Estado de Israel
Ministerio de Educación
Tipo de Examen: Bachillerato
Fecha de examen: verano 2017
Cuestionario número: 899381
Traducción al español (12)

Ciencias de la Computación
Según el Programa de Reforma para el Aprendizaje Significativo

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 x 10) + (1 x 15) — 25 puntos
Segunda parte -Esta parte consta de tres preguntas de las que debes responder a dos.
(2x25) — 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.
(1x25) — 25 puntos
Total — 100 puntos

ג. Material auxiliar permitido:
Cualquier material auxiliar, salvo calculadoras programables.

ד. Instrucciones especiales:
(1) En la primera y segunda partes, 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 **sólamente 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 "טיוטה" 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!

מדינת ישראל
משרד החינוך
סוג הבחינה: בגרות
מועד הבחינה: קיץ תשע"ז, 2017
מספר השאלון: 899381
תרגום לספרדית (12)

מדעי המחשב
על פי תכנית הרפורמה ללמידה משמעותית

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.
— (10×1) + (15×1) — 25 נקודות

פרק שני — בפרק זה שלוש שאלות, ומהן עליך לענות על שתיים.
— (25×2) — 50 נקודות

פרק שלישי — בפרק זה שאלות בארבעה מסלולים שונים. ענה על שאלה אחת במסלול שלמדת. —
(25×1) — 25 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 והשני 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 ואסמבלי, מבוא לחקר ביצועים, מודלים חישוביים, תכנות מונחה עצמים.

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

כתוב במחברת הבחינה בלבד, בעמודים נפרדים, כל מה שברצונך לכתוב **כטיוטה** (ראשי פרקים, חישובים וכדומה). רשום "טיוטה" בראש כל עמוד טיוטה. רישום טיוטות 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. Escribe un método externo big en Java o Big en C# , que reciba un vector de de números enteros. El método devolverá el índice del término cuyo valor es el mayor del vector. Si el número aparece más de una vez en el vector, el método devolverá el menor de los índices que aparezcan.

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

2. Durante la organización de un evento internacional de juegos deportivos se definió una clase **Game** que posee tres atributos: nombre del juego
— gameName del tipo cadena de caracteres, número de jugadores
— numPlayers de tipo entero, si el juego se desarrolla en el agua
— isWater de tipo booleano.
Debes suponer que para todos los atributos se han definido 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 **Game** que reciba valores para cada atributo.

ב. Se definió una clase País— **Country** que posee dos atributos: nombre del país — countryName del tipo cadena de caracteres, y un vector unidimensional games de magnitud 43 de tipo **Game**. El vector contiene los juegos en los que el país interviene.
Debes suponer que para todos los atributos se han definido en Java métodos get y set**,** y en C# métodos Get y Set.

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

(2) Escribe en Java o C# en la clase **Country** un método constructor que reciba el nombre del país. El método construirá un objeto para el país que no intervenga en ningún juego.

(3) Escribe en Java o C# en la clase **Country** un método booleano que reciba el nombre del juego y devuelva true si el país interviene en el juego, de lo contrario el método devolverá false.

/continúa en la página 3/

3. A continuación un segmento de programa escrito en Java y en C#.

Java

```
int i , j;
for (i = 1; i < = n; i++)
{
      a[i] = i;
}
for (i = 2; i < n; i++)
{
      if (a[i] != 0)
      {
            for ( j = i + 1; j < = n; j++)
            {
                  if (a[ j] % a[i] = = 0)
                  {
                        a[ j] = 0;
                  }
            }
      }
}

for (i = 1; i < = n; i++)
{
      if (a[i] != 0)
      {
            System.out.println(a[i]);
      }
}
```

C#

```
int i , j;
for (i = 1; i < = n; i++)
{
      a[i] = i;
}
for (i = 2; i < n; i++)
{
      if (a[i] != 0)
      {
            for ( j = i + 1; j < = n; j++)
            {
                  if (a[ j] % a[i] = = 0)
                  {
                        a[ j] = 0;
                  }
            }
      }
}

for (i = 1; i < = n; i++)
{
      if (a[i] != 0)
      {
            Console.Writeln(a[i]);
      }
}
```

א. Escribe qué se imprimirá para n = 15.

