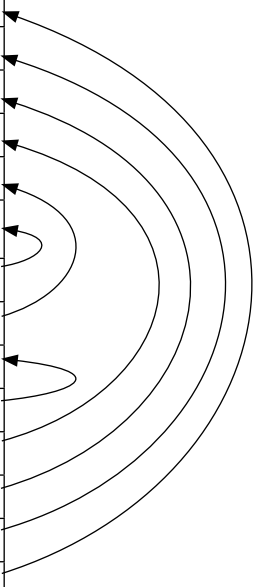
تعرّف الفئة نمط مجمّع مع بروتوكول FIFO لإدخال وإخراج قيَم صحيحة وأكبر من 0 .

اسم العملية	الوصف
UndoQueue()	تبني العملية **صفّ-إلغاء** فارغًا.
boolean isEmpty()	تُعيد العملية true إذا كان الـ**صفّ-إلغاء** الحالي فارغًا، خلاف ذلك – تُعيد العملية false .
void insert(int x)	تُدخِل العملية القيمة x في نهاية الـ**صفّ-إلغاء** الحالي.
int remove()	تُخرِج العملية القيمة التي في رأس الـ**صفّ-إلغاء** الحالي، وتُعيدها. فرضية: الـ**صفّ-إلغاء** الحالي ليس فارغًا.
int head()	تُعيد العملية قيمة الحدّ الذي في رأس الـ**صفّ-إلغاء** بدون إخراجه. فرضية: الـ**صفّ-إلغاء** الحالي ليس فارغًا.
void undo()	تُلغي العملية عملية الـ remove أو عملية الـ insert الأخيرة التي نُفِّذت، وتسترجع الـ**صفّ-إلغاء** كما كان قبل تنفيذ العملية. إذا لم يكن هناك عمليتا remove أو insert تمّ تنفيذهما، لا تفعل العملية شيئًا. انتبه: تشغيل العملية undo، n مرّات، يُعيد الـ**صفّ-إلغاء** إلى الوضع الذي كان عليه قبل تنفيذ n العمليات remove أو insert الأخيرة التي لم يتمّ إلغاؤها بعد.

/ يتبع في صفحة 11 /

مثال:

العملية	وضع الصفّ–**إلغاء** بعد تنفيذ العملية
UndoQueue uq = new UndoQueue();	الصفّ–**إلغاء** فارغ
uq.insert(2);	→2←
uq.insert(3);	→3 , 2←
uq.remove();	→3←
uq.insert(4);	→4 , 3←
uq.remove();	→4←
uq.remove();	الصفّ–**إلغاء** فارغ
uq.undo();	→4←
uq.undo();	→4 , 3←
uq.insert(1);	→1 , 4 , 3←
uq.undo();	→4 , 3←
uq.undo();	→3←
uq.undo();	→3 , 2←
uq.undo();	→2←
uq.undo();	الصفّ–**إلغاء** فارغ
uq.undo();	الصفّ–**إلغاء** فارغ



أ. أمامك قطعة كود.

```
UndoQueue q = new UndoQueue();
q.insert(1);
q.insert(2);
q.insert(3);
q.remove();
q.insert(4);
q.undo();
q.undo();
```

بيّن وضع الصفّ–**إلغاء** بعد تنفيذ كلّ أمر في قطعة الكود.

ب. اكتب بلغة Java عنوان الفئة **UndoQueue** ، وصفاتها. اكتب توثيقًا لكلّ صفة.

جـ. طبِّق بلغة Java العمليات insert وَ remove وَ undo المعروضة في واجهة تطبيق الفئة **UndoQueue**.

بإمكانك استعمال عمليات واجهة التطبيق الأخرى التابعة لِـ **UndoQueue** بدون تطبيقها.
إذا استعملتَ عمليات إضافية، عليك تطبيقها.

/ يتبع في صفحة 12 /

للطلّاب الذين يحلّون بلغة C#

أمامك واجهة تطبيق الفئة "**صفّ-إلغاء**" (**UndoQueue**) :

تعرّف الفئة نمط مجمّع مع بروتوكول FIFO لإدخال وإخراج قيَم صحيحة وأكبر من 0 .

الوصف	اسم العملية
تبني العملية **صفّ-إلغاء** فارغًا.	UndoQueue()
تُعيد العملية true إذا كان ال**صفّ-إلغاء** الحالي فارغًا، خلاف ذلك – تُعيد العملية false .	bool IsEmpty()
تُدخِل العملية القيمة x في نهاية ال**صفّ-إلغاء** الحالي.	void Insert(int x)
تُخرِج العملية القيمة التي في رأس ال**صفّ-إلغاء** الحالي، وتُعيدها. فرضية: ال**صفّ-إلغاء** الحالي ليس فارغًا.	int Remove()
تُعيد العملية قيمة الحدّ الذي في رأس ال**صفّ-إلغاء** بدون إخراجه. فرضية: ال**صفّ-إلغاء** الحالي ليس فارغًا.	int Head()
تُلغي العملية عملية الـ Remove أو عملية الـ Insert الأخيرة التي نُفِّذت، وتسترجع ال**صفّ-إلغاء** كما كان قبل تنفيذ العملية. إذا لم يكن هناك عمليتا Remove أو Insert تمّ تنفيذهما، لا تفعل العملية شيئًا. انتبه: تشغيل العملية Undo، n مرّات، يُعيد ال**صفّ-إلغاء** إلى الوضع الذي كان عليه قبل تنفيذ n العمليات Remove أو Insert الأخيرة التي لم يتمّ إلغاؤها بعد.	void Undo()

/ يتبع في صفحة 13 /

مثال:

العملية	وضع الصفّ–**إلغاء** بعد تنفيذ العملية
UndoQueue uq = new UndoQueue();	الصفّ–**إلغاء** فارغ
uq.Insert(2);	→2→
uq.Insert(3);	→3 , 2→
uq.Remove();	→3→
uq.Insert(4);	→4 , 3→
uq.Remove();	→4→
uq.Remove();	الصفّ–**إلغاء** فارغ
uq.Undo();	→4→
uq.Undo();	→4 , 3→
uq.Insert(1);	→1 , 4 , 3→
uq.Undo();	→4 , 3→
uq.Undo();	→3→
uq.Undo();	→3 , 2→
uq.Undo();	→2→
uq.Undo();	الصفّ–**إلغاء** فارغ
uq.Undo();	الصفّ–**إلغاء** فارغ

أ. أمامك قطعة كود.

```
UndoQueue q = new UndoQueue();
q.Insert(1);
q.Insert(2);
q.Insert(3);
q.Remove();
q.Insert(4);
q.Undo();
q.Undo();
```

بيّن وضع الصفّ–**إلغاء** بعد تنفيذ كلّ أمر في قطعة الكود.

ب. اكتب بلغة C# عنوان الفئة **UndoQueue** ، وصفاتها. اكتب توثيقًا لكلّ صفة.

جـ. طبِّق بلغة C# العمليات Insert وَ Remove وَ Undo المعروضة في واجهة تطبيق الفئة **UndoQueue**.

بإمكانك استعمال عمليات واجهة التطبيق الأخرى التابعة لِـ **UndoQueue** بدون تطبيقها.
إذا استعملتَ عمليات إضافية، عليك تطبيقها.

/ يتبع في صفحة 14 /

٤. أ. أمامك عنوان عملية:

بلغة Java :

```
public static void leaves(BinTreeNode<Integer> t , Stack<Integer> s)
```

بلغة C# :

```
public static void Leaves(BinTreeNode<int> t , Stack<int> s)
```

تتلقّى العملية شجرة بينارية غير فارغة t لأعداد صحيحة وراصّة فارغة s لأعداد صحيحة.

