

Ciencias de la Computación
Según el programa de la reforma de la enseñanza significativa

מדעי המחשב
על פי תכנית הרפורמה ללמידה משמעותית

Instrucciones para el examen

- א. Duración del examen: tres horas.
ב. Estructura del cuestionario y clave de la evaluación:
Este cuestionario consta de tres partes.
Primera parte - Esta parte consta de tres preguntas, respóndelas según las instrucciones de esta parte.
 $(1 \times 10) + (1 \times 15) = 25$ puntos
Segunda parte - Esta parte consta de tres preguntas de las que debes responder a dos.
 $(2 \times 25) = 50$ puntos
Tercera parte - Esta parte consta de preguntas que se pueden responder según cuatro orientaciones diferentes.
Responde solamente a una de la orientación en la que estudiaste.
 $(1 \times 25) = 25$ puntos
Total — 100 puntos

- ג. Material auxiliar permitido:
Cualquier material auxiliar, salvo calculadoras programables.
ד. Instrucciones especiales:
(1) En la primera parte, debes escribir todos los programas que te son requeridos en un solo lenguaje: C# o Java.
(2) **Indica en la cobertura externa** del cuadernillo cuál es el **lenguaje** de programación con el que respondes: C# o Java.
(3) **Indica en la cobertura externa** del cuadernillo el nombre de la **orientación** según la cual estudiaste, una de cuatro orientaciones: Sistemas de computación y Assembler, Introducción a la Investigación Operativa, Modelos computacionales, Programación orientada a objetos.
Nota: No se te descontarán puntos en el caso que escribas minúsculas o mayúsculas unas en lugar de otras en el momento de escribir programas.

Escribe **solamente en el cuadernillo de examen**, y utiliza como páginas de **borrador** para anotaciones varias (cálculos, títulos, etc.) otras páginas del cuadernillo. Indica con la palabra "טיטה" cada una de las hojas que utilices como borrador. ¡Toda anotación realizada en hojas que no pertenezcan al cuadernillo del examen puede causar la anulación del mismo!

¡Buena Suerte!

הוראות לנבחן

- א. משך הבחינה: שלוש שעות.
ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שלושה פרקים.
פרק ראשון — בפרק זה שלוש שאלות, ענה על פי ההוראות בפרק.
 $(10 \times 1) + (15 \times 1) = 25$ נקודות
פרק שני — בפרק זה שלוש שאלות, ומהן עליך לענות על שתיים.
 $(25 \times 2) = 50$ נקודות
פרק שלישי — בפרק זה שאלות בארבעה מסלולים שונים. ענה על שאלות **ב** במסלול שלמדת, לפי ההוראות בקבוצת השאלות במסלול זה.
 $(25 \times 2) = 50$ נקודות
סה"כ — 100 נקודות
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 המסלול 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.

כתוב במחברת הבחינה בלבד, בעמודים נפרדים, כל מה שברצונך לכתוב **בטיטה** (ראשי פרקים, חישובים וכדומה). רשום "טיטה" בראש כל עמוד טיטה. רישום טיטות כלשהן על דפים מחוץ למחברת הבחינה עלול לגרום לפסילת הבחינה!

בהצלחה!

Preguntas

Este cuestionario consta de tres partes:

Debes responder a preguntas de las tres partes, según las instrucciones de cada parte.

Primera parte (25 puntos)

Nota: En cada pregunta en la que haga falta realizar una entrada no es necesario revisar su corrección.

Para quienes resuelven en Java : en cada pregunta en la que haga falta realizar una entrada, debes suponer que en el programa se halla escrita la instrucción:

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

Responde a la pregunta 1 obligatoria (10 puntos).

1. Se da una clase Alumno – **Student** que posee dos atributos:
nombre del alumno — name, de tipo cadena de caracteres, y un vector unidimensional arrTest de magnitud 3 de tipo entero. En cada celda del vector se almacena la nota de examen del alumno.
Escribe en Java o en C# un método en la clase **Student** que devuelva la nota promedio del alumno en los tres exámenes.

Nota: no es necesario analizar la corrección de los datos del vector.

Responde a una de las preguntas 2 - 3 (15 puntos)

2. Se da la clase Actor — **Actor** que posee tres atributos: número de documento de identidad — `id` de tipo cadena de caracteres, sexo del actor — `gender`, de tipo cadena de caracteres ("F" designa femenino, "M" designa masculino), número de películas en las que el actor intervino — `numFilms` de tipo entero.

A continuación se halla la interface de la clase **Actor** escrita en Java y en C#.

Título del método en Java	Descripción del método
<code>public Actor(String id, String gender, int numFilms)</code>	Un método que construye un actor cuyo número de identidad es <code>id</code> , de sexo <code>gender</code> , y número de películas en las que intervino es <code>numFilms</code> .
<code>public void addFilm()</code>	Un método que adiciona 1 al número de películas en las que intervino el actor.
<code>public int compare(Actor other)</code>	Un método que devuelve: 1 — si el número de películas en las que intervino el actor actual es mayor que el número de películas en las que intervino el actor <code>other</code> . 2 — si el número de películas en las que intervino el actor actual es menor que el número de películas en las que intervino el actor <code>other</code> . 3 — si el número de películas en las que intervino el actor actual es idéntico al número de películas en las que intervino el actor <code>other</code> .

(Presta atención : La continuación de la pregunta se halla en la página siguiente.)

Título del método en C#	Descripción del método
public Actor(string id, string gender, int numFilms)	Un método que construye un actor cuyo número de identidad es id, de sexo gender, y número de películas en las que intervino es numFilms.
public void AddFilm()	Un método que adiciona 1 al número de películas en las que intervino el actor.
public int Compare(Actor other)	El método devuelve: 1 — si el número de películas en las que intervino el actor actual es mayor que el número de películas en las que intervino el actor other. 2 — si el número de películas en las que intervino el actor actual es menor que el número de películas en las que intervino el actor other. 3 — si el número de películas en las que intervino el actor actual es idéntico al número de películas en las que intervino el actor other.