ב. Escribe en una oración qué realiza el segmento de programa, o sea, formula el problema para el cual el segmento de programa es la solución.

/continúa en la página 4/

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 se halla la definición de cinco métodos que operan sobre una estructura de datos cualesquiera.

Presta atención: Los nombres de los métodos no están escritos en Java o en C#.

insert(x) — Un método que ingresa en la estructura un término de valor x de tipo entero.

showMin() — Un método que devuelve el valor más bajo de la estructura, sin alterar la estructura.

getMax() — Un método que devuelve el término cuyo valor es el más grande en la estructura, y lo retira de la estructura. Si hubiera más de un término como ese, el método lo devolverá y retirará el que apareció primero.

exists(x) — Un método booleano que devuelve true si el término cuyo valor es x existe en la estructura. De lo contrario — el método devuelve false.

div7() — Un método booleano que devuelve true si existe en la estructura un término cuyo valor es divisible por 7 (sin resto). De lo contrario — el método devuelve false.

Se quiere proponer una estructura de datos que cumplen con diversos requisitos de complejidad para implementar métodos de entre los cinco que fueron definidos.

Ejemplo: Se quiere proponer una estructura de datos que permita ejecutar sobre ella los métodos insert , showMin con complejidad O(1) , y el método exists , getMax con complejidad O(n) . Para ello, definiremos la estructura de datos y explicaremos cómo se implementaron los métodos.

Presta atención: En esta estructura no hay necesidad de referirse al método div7 .

/continúa en la página 5

Una estructura de datos adecuada está compuesta por:

- Una lista doblemente enlazada, lst de tipo entero.
- Puntero sobre el término mínimo que fue introducido en la lista, min .

Los métodos serán ejecutados así:

Método	**Explicación de cómo será implementado**	**Razones por las cuales la implementación responde a las exigencias de complejidad**
insert(x)	— Introducción del término x al comienzo de la lista. — Si el término es menor que el mínimo hasta ahora, una actualización del puntero, min, tal que apunte al nuevo término.	— Introducción de un término al comienzo de la lista — O(1) . — Si fue introducido un término que es menor que el mínimo: una actualización del puntero al término mínimo. — O(1) . — Si fue introducido un primer término — Introducción del término al comienzo de la lista en O(1) y actualización del puntero al término primero que es también el mínimo. En total — O(1).
showMin()	Devolución del valor del término al que apunta min.	Devolución del valor — O(1) .
exists(x)	Paso sobre la lista lst y búsqueda del término de valor x .	En el peor de los casos — pasar sobre toda la lista — O(n) .
getMax()	Paso sobre la lista lst, búsqueda del término cuyo valor es máximo, y su retiro de la lista.	— Búsqueda del término cuyo valor es máximo O(n) . — Retirarlo de la lista O(1) .

A continuación dos apartados ב-א. Para cada uno de los apartados debes proponer una estructura de datos adecuada que responda a las exigencias indicadas en el apartado. La estructura puede estar compuesta de una combinación de varias estructuras y tipos que estudiaste. Para cada uno de los métodos explica cómo lo implementarás y justifica por qué la implementación responde a las exigencias (como se exhibe en la tabla de ejemplo). No es necesario implementar los métodos.

א. Ejecución de los métodos insert , exists con complejidad O(n) , y ejecución de los métodos getMax , showMin con complejidad O(1) .

ב. Ejecución de los métodos insert , getMax con complejidad O(n) , y ejecución del método div7 con complejidad O(1).

/continúa en la página 6/

5. A continuación un programa escrito en Java y en C# .
El método what en Java y What en C# recibe un vector unidimensional que contiene números enteros, y un número entero k, con k > 0. Todos los números del vector se hallan entre 0 y k (incluido).