تُدخِل العملية قيَم جميع أوراق الشجرة t إلى الراصّة، حسب ترتيب التمشيط من اليمين إلى اليسار.

طبِّق العملية بلغة Java أو بلغة C# .

ب. اكتب بلغة Java أو بلغة C# عملية بوليانية تتلقّى شجرتين بينارييتين غير فارغتين لأعداد صحيحة وتُعيد true إذا كان الشرطان التاليان يتحقّقان:

- للشجرتين نفس عدد الأوراق
- حسب ترتيب التمشيط من اليمين إلى اليسار، قيَم الأوراق متساوية

خلاف ذلك – تُعيد العملية false .

عليك استعمال العملية التي طبّقتَها في البند "أ".

/ يتبع في صفحة 15 /

الفصل الثاني (٥٠ درجة)

في هذا الفصل أسئلة في أربعة مسارات:

أنظمة حاسوب وأسمبلر، ص 15-20

مقدّمة لبحث الأداءات، ص 21-31

موديلات حسابية، ص 32-34

برمجة موجّهة كائنات بلغة Java، ص 36-45؛ برمجة موجّهة كائنات بلغة #C ، ص 46-55.

أجب فقط عن سؤالين في المسار الذي تعلّمتَه.

أنظمة حاسوب وأسمبلر

إذا تعلّمتَ هذا المسار، أجب عن اثنين من الأسئلة ٥–٨ (لكلّ سؤال – ٢٥ درجة).

٥. في راصّة معيّنة مخزونة 10 أعداد صحيحة مع إشارة، بكِبَر كلمة.
بالإضافة إلى ذلك، معرَّف في مقطع المعطيات المتغيّر K والمصفوفتان، ARR1 وَ ARR2 .
جميع خلايا المصفوفة ARR1 مبتدَأة لِـ 0 ، والمتغيّر K مبتَدأ لِـ 0 .

```
ARR1    DW    10    DUP (0)
ARR2    DW    10    DUP (?)
K       DW    0
```

يجب نسخ جميع الأعداد المخزونة في الراصّة إلى المصفوفة ARR1 على النحو التالي:
تُخزَن جميع الأعداد السالبة في بداية المصفوفة، حسب ترتيب إخراجها من الراصّة. وتُخزَن بعد الأعداد السالبة جميع الأصفار، وبعد ذلك تُخزَن جميع الأعداد الموجبة، حسب ترتيب إخراجها من الراصّة.

على سبيل المثال، بالنسبة للراصّة التي أمامك التي تحوي 6 حدود:

−5
0
10
−2
0
8

تبدو المصفوفة ARR1 على النحو التالي:

0	1	2	3	4	5
−5	−2	0	0	10	8

/ يتبع في صفحة 16 /

أمامك قطعة برنامج بلغة أسمبلر، تنفّذ المطلوب.

حُذفت من قطعة البرنامج أسطر في الأماكن المرقّمة بالأرقام i-iv .

اكتب في دفترك أرقام الأسطر المرقّمة، واكتب سطر الأمر الناقص بجانب كلّ واحد من الأرقام، كي تنفّذ قطعة البرنامج المطلوب.

```
            MOV     SI , 0
            MOV     DI , 0
            MOV     CX , 10
      A1:   POP     AX
i           _____   _______
            JLE     A2
ii          _____   _______
            ADD     DI , 2
            JMP     A4
      A2:   CMP     AX , 0
            JNE     A3
            INC     K
            JMP     A4
iii   A3:   _____   _______
            ADD     SI , 2
      A4:   LOOP    A1
            ADD     SI , K
            ADD     SI , K
            MOV     CX , DI
            SHR     CX  , 1
            XOR     BX , BX
      A5:   MOV     DX , ARR2[BX]
            MOV     ARR1[SI] , DX
iv          _____   _______
            ADD     BX , 2
            LOOP    A5
```

/ يتبع في صفحة 17 /

٦. تُسمّى المصفوفة A **مشمولة** في المصفوفة B، إذا كان طول المصفوفة A لا يزيد عن طول المصفوفة B ، وجميع قيَم حدود المصفوفة A موجودة في المصفوفة B بالتسلسل وبنفس الترتيب كما في المصفوفة A.

في كلّ مصفوفة مخزونة أعداد صحيحة مع إشارة، تختلف عن بعضها البعض.

مثال:

بالنسبة للمصفوفتين A وَ B اللتين أمامك، المصفوفة A **مشمولة** في المصفوفة B .

المصفوفة A :

0	1	2	3	4	5
3	−5	0	8	5	11

المصفوفة B :

0	1	2	3	4	5	6	7	8	9
15	−7	4	3	−5	0	8	5	11	2

حدود المصفوفة B المشار إليها باللون الرمادي هي قيَم حدود المصفوفة A .

أ. عُرِّفت المعطيات في مقطع المعطيات على النحو التالي:

```
ARR_B   DB  100   DUP (?)
ARR_A   DB  10    DUP (?)
V       DB  ?
P       DB  ?
```

اكتب بلغة أسمبلر برنامجًا فرعيًا باسم TEST يتلقّى قيمة مخزونة في المتغيّر V.
يفحص البرنامج الفرعي إذا كانت القيمة المخزونة في المتغيّر V موجودة في المصفوفة ARR_B .
إذا كانت موجودة – يخزن البرنامج الفرعي مؤشّر الحدّ الملائم في المتغيّر P ،
خلاف ذلك – يخزن البرنامج الفرعي القيمة 1- في المتغيّر P .

ب. اكتب بلغة أسمبلر قطعة برنامج تفحص إذا كانت المصفوفة ARR_A **مشمولة** في المصفوفة ARR_B .
إذا كانت **مشمولة** – تخزن قطعة البرنامج 1 في الخازن BL، خلاف ذلك – تخزن 0 في الخازن BL. عليك استعمال البرنامج الفرعي TEST الذي كتبتَه في البند "أ" .
افترض أنّ طول المصفوفة ARR_A لا يزيد عن طول المصفوفة ARR_B .

/ يتبع في صفحة 18 /

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

أ. عُرِّفت في مقطع المعطيات المصفوفة ARR :

```
ARR    DB 6    DUP (?)
```

يجب قلب ترتيب الحدود في المصفوفة.

على سبيل المثال، معطاة المصفوفة:

0	1	2	3	4	5
3	2	10	0	6	7

بعد قلب ترتيب الحدود، تبدو المصفوفة على النحو التالي:

0	1	2	3	4	5
7	6	0	10	2	3

أمامك قطعتان بلغة أسمبلر، i-ii ، هدفهما قلب ترتيب الحدود في المصفوفة ARR .

تتبّع بواسطة جدول متابعة تنفيذ كلّ واحدة من القطعتين i-ii اللتين أمامك، وحدِّد إذا كانت تنفّذ المطلوب أم لا تنفّذ المطلوب.

i

```
        MOV  SI , 3
        MOV  DI , 2
A1:     MOV  AL , ARR[DI]
        MOV  AH , ARR[SI]
        ADD  ARR[DI] , AH
        SUB  ARR[DI] , AL
        MOV  ARR[SI] , AL
        INC  SI
        DEC  DI
        JNZ  A1
```

---

ii

```
        MOV  CX , 3
        MOV  SI , 3
        MOV  DI , 2
A1:     MOV  AL , ARR[DI]
        MOV  AH , ARR[SI]
        MOV  ARR[DI] , AH
        MOV  ARR[SI] , AL
        INC  SI
        DEC  DI
        LOOP A1
```

/ يتبع في صفحة 19 /

ب. (لا توجد له علاقة مع البند "أ" .)