Debes suponer que para cada atributo se definieron en Java métodos get y set y en C# métodos Get y Set.

⌘. Implementa el método constructor en la clase **Actor** :

En Java el método constructor:

```
public Actor(String id, String gender, int numFilms)
```

En C#, el método constructor:

```
public Actor(string id, string gender, int numFilms)
```

Ⓜ. Implementa en Java el método compare o en C#, el método Compare.

Ⓛ. En el método principal de la clase Program se definió un vector unidimensional actArr de magnitud 37 de tipo **Actor**.

Escribe en Java o en C# un método en la clase Program que reciba un vector de tipo **Actor** y un número num entero y mayor que 0.

El método imprimirá el número de actores del vector que intervinieron en más de num películas.

/continúa en la página 5

3. Se da la clase programa de televisión **TvProgram** que tiene tres atributos: código de programa - `code` de tipo entero; día de la semana en que se emite el programa `day`, de tipo entero que recibe valores de 1 a 7 (incluido), si se trata de un programa de deportes `isSport`, de tipo booleano, cuyo valor es `true` si se trata de un programa de deportes, de lo contrario – `false`.

Debes suponer que para cada atributo fueron definidos en Java métodos `get` y `set`, y en C# métodos `Get` y `Set` .

- א. Escribe en Java o C# un método constructor en la clase

TvProgram que reciba valores para cada atributo.

- ב. Se da la clase programa semanal **TvWeek** que tiene dos atributos: un vector unidimensional `arrProg` de magnitud 100 de tipo **TvProgram**, el número actual de programas semanales – `current` de tipo entero cuyo valor es menor que la magnitud del vector.

Debes suponer que para cada atributo fueron definidos en Java métodos `get` y `set`, y en C# métodos `Get` y `Set` .

- (1) Escribe en Java o C# el título de la clase **TvWeek** y sus atributos.

- (2) Escribe en Java o C# en la clase **TvWeek** un método que reciba el programa **TvProgram**, lo agregue al programa semanal y actualice el número de programas actual de la semana.

- (3) Escribe en Java o C# en la clase **TvWeek** un método que devuelva el número de programas de deportes que hay en la semana en la televisión.

Segunda parte (50 puntos)

Presta atención: en toda pregunta en la que se requiera implementación, puedes utilizar métodos de las clases cola, pila, árbol binario y eslabón, [תור, מחסנית, עץ בינרי וחוליה] sin implementarlos. Si utilizas métodos adicionales, deberás implementarlos.

Responde a dos de las preguntas 4– 6 (cada pregunta – 25 puntos).

4. A continuación la clase anillo **Ring** que posee dos atributos:

La magnitud del anillo del tipo cadena de caracteres ("S" anillo pequeño, "L" anillo grande);

Un número entero que representa el color del anillo.

Java

```
public class Ring
{
    private String size; //tamaño del anillo
    private int color; //color del anillo
    public Ring()
    {
        this.size = "L";
        this.color = 0;
    }
    public Ring(String str, int c)
    {
        this.size = str;
        this.color = c;
    }
    public String getSize()
    {
        return this.size;
    }
    public int getColor()
    {
        return this.color;
    }
}
```

C#

```
public class Ring
{
    private string size; //tamaño del anillo
    private int color; //color del anillo
    public Ring()
    {
        this.size = "L";
        this.color = 0;
    }
    public Ring(string str, int c)
    {
        this.size = str;
        this.color = c;
    }
    public string GetSize()
    {
        return this.size;
    }
    public int GetColor()
    {
        return this.color;
    }
}
```

A continuación se halla la interface de la clase **Pole**.

Java	C#	Método
public Pole()	public Pole()	Un método que construye un palo vacío. La complejidad del tiempo de ejecución del método es $O(1)$
public void add(Ring r)	public void Add(Ring r)	Un método que introduce un anillo r en la cabeza del palo. La complejidad del tiempo de ejecución del método es $O(1)$
public Ring remove()	public Ring Remove()	Un método que devuelve el anillo de la cabeza del palo. Además retira el anillo de la cabeza del palo. La complejidad del tiempo de ejecución del método es $O(1)$
public boolean isEmpty()	public bool IsEmpty()	Si el palo está vacío el método devuelve true, En caso contrario, devuelve false. La complejidad del tiempo de ejecución del método es $O(1)$
public void sort()	public void Sort()	Un método que ordena los anillos que están sobre el palo del siguiente modo: Los anillos grandes son "dispuestos" en la parte inferior del palo y los anillos pequeños sobre ellos.

- א. Implementa el método `sort()` en Java o `Sort()` en C#, de la clase **Pole**. Puedes utilizar los métodos adicionales de la clase **Pole** sin implementarlos.
En tu respuesta utiliza sólo métodos en las clases **Pole** y **Ring**.
- ב. ¿Cuál es la complejidad del tiempo de ejecución del método que implementaste en el apartado א? Justifica tu respuesta.

5.

Presta atención: Esta pregunta tiene dos versiones:
una versión en Java en las páginas 8-9, y
una versión en C# en las páginas 10-11.
Trabaja según el lenguaje que estudiaste.