عُرِّفت المعطيات في مقطع المعطيات على النحو التالي :

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

انتبه: الأعداد هي أعداد عشرية.

أمامك قطعة برنامج بلغة أسمبلر 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.

(١) 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.

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

/ يتبع في صفحة 20 /

٨. أمامك قطعة برنامج بلغة أسمبلر:

```
i    PUSHF
ii   PUSH   BP
iii  MOV    BP , SP
iv   MOV    AX , [BP+4]
v    MOV    BP , [BP]
vi   ADD    SP , 4
vii  POPF
```

معطى أنّه قبل تنفيذ قطعة البرنامج:

- محتوى خازن أوضاع الأعلام هو 3202H
- محتوى الخازن SP هو 96H
- محتوى الخازن BP هو 5678H

أمامك جزء من راصّة تحوي أعدادًا سداسية عشرية.
قبل تنفيذ قطعة البرنامج، يشير الخازن SP إلى الخلية المشار إليها بالسهم، التي عنوانها النسبي 96H .

.....	*	*	*	*	*	*	02H	32H	*	*
إزاحات عناوين الخلايا	90H	91H	92H	93H	94H	95H	96H	97H	98H	99H

الخلية المشار إليها بـ * هي خلية لا أهمّية لمحتواها.

أ. تتبّع بواسطة جدول متابعة تنفيذ قطعة البرنامج.
يجب أن يشمل جدول المتابعة أعمدة بالنسبة للخوازن: AX , BP , SP .

ب. حسب جدول المتابعة، أجب عن البنود الفرعية (١)–(٤).

(١) أيّ عدد يُسجَّل في الراصّة المعطاة بعد تنفيذ الأمر i ؟
اكتب العنوان النسبي للخلايا التي يُسجَّل فيها العدد.

(٢) ماذا يكون محتوى الخازن BP بعد تنفيذ الأمر iii ؟

(٣) ماذا يكون محتوى الخازن AX بعد تنفيذ الأمر iv ؟

(٤) ماذا يكون محتوى الخازن BP بعد تنفيذ الأمر v ؟

/ يتبع في صفحة 21 /

مقدّمة لبحث الأداءات

إذا تعلّمتَ هذا المسار، أجب عن اثنين من الأسئلة ٩–١٢ (لكلّ سؤال – ٢٥ درجة).

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

أ. ليكن $G = (V, E)$ رسمًا بيانيًا موجّهًا، V هي مجموعة الرؤوس وَ E هي مجموعة الأقواس. G ممثَّل بواسطة مصفوفة المُجاوَرات التي أمامك:

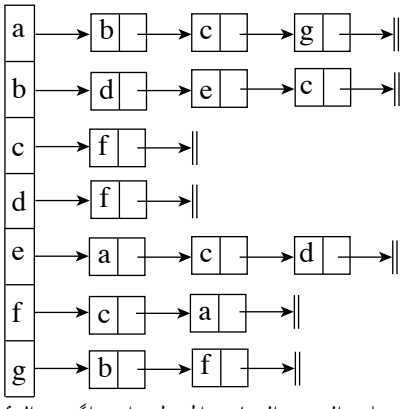
	a	**b**	**c**	**d**	**e**
a	0	1	1	0	0
b	0	0	1	0	0
c	0	1	0	1	0
d	0	0	0	0	1
e	1	0	1	0	0

(١) ارسم الرسم البياني G الممثَّل بواسطة المصفوفة.

(٢) جد مركِّب/مركِّبات الربط القويّ الذي/التي في الرسم البياني G.
بالنسبة لكلّ مركِّب ربط قويّ وجدتَه، اكتب مجموعة رؤوسه.

(٣) أَزِل قوسًا واحدًا من الرسم البياني G الذي رسمتَه، بحيث يكبر عدد مركِّبات الربط القويّ بـ 1. اذكر القوس الذي أزلتَه، واكتب بالنسبة لكلّ مركِّب ربط قويّ مجموعة رؤوسه.

ب. ليكن $G = (V, E)$ رسمًا بيانيًا موجّهًا، V هي مجموعة الرؤوس وَ E هي مجموعة الأقواس. G ممثَّل بواسطة قائمة المُجاوَرات التالية:



(١) فَعِّل ألغوريثمًا لتمشيط عميق (DFS) على الرسم البياني المعطى ابتداءً من الرأس a. ارسم في دفترك الشجرة الامتدادية DFS التي تنتج.

(٢) فَعِّل ألغوريثمًا لتمشيط عرضي (BFS) على الرسم البياني المعطى ابتداءً من الرأس d. ارسم في دفترك الشجرة الامتدادية BFS التي تنتج.

/ يتبع في صفحة 22 /

جـ. ليكن G رسمًا بيانيًا موجّهًا.

نفعّل ألغوريثمًا لتمشيط عميق (DFS) على الرسم البياني الموجَّه G .

نعرّف أقواس الرسم البياني G بمساعدة الشجرة الامتدادية / الغابة الامتدادية التي نتجت.

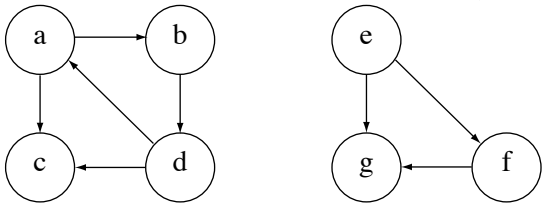
قوس الشجرة هو قوس للرسم البياني G ، موجود أيضًا في الشجرة الامتدادية / في الغابة الامتدادية التابعة لـ G.

القوس الخلفي هو قوس (u , v) في الرسم البياني G، غير موجود في الشجرة الامتدادية / في الغابة الامتدادية، وهو قوس من الرأس u إلى الرأس v في الرسم البياني G ، بحيث v هو أب قديم لـ u في الشجرة الامتدادية / في الغابة الامتدادية.

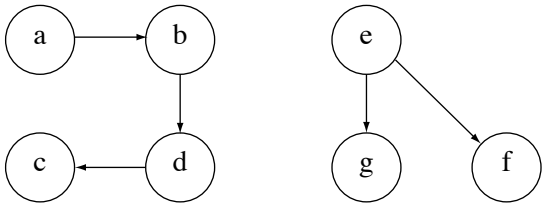
القوس الأمامي هو قوس (u , v) في الرسم البياني G ، غير موجود في الشجرة الامتدادية / في الغابة الامتدادية، وهو قوس من الرأس u إلى الرأس v في الرسم البياني G ، بحيث v هو فرد نسل لـ u في الشجرة الامتدادية / في الغابة الامتدادية.

القوس العابر هو قوس (u , v) في الرسم البياني G ، غير موجود في الشجرة الامتدادية / في الغابة الامتدادية، وهو ليس **قوسًا خلفيًا** أو **قوسًا أماميًا**، وهو قوس من الرأس u إلى الرأس v في الرسم البياني G ، بشرط أن لا يكون أيّ رأس من هذين الرأسين فرد نسل للرأس الآخر في الشجرة الامتدادية / في الغابة الامتدادية.

(١) أمامك الرسم البياني الموجَّه G ، الذي يحوي أكثر من مركِّب ربط قويّ واحد.



بعد تفعيل ألغوريثم التمشيط العميق (DFS) على الرسم البياني الموجَّه G المعطى، نتجت الغابة الامتدادية التي أمامك:



/ يتبع في صفحة 23 /

انسخ الرسم البياني G إلى دفترك. اعتمد على الغابة الامتدادية المعطاة، واكتب بجانب كلّ واحد من أقواس الرسم البياني G إذا كان:

- **قوس شجرة**
- **قوسًا خلفيًا**
- **قوسًا أماميًا**
- **قوسًا عابرًا**

(٢) أمامك ادّعاء فيه تعبير ناقص في المكان المشار إليه بخطّ.

إذا كان بعد تفعيل الألغوريثم تمشيط عميق (DFS) على الرسم البياني الموجَّه G يوجد في G _____ ، عندها توجد في G دائرة/دوائر.

أمامك أربعة تعابير ممكنة لإكمال الادّعاء:

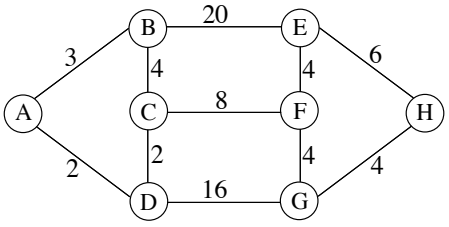
- **قوس شجرة**
- **قوس خلفي**
- **قوس أمامي**
- **قوس عابر**

انسخ إلى دفترك الإمكانية الملائمة لإكمال الادّعاء، وعلّل تحديدك.

/ يتبع في صفحة 24 /

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

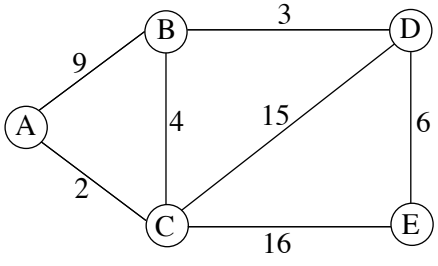
أ. أمامك الشبكة $G = (V, E)$ ، V هي مجموعة الرؤوس وَ E هي مجموعة الأقواس:



استعمِل الألغوريثم Prim لإيجاد شجرة امتدادية صغرى، وارسم الشجرة الامتدادية الصغرى التي تنتج بالنسبة للشبكة المعطاة، ابتداءً من الرأس C .

ب. (لا توجد له علاقة مع البند "أ".)

معطاة الشبكة:



(١) استعمِل الألغوريثم Kruskal لإيجاد شجرة امتدادية صغرى، وارسم الشجرة الامتدادية الصغرى للشبكة المعطاة.

(٢) اقترح أحد الطلّاب للمعلّمة ألغوريثمًا لإيجاد شجرة امتدادية صغرى لشبكة $G = (V, E)$ ، بحيث V هي مجموعة الرؤوس وَ E هي مجموعة أقواس الشبكة.

أمامك الألغوريثم الذي اقترحه الطالب.

الخطوة 1: كوِّن رسمًا بيانيًا G_1 ، مجموعة رؤوسه هي مجموعة رؤوس الشبكة G ومجموعة أقواسه، E_T ، هي مجموعة فارغة.

الخطوة 2: رتِّب أقواس الشبكة G بترتيب تصاعدي، حسب الأوزان المنسوبة لها. تنتج مجموعة أقواس مرتَّبة نرمز إليها بِـ E^* .

/ يتبع في صفحة 25 /

الخطوة 3: لكلّ قوس (u , v) في E^* ، ابتداءً من القوس الذي وزنه هو الأصغر، نفِّذ:

إذا كانت إضافة القوس (u , v) لا تكوّن دائرة في G_1 – أضِف القوس (u , v) إلى E_T .

خلاف ذلك – أعِد الرسم البياني G_1 الذي هو شجرة امتدادية صغرى.

الخطوة 4: أعِد G_1 الذي هو شجرة امتدادية صغرى.

فعِّل الألغوريثم الذي اقترحه الطالب على الشبكة المعطاة.

أجب عن البنود (i)-(iv) .

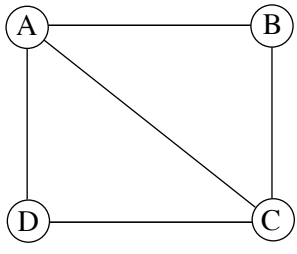
(i) ارسم في دفترك G_1 الذي تكوّن بعد تنفيذ الخطوة 1.

(ii) اكتب في دفترك جميع الأقواس التي في E^* بعد تنفيذ الخطوة 2.

(iii) ارسم في دفترك كلّ واحد من الرسوم البيانية G_1 التي نتجت خلال الخطوة 3.

(iv) ارسم في دفترك الشجرة الامتدادية التي أعادها الألغوريثم.

(٣) ردّت المعلّمة على الطالب بأنّ الألغوريثم الذي اقترحه خاطئ، لأنّ مُخرَجه ليس بالضرورة شجرة امتدادية صغرى. على سبيل المثال، بالنسبة للشبكة التالية:



يمكن نَسْب أوزان لأقواس الشبكة بحيث لو استعملنا الألغوريثم الذي اقترحه الطالب لا تُعاد الشجرة الامتدادية الصغرى.

انسخ الشبكة إلى دفترك، واكتب أوزانًا بجانب الأقواس، بحيث لا تكون الشجرة التي تُعاد بواسطة الألغوريثم الذي اقترحه الطالب شجرة امتدادية صغرى.

/ يتبع في صفحة 26 /

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

أ. معطاة مسألة تخطيط خطّي: $\max\{z = kx_1 + 20x_2\}$

تخضع للاضطرارات التالية:

$$-3x_1 + 4x_2 \le 24$$

$$x_1 + 2x_2 \ge 0$$

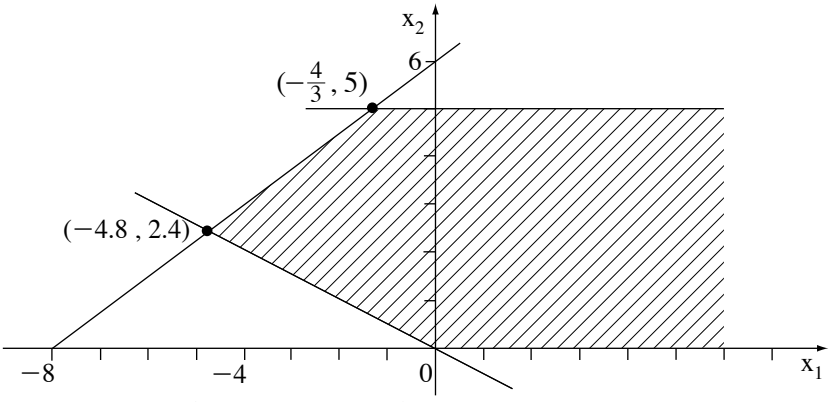
$$x_2 \le 5$$

$$x_2 \ge 0$$

x_1 ليس مقيّدًا بإشارة.

k هو بارامتر.

أمامك رسم لمجال الحلول الممكنة للمسألة المعطاة.



أمامك أربعة بنود فرعية (١)–(٤)، معطاة في كلّ واحد منها قيمة معيّنة للبارامتر k.

(١) $k = 20$.

(٢) $k = -20$.

(٣) $k = -15$.

(٤) $k = -15$، ونضيف الاضطرار: $x_1 \le -5$.

البنود الفرعية لا تتعلّق ببعضها البعض.

بالنسبة لكلّ واحد من البنود الفرعية (١) – (٤):

- إذا كان الحلّ الأمثل وحيدًا، عليك إيجاد الحلّ الأمثل الوحيد، وقيمة دالة الهدف في هذا الحلّ.
- إذا كان هناك لانهاية من الحلول المثلى، عليك كتابة الحلّ الأمثل العامّ للمسألة، وقيمة دالة الهدف في هذا الحلّ.
- إذا كان الحلّ الأمثل ليس محصورًا، علّل تحديدك.
- إذا لم يكن هناك حلّ ممكن، علّل تحديدك.

/ يتبع في صفحة 27 /

ب. (لا توجد له علاقة مع البند "أ".)