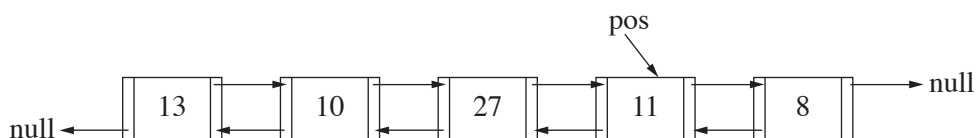
Para los que resuelven en Java

Definiremos como **lista bidireccional** a una colección ordenada de eslabones de tipo `BinNode<Integer>` enlazados del siguiente modo:

Para cada par de eslabones $p1, p2$ en la lista, si se cumple $p1.getRight() == p2$, se cumple también $p2.getLeft() == p1$. En la **lista bidireccional** hay por lo menos dos eslabones.

Es decir: cada eslabón en la lista, fuera del eslabón del extremo derecho de la lista y del eslabón del extremo izquierdo de la lista, apunta al eslabón que lo precede y al eslabón que lo sucede.

A continuación hay un ejemplo de **lista bidireccional** y una variable `pos` de tipo `BinNode<Integer>` que apunta a un eslabón cualquiera en la lista bidireccional.



El método `firstLeft` recibe un indicador `pos` distinto de `null` de tipo `BinNode<Integer>` que apunta a un eslabón cualquiera de la lista bidireccional y devuelve el eslabón extremo izquierdo de la lista.

El método `firstRight` recibe un indicador `pos` distinto de `null` de tipo `BinNode<Integer>` que apunta a un eslabón cualquiera de la lista bidireccional y devuelve el eslabón extremo derecho de la lista.

✖. A continuación el esqueleto del método `firstLeft`.

Cópialo en tu cuadernillo y complétalo, de modo que el método realice lo que se pide.

```
public static BinNode<Integer> firstLeft(BinNode<Integer> pos)
{
    while ( _____ )
        pos = _____ ;
    return _____ ;
}
```

/continúa en la página 9/

- ב. A continuación se halla el método `what(BinNode<Integer> pos)` que recibe un indicador que apunta a un eslabón cualquiera en la **lista bidireccional** y devuelve un valor booleano. La **lista bidireccional** contiene por lo menos 3 eslabones.

- (1) Realiza el seguimiento de la ejecución del método para la variable `pos` y la lista del ejemplo del comienzo de la pregunta. En el seguimiento exhibe la lista bidireccional y los valores de las variables `sum`, `right`, `left`, `pos`.

```
public static boolean what(BinNode<Integer> pos)
{
    BinNode<Integer> left = firstLeft(pos);
    BinNode<Integer> right = firstRight(pos);

    int sum = left.getValue() + right.getValue();
    left = left.getRight();
    right = right.getLeft();

    while ((left != right) && (left.getRight() != right) &&
           (left.getValue() + right.getValue() == sum))
    {
        left = left.getRight();
        right = right.getLeft();
    }
    if (left == right)
        return right.getValue() == sum;

    if (left.getRight() == right)
        return left.getValue() + right.getValue() == sum;
    return false;
}
```

- (2) Determina si es o no posible reemplazar las últimas 3 líneas del método — las líneas encerradas en el recuadro — por la instrucción:

```
return left.getValue() + right.getValue() == sum;
```

Fundamenta tu determinación.

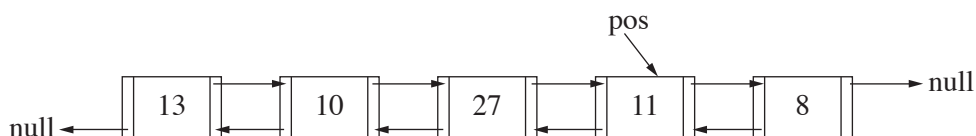
Para los que resuelven en C#

Definiremos como **lista bidireccional** a una colección ordenada de eslabones de tipo `BinNode<int>` enlazados del siguiente modo:

Para cada par de eslabones $p1, p2$ en la lista, si se cumple $p1.GetRight() = p2$, se cumple también $p2.GetLeft() = p1$. En la **lista bidireccional** hay por lo menos dos eslabones.

Es decir: cada eslabón en la lista, fuera del eslabón del extremo derecho de la lista y del eslabón del extremo izquierdo de la lista, apunta al eslabón que lo precede y al eslabón que lo sucede.

A continuación hay un ejemplo de **lista bidireccional** y una variable `pos` de tipo `BinNode<int>` que apunta a un eslabón cualquiera en la lista bidireccional.



El método `FirstLeft` recibe un indicador `pos` distinto de `null` de tipo `BinNode<int>` que apunta a un eslabón cualquiera de la lista bidireccional que devuelve el eslabón extremo izquierdo de la lista. El método `FirstRight` recibe un indicador `pos` distinto de `null` de tipo `BinNode<int>` que apunta a un eslabón cualquiera de la lista bidireccional que devuelve el eslabón extremo derecho de la lista.

✖. A continuación el esqueleto del método `FirstLeft`.

Cópialo en tu cuadernillo y complétalo, de modo que el método realice lo que se pide.

```
public static BinNode<int> FirstLeft(BinNode<int> pos)
{
    while ( _____ )
        pos = _____ ;
    return _____ ;
}
```

- ב. A continuación se halla el método `What(BinNode<int> pos)` que recibe un indicador que apunta a un eslabón cualquiera en la **lista bidireccional** y devuelve un valor booleano. La **lista bidireccional** contiene por lo menos 3 eslabones.

- (1) Realiza el seguimiento de la ejecución del método para la variable `pos` y la lista del ejemplo del comienzo de la pregunta. En el seguimiento exhibe la lista bidireccional y los valores de las variables `sum`, `right`, `left`, `pos`.

```
public static bool What(BinNode<int> pos)
{
    BinNode<int> left = FirstLeft(pos);
    BinNode<int> right = FirstRight(pos);

    int sum = left.GetValue() + right.GetValue();
    left = left.GetRight();
    right = right.GetLeft();

    while ((left != right) && (left.GetRight() != right) &&
           (left.GetValue() + right.GetValue() == sum))
    {
        left = left.GetRight();
        right = right.GetLeft();
    }
    if (left == right)
        return right.GetValue() == sum;

    if (left.GetRight() == right)
        return left.GetValue() + right.GetValue() == sum;
    return false;
}
```

- (2) Determina si es o no posible reemplazar las últimas 3 líneas del método — las líneas encerradas en el recuadro — por la instrucción:

```
return left.GetValue() + right.GetValue() == sum;
```

Fundamenta tu determinación.