معطاة مسألة تخطيط خطّي:

$$\max\{z = ax_1 + 3x_2\}$$

تخضع للاضطرارات التالية:

$$x_1 + \frac{3}{8}x_2 \le 3$$

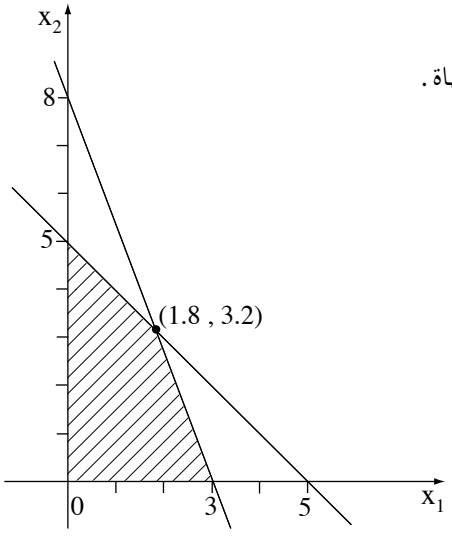
$$x_1 + x_2 \le 5$$

$$x_1 \ge 0$$

$$x_2 \ge 0$$

$a > 0$ وهو بارامتر للمسألة.

أمامك رسم لمجال الحلول الممكنة للمسألة المعطاة.



أمامك أربعة مجالات، i-iv ، بالنسبة للبارامتر a :

i $a < 3$

ii $a > 8$

iii $3 < a < 8$

iv $3 < a < 24$

حدّد بالنسبة لأيّ مجال من المجالات i-iv ، يكون الحلّ (1.8 , 3.2) حلّاً أمثل **وحيدًا**.
انسخ المجال إلى دفترك وعلّل تحديدك.

/ يتبع في صفحة 28 /

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

أ. في الجدول الذي أمامك معطى جزء من حلّ أساسي ممكن لمسألة نقل: $x_{11}=80$، $x_{21}=20$.

المصادر	الغايات			العرض
	1	2	3	
1	10 (80)	15	17	80
2	10 (20)	18	14	120
3	15	20	18	50
الطلب	100	100	50	

انسخ الجدول إلى دفترك، وأكمله حسب طريقة الزاوية الشمالية الغربية.

ب. في الجدول الذي أمامك معطى جزء من حلّ أساسي 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	10 (80)	15	12	80	0
2	10	18 (100)	10	100	3
3	20	10	14 (50)	50	−5
الطلب	80	100	50		
v_j	10	15	19		

انسخ الجدول إلى دفترك، وأكمله حسب طريقة الزاوية الشمالية الغربية مع الأخذ بالحسبان القيم المتسلسلة لـِ u_i والقيم المتسلسلة لـِ v_j، بحيث ينتج حلّ أساسي ممكن.

/ يتبع في صفحة 29 /

جـ. في الجدول الذي أمامك معطى حلّ أساسي 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	10 (100)	18	13 (30)	130	0
2	6 (100)	21	12	100	−4
3	11	16 (50)	14 (100)	150	1
الطلب	200	50	130		
v_j	10	15	13		

هل الحلّ هو حلّ أمثل؟ علّل إجابتك.

/ يتبع في صفحة 30 /

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

المصادر	الغايات			العرض	u_i
	1	2	3		
1	10 (20)	25	16	20	10
2	10 (30)	22	14 (20)	50	10
3	18	20 (40)	20 (20)	60	16
الطلب	50	40	40		
v_j	0	4	4		

(١) هل الحلّ المعطى هو حلّ أمثل وحيد؟ علّل إجابتك.

(٢) في الجدول المعطى في بداية البند، أنقِص 4 من التكاليف للوحدة من كلّ مصدر إلى كلّ غاية. ارسم الجدول الذي ينتج.

حدّد أيّ قول من الأقوال الأربعة i -iv التي أمامك هو الصحيح.

انسخه إلى دفترك وعلّل اختيارك.

i الحلّ المعطى ليس حلّاً ممكنًا بالنسبة للمسألة التي نتجت.

ii الحلّ المعطى هو حلّ أساسي ممكن لكنّه ليس حلّاً أمثل بالنسبة للمسألة التي نتجت.

iii الحلّ المعطى هو حلّ أمثل وحيد بالنسبة للمسألة التي نتجت.

iv الحلّ المعطى هو حلّ أمثل لكنّه ليس حلّاً أمثل وحيدًا بالنسبة للمسألة التي نتجت.

/ يتبع في صفحة 31 /

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

المصادر	الغايات				العرض	u_i
	1	2	3	4		
1	5 (150)	2 (80)	4 −4	2 −6	230	0
2	2 (70)	3 4	5 (80)	3 −2	150	−3
3	7 7	4 7	3 (200)	3 (340)	540	−5
الطلب	220	80	280	340		Z = 3070
v_j	5	2	8	8		

الحلّ الموصوف في هذا الجدول ليس حلّاً أمثل، لأنّ هناك متغيّرات من خارج القاعدة بحيث قيمة الـ $(c_{ij} - u_i - v_j)$ الخاصّة بها سالبة.

عليك تنفيذ تكرار إضافي، أي تكرار $k + 1$.

(١) ما هو المتغيّر الذي سيخرج من القاعدة في هذا التكرار؟

(٢) ابنِ في دفترك جدولاً جديدًا، واكتب فيه الحلّ الذي سينتج بعد هذا التكرار.

/ يتبع في صفحة 32 /

موديلات حسابية

إذا تعلّمتَ هذا المسار، أجب عن اثنين من الأسئلة ١٣–١٦ (لكلّ سؤال – ٢٥ درجة).

١٣. أمامك اللغة L فوق الأبجدية $\{a, b, c\}$:

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

أ. اكتب أقصر كلمة في اللغة L .

ب. ابنِ أوتومات راصّة يتلقّى اللغة L .

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

أ. أمامك اللغة L فوق الأبجدية $\{a, b\}$:

$L = \{w \mid |w| > 0$، { الرمز الأخير في w يظهر في كلّ الكلمة عددًا زوجيًا من المرّات

مثال:

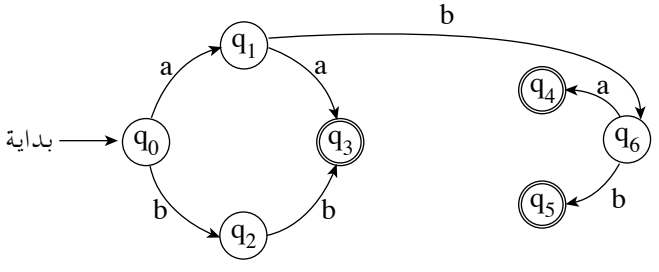
الكلمة abaabbab تتبع للّغة.

الكلمة abbabaa تتبع للّغة.

الكلمة babbaba لا تتبع للّغة.

أمامك رسم جزئي لأوتومات نهائي محدود يتلقّى اللغة L .

هناك انتقالات وإشارات إدخال ناقصة في الرسم.



يحوي الرسم جميع حالات الأوتومات، وجميع الحالات المتلقِّية.

انسخ الرسم إلى دفترك، وأكمله بحيث يتلقّى الأوتومات اللغة L .

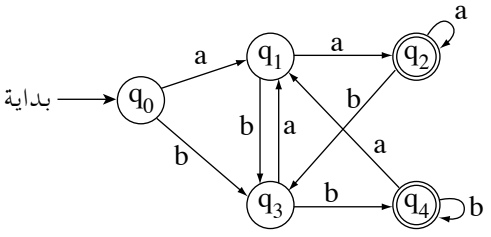
عليك إكمال الانتقالات الناقصة وإشارات الإدخال الناقصة.