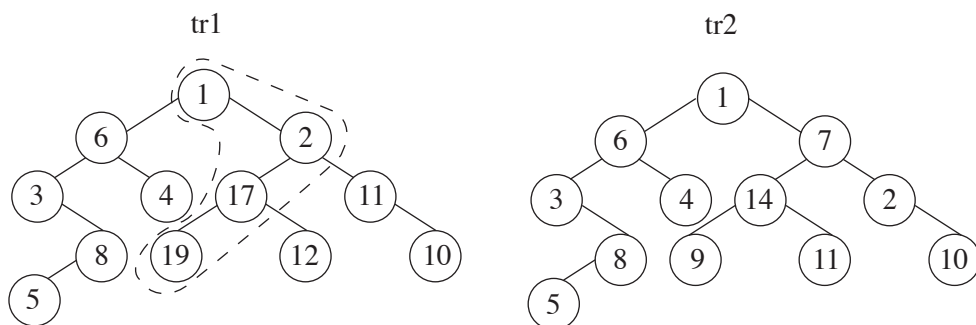
6. Un árbol de números es un árbol binario no vacío de tipo entero, tal que los valores en sus nodos [צומת] son números enteros mayores que 0 y diferentes entre sí.

Sobre el **árbol de números** se define un método "recorrido ascendente", que devuelve true, si en el árbol hay un recorrido que comienza en la raíz del árbol y finaliza en una de sus hojas, y los valores de los nodos están ordenados en orden ascendente desde la raíz hasta la hoja. Si no hay un recorrido tal, el método devuelve false.

Por ejemplo:

Para el **árbol de números** tr1 el método "recorrido ascendente" devuelve true, el método rodeado con una línea punteada.

Para el **árbol de números** tr2 el método "recorrido ascendente" devuelve false.



Implementa en Java o en C# el método "recorrido ascendente" para **árbol de números** tr.

Título del método en Java: `public static boolean upPath(BinNode<Integer> tr)`

Título del método en C#: `public static bool UpPath(BinNode<int> tr)`

Tercera parte (25 puntos)

Esta parte consta de preguntas según cuatro orientaciones:

Sistemas de computación y Assembler, págs. 13-16

Introducción a la investigación operativa, págs. 17-20

Modelos computacionales, págs. 21 .

Programación orientada a objetos en Java, págs. 22-26.

Programación orientada a objetos en C#, págs. 27-31.

Responde solamente a las preguntas de la orientación que estudiaste.

Sistemas de computación y Assembler

Si estudiaste en esta orientación, responde a una de las preguntas 7-8 (25 puntos).

7. Esta pregunta consta de dos apartados א-ב, los cuales no tienen relación entre sí. Responde a ambos.

- א. Escribe en Assembler un segmento de programa para encriptar un carácter.

El carácter se definió del siguiente modo en el segmento de datos:

TAV DB ?

El encriptado se realiza del siguiente modo:

Se cuenta el número de bits encendidos en el carácter.

Si el número es par se realiza un corrimiento circular hacia la derecha de los bits del carácter, un número de veces equivalente al número de bits encendidos.

Si el número es impar se realiza un corrimiento circular hacia la izquierda de los bits del carácter, un número de veces equivalente al número de bits encendidos.

- ב. (Sin relación con el apartado א)

Para cada una de las proposiciones 1-6 que siguen, determina si es verdadera o falsa. Si la proposición es falsa, explica por qué

(1) A continuación hay un segmento de programa en Assembler.

MOV AX , 8

MOV BX , 2

DIV BX

Al finalizar el segmento de programa el acumulador AX contendrá necesariamente 4.

(Presta atención : La continuación de este apartado se halla en la página siguiente.)

/continúa en la página 14/

- (2) A continuación hay un segmento de programa en Assembler.

```
MOV    AL , 56
ADD    AL , 200
JZ     STOP
INC    AL
```

STOP:

Al finalizar el segmento de programa el acumulador AX contendrá 1.

- (3) A continuación hay un segmento de programa en Assembler.

```
ARRAY DW 1, 2, 3, 4
MOV    BX , ARRAY
ADD    BX , 2
MOV    AX , [BX]
```

Al finalizar el segmento de programa el acumulador AX contendrá 3.

- (4) A continuación hay un segmento de programa en Assembler.

```
MOV    CX , 3
MOV    AX , 1
```

DO:

```
SHL    AX , 1
LOOP   DO
```

Al finalizar el segmento de programa el acumulador AX contendrá 8.

- (5) A continuación hay un segmento de programa en Assembler.

```
MOV AX , 11000001B
MOV BX , 01000001B
```

El número que se halla en AX es necesariamente mayor que el número que se halla en BX.

- (6) A continuación se halla la instrucción:

```
OR AL , 3
```

Después de la ejecución de la instrucción el valor de AL es siempre impar.

8. Esta pregunta consta de dos apartados א-ב, los cuales no tienen relación entre sí. Responde a ambos.

א. En un segmento de datos se definieron los datos del siguiente modo:

VEC1 DB 45H , 26H , 32H , 82H

VEC2 DB 4 DUP(0)

A continuación se halla un segmento de programa en Assembler.

```
START:  MOV    CL , 4
        XOR    CH , CH
        XOR    SI , SI
NEXT:   PUSH   CX
        MOV    AL , VEC1[SI]
        MOV    AH , AL
        AND    AL , 0FH
        MOV    CL , 4
        SHR    AH , CL
        MUL    AH      ;(*)
        MOV    VEC2[SI] , AL
        INC    SI
        POP    CX
        LOOP   NEXT
SOF:    NOP
```

(1) Realiza un seguimiento de la ejecución de este segmento de programa con la ayuda de una tabla de seguimiento, y traza VEC2 después de la ejecución.

En la tabla de seguimiento hay que incluir una columna para cada uno de los acumuladores : AL, AH, CL, CH, SI.

Dibuja la pila en cada etapa.

(2) ¿Qué ejecuta el segmento de programa?

(3) En el segmento de programa, en lugar de la línea indicada con (*) se escribió la instrucción: IMUL AH.

¿Se alterará la ejecución del programa? Fundamenta tu respuesta.

(Presta atención : La continuación de este apartado se halla en la página siguiente.)

/continúa en la página 16/

- ב. En el acumulador AL está almacenado el número 2 y en el acumulador BL está almacenado el número 5.

Se debe almacenar en el acumulador DX la suma de los números desde AL hasta BL (incluido).

A continuación se hallan tres segmentos i- iii en Assembler. Realiza un seguimiento de la ejecución de cada uno de los segmentos de programa con la ayuda de una tabla de seguimiento, y determina para cada uno si realiza, o no, lo que se pretende.

```
i          MOV    DX , 0
           MOV    AH , 0
           MOV    CL , AL
           SUB    CL , BL
AGAIN:     ADD    DX , AX
           INC    AL
           LOOPE  AGAIN
           NOP
```

```
ii         XOR    DX , DX
           MOV    BH , 0
AGAIN:     ADD    DX , BX
           DEC    BL
           CMP    BL , AL
           JGE    AGAIN
           NOP
```

```
iii        XOR    DX , DX
           XOR    AX , AX
           MOV    BX , 0
AGAIN:     ADD    DX , AX
           ADD    AX , 1
           CMP    AX , BX
           JL     AGAIN
           NOP
```

Introducción a la Investigación operativa [חקר ביצועים]

Si estudiaste en esta orientación, responde a dos de las preguntas 9-10 (cada pregunta – 25 puntos).

9. Se da un problema de programación lineal:

$$\max \{z = (2 + 2k)x_1 + 2x_2\}$$

Sometido a las restricciones siguientes:

$$(1) \quad 2x_1 + x_2 \leq 10$$

$$(2) \quad x_1 + x_2 \leq 6$$

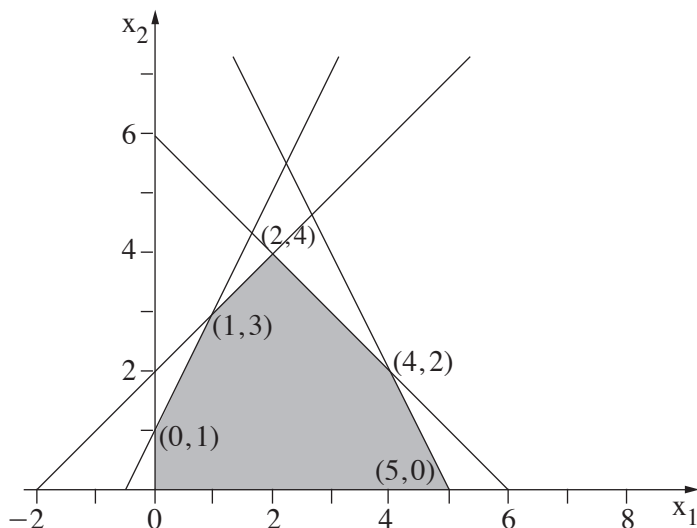
$$(3) \quad -x_1 + x_2 \leq 2$$

$$(4) \quad -2x_1 + x_2 \leq 1$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

A continuación se halla el dibujo del intervalo de las soluciones posibles del problema dado.



Cada uno de los apartados א-ז se refiere al problema de programación lineal dado.

Los apartados son independientes entre sí. Responde a todos los apartados

(Presta atención : Estos apartados se hallan en la página siguiente.)

- א. A continuación se hallan dos subapartados (1)-(2) **independientes entre sí**. En cada uno de ellos se da un valor determinado para el parámetro k .

(1) $k = -1$

(2) $k = -3$

Para cada uno de los subapartados (1)-(2) se dan las cuatro proposiciones i-iv que siguen:

- i La solución óptima es única.
- ii Existe un número infinito de soluciones óptimas.
- iii La solución óptima no está acotada.
- iv No existe solución óptima.

En cada subapartado sólo una de las proposiciones i-iv es verdadera.

Para cada uno de los subapartados (1)-(2) determina cuál es la proposición verdadera, cópiala en tu cuadernillo y fundamenta tu determinación.

- Si determinaste que i es verdadera, debes encontrar la solución óptima única de este subapartado, y el valor de la función objetivo en la solución que encuentres.
 - Si determinaste que ii es verdadera, debes encontrar la solución óptima general del problema, y el valor de la función objetivo en el intervalo de soluciones óptimas.
- ב. ¿Para qué valores de k será (2, 4) la solución óptima para el problema de programación lineal dado al comienzo de la pregunta? Justifica tu respuesta.
- ג. Se anula la restricción $x_1 \geq 0$ del problema dado al comienzo de la pregunta, o sea $-\infty \leq x_1 \leq \infty$.
¿Existe un valor de k para el cual la solución óptima sea no acotada? Justifica tu respuesta.
- ד. En lugar de la restricción $x_2 \geq 0$ del problema dado al comienzo de la pregunta se escribe: $x_2 \geq 4$.
¿Tiene el problema solución óptima después del cambio de la restricción? En caso afirmativo, escribe cuál es la solución.
En caso contrario, explica por qué no hay solución óptima.