انتبه: لا تُضِف حالات إلى الأوتومات ولا تُنقِص منه حالات.

/ يتبع في صفحة 33 /

ب. (لا توجد له علاقة مع البند "أ".)

أمامك أوتومات نهائي محدود يتلقّى اللغة L فوق الأبجدية {a , b} .



(١) حدّد بالنسبة لكلّ واحدة من أربع الكلمات (i)-(iv) التي أمامك إذا كان الأوتومات يتلقّاها. إذا كان الأوتومات يتلقّاها، اكتب المسار المتلقّي بالنسبة لهذه الكلمة.

(i) aaba

(ii) bbaabb

(iii) abaa

(iv) bb

(٢) ما هي اللغة L المعرَّفة بواسطة الأوتومات؟

١٥. أمامك اللغة L فوق الأبجدية {0 , 1 , 2} :

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

أ. برهن أنّ اللغة L غير نظامية.

ب. معطاة اللغة L_1 فوق الأبجدية {1 , 2} :

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

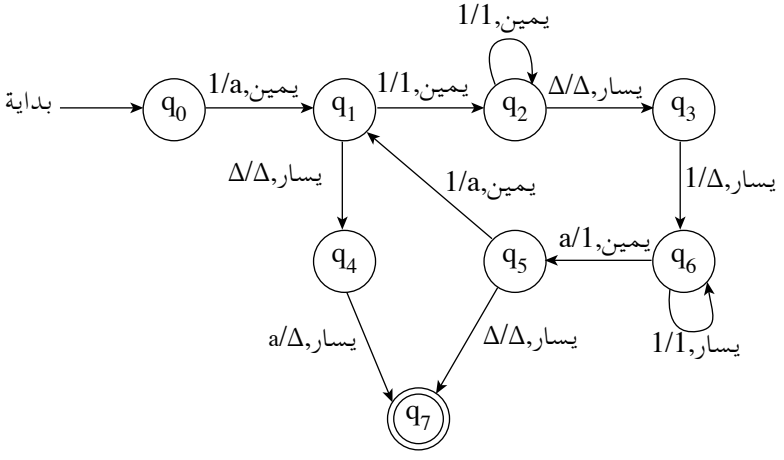
ما هي اللغة $L \cap L_1$ ؟ علّل.

/ يتبع في صفحة 34 /

١٦. أمامك آلة تيورينج تحسب الدالة f(x) .

تتلقّى الآلة كمدخَل عددًا x صحيحًا وأكبر من 0 ، مسجّلاً على الشريط كعدد أوناري بواسطة x رموز من 1 ، وبعدها الإشارة Δ .

تسجّل الآلة نتيجة حساب f(x) على الشريط كعدد أوناري بعد الإشارة ⊢ ، وتظهر بعد العدد الإشارة Δ .



أ. ماذا يحوي الشريط بعد حساب f(3)؟ بيّن مسار حساب الآلة.

اكتب حالة الشريط في كلّ مرحلة، وأشر أين يتواجد رأس الآلة، واكتب في أيّ من الحالات q_0 - q_7 تتواجد الآلة.

ب. ماذا يحوي الشريط بعد حساب f(5) ؟

جـ. ماذا يحوي الشريط بعد حساب f(6) ؟

د. ما هي الدالة f(x) التي تحسبها الآلة؟

هـ. أضِف إلى الآلة انتقالاً/انتقالات كي تحسب f(0) أيضًا.

اكتب في دفترك الانتقال الذي أضفتَه/الانتقالات التي أضفتَها.

اكتب بالنسبة لكلّ انتقال: من أيّة حالة يخرج، وإلى أيّة حالة يصل، وماذا يُسجَّل عليه.

/ يتبع في صفحة 35 /

انتبه: برمجة موجّهة كائنات تبدأ في صفحة 36.

برمجة موجّهة كائنات

إذا تعلّمتَ هذا المسار وتكتب بلغة Java، أجب عن اثنين من الأسئلة ١٧–٢٠.

(لكلّ سؤال – ٢٥ درجة).

١٧. 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** عملية بوليانية 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 الحالي – تُعيد العملية true، خلاف ذلك – تُعيد false.

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

/ يتبع في صفحة 37 /

جـ. أمامك قطعة من عملية رئيسية:

```
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 للعملية isLike تُفعِّل – 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 عملية رئيسية:

```
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 للعملية isLike تُفعِّل – 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 عملية خارجية باسم longString تتلقّى مصفوفة لكائنات من نمط Object .

تُعيد العملية 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 38 /

١٨. أمامك مشروع فيه الفئات: **B** وَ **D** وَ **A** وَ **OopTest** :

```
public class B
{
    private static int numB = 0;
    private int m1;
    private int m2;

    public B(int m1, int m2)
    {
        this.m1 = m1;
        this.m2 = m2;
        numB++;
        System.out.println("B(" + m1 + ", " + m2 + ") ,  #" + numB);
    }
}

public class D extends B
{
    private static int numD = 0;
    private double d;

    public D(double d, int x)
    {
        super(x, x);
        this.d = d;
        numD++;
        System.out.println("D(" + d + ", " + x + ") ,  #" + numD);
    }
    public D(double d, int x, int y)
    {
        super(x , y);
        this.d = d;
        numD++;
        System.out.println("D(" + d +", "+ x + " , " + y + ") ,  #" + numD);
    }
}
```

/ يتبع في صفحة 39 /

```
public class A
{
    private static int  numA = 0;
    private A  a;
    private B  b;

    public A(A  a , B  b)
    {
        this.a = a;
        this.b = b;
        numA++;
        System.out.println("A Constructor ,  #" + numA);
    }
}

public class OopTest
{
    public static void main(String [] args)
    {
        B   w1 = new B(2, 3);
        B   w2 = new D(1.5 , 6);
        B   w3 = new D(2.3 , 8 , 9);
        A   w4 = new A(null , w1);
        A   w5 = new A(w4 , w3);
    }
}
```

اكتب متابعة للعملية main في الفئة **OopTest** ، واكتب المُخرَج.

يجب أن تكتب في المتابعة قيَم المتغيّرات، وبالنسبة لكلّ كائن – قيَم صفاته.

/ يتبع في صفحة 40 /

١٩. طوّرت شركة لتوزيع برامج الحاسوب برنامج حاسوب يتناول متواليات الأعداد الصحيحة.
تمّ تطوير المنظومة على مراحل.
بالنسبة لكلّ متوالية أعداد، يتطرّقون إلى:
(١) الحدّ الأوّل في المتوالية الذي رقمه التسلسلي هو 1.
(٢) الحدّ الذي رقمه التسلسلي في المتوالية هو n .
(٣) طباعة n الحدود الأولى في المتوالية.

طُوّرت **في المرحلة الأولى** فئتان:
متوالية حسابية (**ASeq**) – متوالية الفرق فيها بين كلّ حدّ والحدّ الذي قبله هو قيمة ثابتة.
متوالية هندسية (**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 41 /

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

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:

(١) الحدّ الأوّل في المتوالية الذي رقمه التسلسلي هو 1.

(٢) الحدّ الذي رقمه التسلسلي في المتوالية هو n .

(٣) 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 عملية 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 43 /

تَقَرَّرَ **في المرحلة الرابعة** من التطوير، تطوير عملية ستاتيّة check تتلقّى قيمة صحيحة n ، وكائنين لمتواليات: أحدهما من نمط **ASeq** والآخر من نمط **GSeq**.
تحسب العملية 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 العملية الستاتيّة check حسب المتطلّبات التي عُرِّفت في المرحلة الرابعة من التطوير.

/ يتبع في صفحة 44 /

٢٠. تركّز عيادة بيطرية للحيوانات البيتية معلومات تتعلّق بالأطبّاء البيطريين الذين يعملون في العيادة وبالحيوانات البيتية التي تتلقّى العلاج فيها. عدد الأطبّاء البيطريين الذين يعملون في العيادة هو 10 على الأكثر، وعدد الحيوانات البيتية التي تُعالَج في العيادة هو 500 على الأكثر.

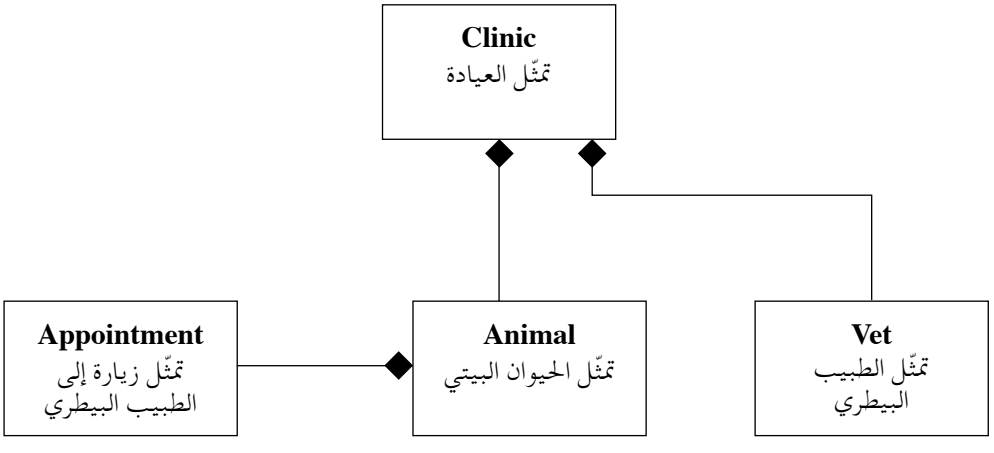
النقاط الموجَّهة في معالجة المعلومات هي:

- يُحفَظ لكلّ طبيب بيطري: رقم الهوية والاسم والأقدمية في العمل بالسنوات.
- يُحفَظ لكلّ حيوان بيتي: رقم الرخصة والاسم والنوع (مثلاً: كلب أو قطّة أو أرنب)، والعمر، وتفصيل حتّى 50 الزيارات الأخيرة في العيادة وعدد الزيارات المحفوظة.
- يُحفَظ لكلّ زيارة إلى العيادة: رقم هوية الطبيب البيطري الذي عالج الحيوان البيتي في هذه الزيارة، ونصّ من رموز يحوي أكواد العلاجات التي حصل عليها الحيوان البيتي في تلك الزيارة. كود العلاج هو حرف كبير من بين جميع أحرف الأبجدية ABC .

جزء من العمليات التي تستطيع المنظومة تنفيذها هي:

- إعداد تقرير عن جميع الحيوانات البيتية من نوع معيّن (مثلاً جميع الأرانب). يشمل التقرير بالنسبة لكلّ حيوان رقم رخصته واسمه وعمره.
- في بداية كلّ سنة: تعديل عمر كلّ حيوان بيتي، أي تكبير العمر بـِ 1 ، وتعديل أقدمية كلّ طبيب بيطري، أي تكبير الأقدمية بـِ 1 .
- إعادة اسم الطبيب البيطري حسب رقم هويته.

أمامك مخطّط لهرمية الفئات اللازمة لمعالجة المعلومات في العيادة:



الإشارة ◆——— في المخطّط تمثّل احتواءً.

/ يتبع في صفحة 45 /

أ. بالنسبة لكلّ فئة في المخطّط، عرِّف صفاتها وعملياتها.
عليك فقط تعريف العمليات اللازمة للإيفاء بالمتطلّبات التي وُصفت في مقدّمة السؤال (التي في صفحة 44) وفي المخطّط.

افترض أنّه معطاة عمليات بنائية تتلقّى بارامترًا لكلّ صفة، وعمليات مُعيدة (عمليات get)، ولا حاجة لكتابتها. عمليات محدِّدة (عمليات set) غير معطاة.
بالنسبة لكلّ صفة – اكتب تعريفها بلغة Java ، واكتب توثيقها.
بالنسبة لكلّ عملية – اكتب عنوانها بلغة Java ، واكتب توثيقًا يشمل ماذا تتلقّى وماذا تُعيد. لا حاجة لتطبيق العملية.

ب. افترض أنّ المنظومة طُوِّرت حسب التخطيط الذي عرضتَه في البند "أ"، وأنّ جميع العمليات المعطاة وتلك التي عرّفتَها قد تمّ تطبيقها.

في الفئة **Clinic** نضيف العملية:

public void addAppointment (Animal p , String t , Vet v)

التي تتلقّى حيوانًا بيتيًا p ، ونصًّا t لأكواد العلاجات التي حصل عليها الحيوان البيتي في الزيارة الحالية، والطبيب البيطري v الذي عالجه. تضيف العملية الزيارة إلى الحيوان البيتي.
طبِّق العملية بشكل كامل.

افترض أنّ عدد الزيارات السابقة للحيوان في العيادة هو أصغر من 50.
إذا استعملتَ عمليات أخرى بالإضافة إلى العمليات المعطاة والعمليات التي عرّفتَها في البند "أ"، عليك أن تطبّقها بشكل كامل وأن تذكر بالنسبة لكلّ عملية في أيّة فئة يجب تطبيقها.

/ يتبع في صفحة 46 /

برمجة موجّهة كائنات

إذا تعلّمتَ هذا المسار وتكتب بلغة C#، أجب عن اثنين من الأسئلة ٢١–٢٤.
(لكلّ سؤال – ٢٥ درجة)

٢١. 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** عملية بوليانية يمكن دهسها، 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 لمحتوى النصّ st التابع للكائن الحالي – تُعيد العملية true ، خلاف ذلك – تُعيد false .

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

/ يتبع في صفحة 47 /

جـ. أمامك قطعة من عملية رئيسية:

```
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 للعملية 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 عملية رئيسية:

```
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 للعملية 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 عملية خارجية باسم LongString تتلقّى مصفوفة لكائنات من نمط Object .
تُعيد العملية 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 48/

٢٢. أمامك مشروع فيه الفئات **B** وَ **D** وَ **A** وَ **OopTest**:

```
public class B
{
    private static int numB = 0;
    private int m1;
    private int m2;

    public B(int m1, int m2)
    {
        this.m1 = m1;
        this.m2 = m2;
        numB++;
        Console.WriteLine("B(" + m1 + ", " + m2 + ") ,  #" + numB);
    }
}

public class D : B
{
    private static int numD = 0;
    private double d;

    public D(double d, int x) : base (x , x)
    {
        this.d = d;
        numD++;
        Console.WriteLine("D(" + d + ", " + x + ") ,  #" + numD);
    }

    public D(double d, int x, int y) : base(x , y)
    {
        this.d = d;
        numD++;
        Console.WriteLine("D(" + d +", "+ x + " , " + y + ") ,  #" + numD);
    }
}
```

/ يتبع في صفحة 49 /

```
public class A
{
    private static int  numA = 0;
    private A  a;
    private B  b;

    public A(A  a , B  b)
    {
        this.a = a;
        this.b = b;
        numA++;
        Console.WriteLine("A Constructor ,  #" + numA);
    }
}

public class OopTest
{
    public static void Main(string[ ] args)
    {
        B   w1 = new B(2 , 3);
        B   w2 = new D(1.5 , 6);
        B   w3 = new D(2.3 , 8 , 9);
        A   w4 = new A(null , w1);
        A   w5 = new A(w4 , w3);
    }
}
```

اكتب متابعة للعملية Main في الفئة **OopTest** ، واكتب الـمُخرَج.