10. En esta pregunta hay tres apartados α - γ independientes uno del otro.

Responde a los tres.

α . $G = (V, E)$ es un grafo **orientado** representado por la siguiente matriz de adyacencias:

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

- (1) Dibuja el grafo G representado por la matriz de adyacencias.
- (2) Halla los componentes de conexidad fuerte (c.c.f. Strong Connected Components, c.c.f. – רק"חים) que hay en el grafo dado.
Para cada c.c.f. que halles, escribe el conjunto de sus vértices.
- (3) Halla en el grafo dado un circuito de longitud mínima y cuyo número de arcos sea par, y dibújalo en tu cuadernillo.
- (4) ¿Cuál es el número mínimo de arcos que hay que agregar al grafo dado para que el grafo contenga un solo c.c.f.?
¿Cuál es el arco o los arcos que hay que agregar?

β . En la tabla que está a continuación se da una parte de una solución básica posible de un problema de transporte: $x_{11} = 10$

Orígenes	Destinos				Oferta
	1	2	3	4	
1	20 10	20	30	21	20
2	22	17	29	30	30
3	10	24	26	38	10
Demanda	10	20	20	10	

- (1) Copia la tabla en tu cuadernillo y completa en ella la solución básica posible según el método de la esquina noroeste.
- (2) Se varía el precio de la celda (3, 1) de 10 a 20. ¿Hará eso cambiar la solución básica posible que hallaste en el subapartado (1)?
Justifica tu respuesta.

(Presta atención : La continuación del apartado γ se halla en la página siguiente.)

/continúa en la página 20/

- ג. En la tabla que está a continuación se da una solución básica posible de un problema de transporte y se da el valor de u_3 .

Orígenes	Destinos			Oferta	u_i
	1	2	3		
1	10	8	3	10	
2	12	15	9	3	
3	2	7	1	90	0
Demanda	20	40	43		
v_j					

- (1) Copia la tabla en tu cuadernillo y completa en ella los valores de u_1, u_2, v_1, v_2, v_3
- (2) Explica por qué la solución dada no es una solución óptima.

Modelos computacionales

Si estudiaste en esta orientación, responde a una de las preguntas 11-12 (25 puntos).

11. Se dan los lenguajes regulares L_1 y L_2 .

$$L_1 = \{a^{2n} \mid n \geq 0\} \text{ sobre el alfabeto } \{a\}$$

$$L_2 = \{b^{2n+1} \mid n \geq 0\} \text{ sobre el alfabeto } \{b\}.$$

Se da el lenguaje L sobre el alfabeto $\{a, b\}$.

$$L = \{a^n b^k \mid n \geq 0, k \geq 0, \text{ con } n \text{ par y } k \text{ impar o } n \text{ impar y } k \text{ par}\}$$

א. Demuestra por medio de los lenguajes L_1 y L_2 solamente, y por medio de propiedades de clausura [תכונות סגירות] solamente, que el lenguaje L es regular.

ב. Construye un autómata finito no determinístico que admita el lenguaje L .

12. Se da un método escrito en Java y en C#.

El método admite tres números enteros mayores que 0.

Java	C#
<pre>public static int foo(int x , int y , int z) { if ((x % 3) == 0) return x; if ((x % 3) == 1) return y; return z; }</pre>	<pre>public static int Foo(int x , int y , int z) { if ((x % 3) == 0) return x; if ((x % 3) == 1) return y; return z; }</pre>

Escribe una máquina de Turing que implemente el método dado.

La entrada para la máquina son los tres números x, y, z que el método recibe y está escrita sobre una cinta desde el comienzo. Cada número está escrito de modo unario. Entre número y número separa el signo #.

Por ejemplo, si el método recibe 2 para x , 3 para y , y 1 para z , la cinta se verá así:

⊢	1	1	#	1	1	1	#	1	△	△	△	
---	---	---	---	---	---	---	---	---	---	---	---	-------

↑
Cabezal lector escritor

La salida es el valor que el método devuelve y que estará escrito en la cinta en un lugar cualquiera como valor unario entre dos signos \$.

Programación orientada a objetos

Si estudiaste en esta orientación y escribes en Java, responde a dos de las preguntas 13-14. (Cada pregunta – 25 puntos)

13. En esta pregunta hay dos apartados א - א' independientes uno del otro.

Responde a ambos apartados.

א. Se dan cinco clases B, X, Y, Z, Run. Los subapartados (1)-(4) se refieren a dichas clases. Los subapartados son independientes entre sí.

Responde a todos ellos.

- (1) En cada una de las clases B , X , Y , Z se definió el método
`public void foo() {}`
y en la clase Run se definió el método
`public void bar (Object g) {g.foo(); }`
¿Se producirá un error de compilación? En caso afirmativo, explica por qué.
- (2) Debes suponer que las clases X , Y , Z son herederas de la clase B.
En cada una de las clases X , Y , Z se definió el método
`public void foo() {}`
y en la clase Run se definió el método
`public void bar(B g) {g.foo(); }`
¿Se producirá un error de compilación? En caso afirmativo, explica por qué.
- (3) Debes suponer que las clases X , Y , Z son herederas de la clase B.
En cada una de las clases B , X , Y , Z se definió el método
`public void foo() {}`
y en la clase Run se definió el método
`public void bar(B g) {g.foo(); }`
¿Se producirá un error de compilación? En caso afirmativo, explica por qué.
- (4) Debes suponer que las clases X , Y , Z son herederas de la clase B.
En cada una de las clases B , X , Y , Z se definió el método
`public void foo() {}`
y en la clase Run se definió el método
`public void bar(Object g) {g.foo(); }`
¿Se producirá un error de compilación? En caso afirmativo, explica por qué.