يجب أن تكتب في المتابعة قيَم المتغيّرات، وبالنسبة لكلّ كائن – قيَم صفاته.

/ يتبع في صفحة 50 /

٢٣. طوّرت شركة لتوزيع برامج الحاسوب برنامج حاسوب يتناول متواليات الأعداد الصحيحة.
تمّ تطوير المنظومة على مراحل.
بالنسبة لكلّ متوالية أعداد، يتطرّقون إلى:
(١) الحدّ الأوّل في المتوالية الذي رقمه التسلسلي هو 1.
(٢) الحدّ الذي رقمه التسلسلي في المتوالية هو n .
(٣) طباعة n الحدود الأولى في المتوالية.

طُوِّرت **في المرحلة الأولى** فئتان:
متوالية حسابية (**ASeq**) – متوالية الفرق فيها بين كلّ حدّ والحدّ الذي قبله هو قيمة ثابتة.
متوالية هندسية (**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 51 /

```
    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 52 /

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:

(١) الحدّ الأوّل في المتوالية الذي رقمه التسلسلي هو 1.

(٢) الحدّ الذي رقمه التسلسلي في المتوالية هو n .

(٣) 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 عملية 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 53 /

تَقَرَّرَ **في المرحلة الرابعة** من التطوير، تطوير عملية ستاتيّة Check تتلقّى قيمة صحيحة n ، وكائنين لمتواليات: أحدهما من نمط **ASeq** والآخر من نمط **GSeq**. تحسب العملية 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 العملية الستاتيّة Check حسب المتطلّبات التي عُرِّفت في المرحلة الرابعة من التطوير.

/ يتبع في صفحة 54 /

٢٤. تركّز عيادة بيطرية للحيوانات البيتية معلومات تتعلّق بالأطبّاء البيطريين الذين يعملون في العيادة وبالحيوانات البيتية التي تتلقّى العلاج فيها. عدد الأطبّاء البيطريين الذين يعملون في العيادة هو 10 على الأكثر، وعدد الحيوانات البيتية التي تُعالج في العيادة هو 500 على الأكثر.

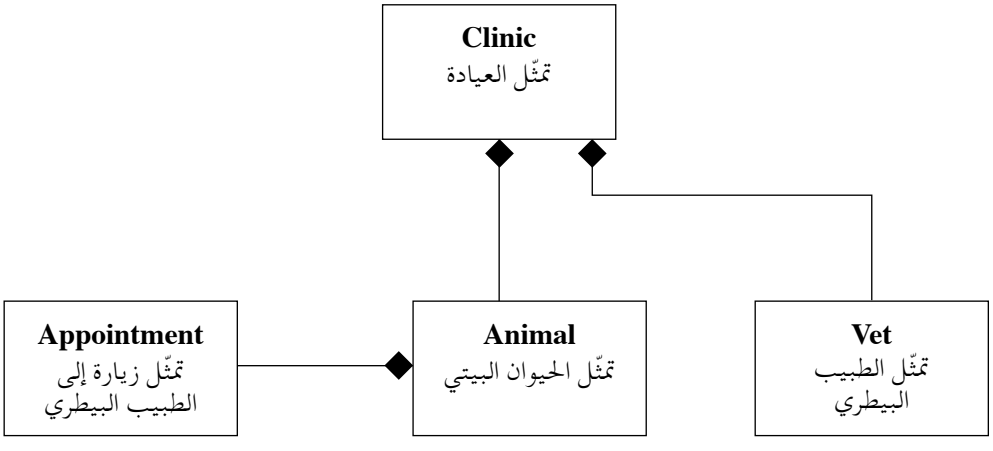
النقاط الموجِّهة في معالجة المعلومات هي:

- يُحفَظ لكلّ طبيب بيطري: رقم الهوية والاسم والأقدمية في العمل بالسنوات.
- يُحفَظ لكلّ حيوان بيتي: رقم الرخصة والاسم والنوع (مثلاً: كلب أو قطّة أو أرنب)، والعمر، وتفصيل حتّى 50 الزيارات الأخيرة في العيادة وعدد الزيارات المحفوظة.
- يُحفَظ لكلّ زيارة إلى العيادة: رقم هوية الطبيب البيطري الذي عالج الحيوان البيتي في هذه الزيارة، ونصّ من رموز يحوي أكواد العلاجات التي حصل عليها الحيوان البيتي في تلك الزيارة. كود العلاج هو حرف كبير من بين جميع أحرف الأبجدية ABC.

جزء من العمليات التي تستطيع المنظومة تنفيذها هي:

- إعداد تقرير عن جميع الحيوانات البيتية من نوع معيّن (مثلاً جميع الأرانب). يشمل التقرير بالنسبة لكلّ حيوان رقم رخصته واسمه وعمره.
- في بداية كلّ سنة: تعديل عمر كلّ حيوان بيتي، أي تكبير العمر بـِ 1، وتعديل أقدمية كلّ طبيب بيطري، أي تكبير الأقدمية بـِ 1.
- إعادة اسم الطبيب البيطري حسب رقم هويته.

أمامك مخطّط لهرمية الفئات اللازمة لمعالجة المعلومات في العيادة:



الإشارة ◆——— في المخطّط تمثّل احتواءً.

/ يتبع في صفحة 55 /

أ. بالنسبة لكلّ فئة في المخطّط، عرِّف صفاتها وعملياتها.
عليك فقط تعريف العمليات اللازمة للإيفاء بالمتطلّبات التي وُصفت في مقدّمة السؤال (التي في صفحة 54) وفي المخطّط.

افترض أنّه معطاة عمليات بنائية تتلقّى بارامترًا لكلّ صفة، وعمليات مُعيدة (عمليات Get)، ولا حاجة لكتابتها. عمليات محدِّدة (عمليات Set) غير معطاة.
بالنسبة لكلّ صفة – اكتب تعريفها بلغة C# ، واكتب توثيقها.
بالنسبة لكلّ عملية – اكتب عنوانها بلغة C# ، واكتب توثيقًا يشمل ماذا تتلقّى وماذا تُعيد. لا حاجة لتطبيق العملية.

ب. افترض أنّ المنظومة طُوِّرت حسب التخطيط الذي عرضتَه في البند "أ"، وأنّ جميع العمليات المعطاة وتلك التي عرّفتَها قد تمّ تطبيقها.

في الفئة **Clinic** نضيف العملية:

public void AddAppointment (Animal p , string t , Vet v)

التي تتلقّى حيوانًا بيتيًا p ، ونصًّا t لأكواد العلاجات التي حصل عليها الحيوان البيتي في الزيارة الحالية، والطبيب البيطري v الذي عالجه. تضيف العملية الزيارة إلى الحيوان البيتي.
طبِّق العملية بشكل كامل.
افترض أنّ عدد الزيارات السابقة للحيوان إلى العيادة أصغر من 50 .
إذا استعملتَ عمليات أخرى بالإضافة إلى العمليات المعطاة والعمليات التي عرّفتَها في البند "أ"، عليك أن تطبّقها بشكل كامل وأن تذكر بالنسبة لكلّ عملية في أيّة فئة يجب تطبيقها.

ב ה צ ל ח ה!
نتمنّى لك النجاح!

זכות היוצרים שמורה למדינת ישראל.
אין להעתיק או לפרסם אלא ברשות משרד החינוך.
حقوق الطبع محفوظة لدولة إسرائيل.
النسخ أو النشر ممنوعان إلا بإذن من وزارة المعارف.