ב. (Sin relación con el apartado א.)

A continuación se halla la clase **Singer** que hereda a la clase **Artist** y la interface **IPerform** que la clase **Singer** implementa.

```
interface IPerform
{
    void act();
    int train();
}

public class Singer extends Artist implements IPerform
{
    private int hits;
    public Singer(String name, double sal)
    {
        super(sal, name);
        this.hits = 5;
    }
    public Singer(double sal, int hits)
    {
        super(sal, "Singer Name");
        this.hits = hits;
    }
    public Singer(int hits)
    {
        super(6532.6, "Some", "One");
        this.hits = hits;
    }
    public double value() {return this.hits * this.price();}
    public int getNum()   {return Artist.num; }
    public double calc(double d)    {return d * super.calc(10.2);    }
    public void print()      {super.print(); System.out.println("Singer");}
    public void act()        {System.out.println("I am singing");      }
}
```

Escribe en Java el título de la clase **Artist**, los atributos y los títulos de los métodos necesarios de la clase **Singer** y de la interface **IPerform** dados. No es necesario implementar los métodos de la clase **Artist**. No se deben alterar ni la clase **Singer** ni la interface **IPerform**.

/continúa en la página 24/

14. A continuación las clases A, B.

Los tres apartados α - γ se refieren a dichas clases, pero no están vinculados entre sí. Responde a los tres.

```
public class A {
    private int x;
    public A() { this.x = 0; }
    public A(int x) { this.x = x; }
    public int getX() { return x; }
    public void doubleX() { this.x = 2 * getX(); }
    public void tripleX() { this.x = 3 * getX(); }
    public void sub() { this.x = x - 1; }
    public void calc() { sub(); }
    public String toString() { return "xA="+this.x; }
}

public class B extends A {
    private int x;
    public B() { super(); this.x = 1; }
    public B(int x) { super(x); this.x = - x; }
    public B(int xA, int xB) { super(xA); this.x = xB; }
    public int getX() { return x; }
    public int baseX() { return super.getX(); }
    public void tenTimesX() { this.x = 10 * getX(); }
    public void tripleX() { this.x = 3 * getX(); }
    public void sub() { this.x = x - 2; }
    public String toString() { return super.toString()+" xB="+this.x+";" }
}
```

א. A continuación una serie de instrucciones cuya salida debe ser:

`xA=1 xB=20; xA=1 xB=20; xA=1 xB=20; xA=1 xB=20;`

En la serie de instrucciones se produjo un error de compilación. Corrige el error para que se produzca la salida correcta.

`A a1 = new B(1, 20);`

`Object obj = a1;`

`B b1 = a1;`

`A a2 = a1;`

`System.out.println(a1+" "+obj+" "+a2+" "+b1);`

ב. A continuación una serie de instrucciones. Muestra los objetos que se generan y, para cada objeto muestra los valores de sus atributos.

`A aa = new B(3, 10);`

`aa.sub();`

`Object[] ar = new Object[6];`

`ar[0] = new A();`

`ar[1] = new A(5);`

`ar[2] = new B();`

`ar[3] = new B(5);`

`ar[4] = new B(2, 4);`

`ar[5] = aa;`

`((A)ar[3]).tripleX();`

`((B)ar[4]).tenTimesX();`

(Presta atención : El apartado ב de la pregunta se halla en la página siguiente.)

ג. A continuación un método principal.

```
public static void main(String [] args)
{
    A a1 = new A(1);
    A a2 = new B(2, 99);
    /***
}
```

A continuación los segmentos (i)-(vi).

- (i) a2.doubleX();
 System.out.println(a2);
- (ii) a2.tenTimesX();
 System.out.println(a2.tenTimesX());
- (iii) if (a2 instanceof B)
 {
 a2.tenTimesX();
 System.out.println(a2);
 }
- (iv) ((B)a1).tenTimesX();
 System.out.println(a1);
- (v) a2.calc();
 System.out.println(a2);
- (vi) B bb = (B)a2;
 System.out.println(bb.baseX());

Para cada uno de los segmentos (i)-(vi) ejecuta:

- Escribe el segmento en lugar de /*** del método principal.
- Determina si el segmento es correcto o no.
- Si el segmento es correcto — escribe la salida que se obtiene después de la ejecución.

Si el segmento no es correcto — escribe si el error es de compilación o de tiempo de ejecución. /continúa en la página 27/

Programación orientada a objetos:

Si estudiaste en esta orientación y escribes en C#, responde a una de las preguntas 15-16. (25 puntos)

15. En esta pregunta hay dos apartados א - ב independientes uno del otro.
Responde a ambos apartados.

א. Se dan cinco clases B, X, Y, Z, Run. Los subapartados (1)-(4) se refieren a dichas clases. Los subapartados son independientes entre sí.
Responde a todos ellos.

(1) En cada una de las clases B, X, Y, Z se definió el método
`public virtual void Foo() {}`
y en la clase Run se definió el método
`public void Bar(Object g) {g.Foo(); }`

¿Se producirá un error de compilación? En caso afirmativo, explica por qué.

(2) Debes suponer que las clases X, Y, Z son herederas de la clase B.
En cada una de las clases X, Y, Z se definió el método
`public virtual void Foo() {}`
y en la clase Run se definió el método
`public void Bar(B g) {g.Foo(); }`

¿Se producirá un error de compilación? En caso afirmativo, explica por qué.

(3) Debes suponer que las clases X, Y, Z son herederas de la clase B.
En cada una de las clases B, X, Y, Z se definió el método
`public virtual void Foo() {}`
y en la clase Run se definió el método
`public void Bar(B g) {g.Foo(); }`

¿Se producirá un error de compilación? En caso afirmativo, explica por qué.

(4) Debes suponer que las clases X, Y, Z son herederas de la clase B.
En cada una de las clases B, X, Y, Z se definió el método
`public virtual void Foo() {}`
y en la clase Run se definió el método
`public void Bar(Object g) {g.Foo(); }`

¿Se producirá un error de compilación? En caso afirmativo, explica por qué.

(Presta atención : El apartado ב de la pregunta se halla en la página siguiente.)

ב. (Sin relación con el apartado א.)

A continuación se halla la clase **Singer** que hereda a la clase **Artist** y la interface **IPerform** que la clase **Singer** implementa.

```
interface IPerform
```

```
{
```

```
    void Act();
```

```
    int Train();
```

```
}
```

```
public class Singer : Artist , IPerform
```

```
{
```

```
    private int hits;
```

```
    public Singer(string name, double sal) : base(sal, name)
```

```
    {
```

```
        this.hits = 5;
```

```
    }
```

```
    public Singer(double sal, int hits) : base(sal, "Singer Name")
```

```
    {
```

```
        this.hits = hits;
```

```
    }
```

```
    public Singer(int hits) : base(6532.6, "Some", "One")
```

```
    {
```

```
        this.hits = hits;
```

```
    }
```

```
    public double Value()                {return this.hits * this.Price(); }
```

```
    public int GetNum()                  {return Artist.num;           }
```

```
    public override double Calc(double d) {return d * base.Calc(10.2); }
```

```
    public override void Print() {base.Print(); Console.WriteLine("Singer"); }
```

```
    public void Act()                    {Console.WriteLine("I am singing"); }
```

```
}
```

Escribe en C# el título de la clase **Artist**, los atributos y los métodos necesarios de la clase **Singer** y de la interface **IPerform** dados.

No es necesario implementar los métodos de la clase **Artist**, y no se deben alterar ni la clase **Singer** ni la interface **IPerform**.

16. A continuación las clases A, B.

Los tres apartados α - γ se refieren a dichas clases, pero no están vinculados entre sí. Responde a los tres.

```
public class A {
    private int x;
    public A() { this.x = 0; }
    public A(int x) { this.x = x; }
    public virtual int GetX() { return x; }
    public void DoubleX() { this.x = 2 * GetX(); }
    public virtual void TripleX() { this.x = 3 * GetX(); }
    public virtual void Sub() { this.x = x - 1; }
    public void Calc() { Sub(); }
    public override string ToString(){ return "xA="+this.x; }
}

public class B:A {
    private int x;
    public B() : base() { this.x = 1; }
    public B(int x) : base(x) { this.x = -x; }
    public B(int xA, int xB) : base(xA) { this.x = xB; }
    public override int GetX() { return x; }
    public int BaseX() { return base.GetX(); }
    public void TenTimesX() { this.x = 10 * GetX(); }
    public override void TripleX() { this.x = 3 * GetX(); }
    public override void Sub() { this.x = x - 2; }
    public override string ToString(){ return base.ToString()+" xB="+this.x+";"; }
}
```

(Presta atención : Los apartados de la pregunta se hallan en la página siguiente.)

א. A continuación una serie de instrucciones cuya salida debe ser:

$x_A=1 \ x_B=20$; $x_A=1 \ x_B=20$; $x_A=1 \ x_B=20$; $x_A=1 \ x_B=20$;

En la serie de instrucciones se produjo un error de compilación. Corrige el error para que se produzca la salida correcta.

```
A a1 = new B(1, 20);
```

```
Object obj = a1;
```

```
B b1 = a1;
```

```
A a2 = a1;
```

```
Console.WriteLine(a1+" "+obj+" "+a2+" "+b1);
```

ב. A continuación una serie de instrucciones. Muestra los objetos que se generan y para cada objeto, muestra los valores de sus atributos.

```
A aa = new B(3, 10);
```

```
aa.Sub();
```

```
Object[] ar = new Object[6];
```

```
ar[0] = new A();
```

```
ar[1] = new A(5);
```

```
ar[2] = new B();
```

```
ar[3] = new B(5);
```

```
ar[4] = new B(2, 4);
```

```
ar[5] = aa;
```

```
((A)ar[3]).TripleX();
```

```
((B)ar[4]).TenTimesX();
```

ג. A continuación un método principal.

```
public static void Main()
```

```
{
```

```
    A a1 = new A(1);
```

```
    A a2 = new B(2, 99);
```

```
    /***
```

```
}
```

A continuación los segmentos (i)-(vi).

(i) a2.DoubleX();

```
Console.WriteLine(a2);
```

(ii) a2.TenTimesX();

```
Console.WriteLine(a2.TenTimesX());
```

(iii) if (a2 is B)

```
{
```

```
    a2.TenTimesX();
```

```
    Console.WriteLine(a2);
```

```
}
```

(iv) ((B)a1).TenTimesX();

```
Console.WriteLine(a1);
```

(v) a2.Calc();

```
Console.WriteLine(a2);
```

(vi) B bb = (B)a2;

```
Console.WriteLine(bb.BaseX());
```

Para cada uno de los segmentos (i)-(vi) ejecuta:

- Escribe el segmento en lugar de `/***` en el método principal.
- Determina si el segmento es correcto o no.
- Si el segmento es correcto — escribe la salida que se obtiene después de la ejecución.

Si el segmento no es correcto — escribe si el error es de compilación o de tiempo de ejecución.

¡Buena Suerte!

Los derechos de autor están reservados al Estado de Israel.
Está prohibida su copia o difusión a menos que esté autorizada de
manera expresa por el Ministerio de Educación.

בהצלחה!

זכות היוצרים שמורה למדינת ישראל.
אין להעתיק או לפרסם אלא ברשות משרד החינוך.