Java

```
public class Program
{
    public static int[]  what(int[] arr, int k)
    {
       int n = arr.length;
       int[] b = new int[n];
       int[] c = new int[k+1];

       for (int i = 0; i < k+1; i++) c[i] = 0;
       for (int j = 0; j < n; j++)  c[arr[ j]] = c[arr[ j]] + 1;      //*

       for (int i = 1; i < k+1 ; i++) c[i] = c[i] + c[i− 1];       //**

       for (int j = n− 1 ; j >= 0; j--)                              //***
       {
          b[c[arr[ j]] − 1] = arr[ j];
          c[arr[ j]] = c[arr[ j]] −1;
       }
       return b;
    }


    public static void main(String[] args)
    {
          int[] arr = new int[] {5,0,2,1,3,0,5};
          arr = what(arr, 5);
          for (int i = 0; i < arr.length; i++) System.out.println(arr[i]);
    }
}
```

/continúa en la página 7/

C#

```
public class Program
{
    public static int[]  What(int[] arr, int k)
    {
       int n = arr.Length;
       int[] b = new int[n];
       int[] c = new int[k+1];

       for (int i = 0; i < k+1; i++) c[i] = 0;
       for (int j = 0; j< n; j++)  c[arr[ j]] = c[arr[ j]] + 1;    //*

       for (int i = 1; i< k+1 ; i++) c[i] = c[i] + c[i− 1];      //**

       for (int j = n− 1 ; j >= 0; j--)                          //***
       {
          b[c[arr[ j]] −1] = arr[ j];
          c[arr[ j]] = c[arr[ j]] −1;
       }
       return b;
    }


    public static void Main(string[] args)
    {
         int[] arr = new int[] {5,0,2,1,3,0,5};
         arr = What(arr, 5);
         for (int i = 0; i < arr.Length; i++) Console.WriteLine(arr[i]);
    }
}
```

א. **(1)** Dibuja el vector c después de la ejecución del bucle indicado //*.
(2) Dibuja el vector c después de la ejecución del bucle indicado //**.
(3) Realiza el seguimiento de la ejecución del bucle indicado //*** .
En el seguimiento debes mostrar: j , arr[j] , c[arr[j]] ,
b[c[arr[j]] − 1] , el vector b y el vector c .

ב. ¿Qué ejecuta el método what en Java o What en C# ?

ג. ¿Cuál es la complejidad de tiempo de ejecución del método what en Java o What en C# ? Justifica tu respuesta.

/continúa en la página 8/

6. **א**. Implementa el método externo exist en Java o Exist en C#.
El método recibe un árbol binario t de tipo entero y un número entero x . El método devuelve true si el árbol posee un nodo cuyo valor es x, en caso contrario el método devolverá false. Si el árbol es vacío, el método devolverá false .

ב. A continuación el método en Java check(t1, t2) y en C# Check(t1, t2).
El método recibe dos árboles binarios no vacíos de tipo entero, t1 y t2, y devuelve una lista que contiene todos los números que se hallan en el árbol t1 y no se hallan en el árbol t2. El método convoca un método adicional que recibe tres parámetros.

Java

```
public static Node<Integer> check(BinNode<Integer> t1, BinNode<Integer> t2)
{
    Node<Integer> first = new Node<Integer> (− 1);
    first = check(t1 , t2 , first);
    return first.getNext();
}
```

C#

```
public static Node<int> Check(BinNode<int> t1 , BinNode<int> t2)
{
    Node<int> first = new Node<int> (− 1);
    first = Check(t1 , t2 , first);
    return first.GetNext();
}
```

Implementa el método:
en Java:

```
public static Node<Integer> check(BinNode<Integer> t1,
                                  BinNode<Integer> t2 , Node<Integer> list)
```

o en C#:

```
public static Node<int> Check(BinNode<int> t1 ,
                              BinNode<int> t2 , Node<int> list)
```

Puedes utilizar el método que implementaste en el apartado א.

ג. ¿Cuál es la complejidad de tiempo de ejecución del método que implementaste en el apartado ב ? Justifica tu respuesta.

/continúa en la página 9/

Tercera parte (25 puntos)

Esta parte consta de preguntas según cuatro orientaciones:
Sistemas de computación y Assembler, págs. 9-11
Introducción a la investigación operativa, págs. 12-16
Modelos computacionales, pág. 17 .
Programación orientada a objetos en Java, págs. 18-21.
Programación orientada a objetos en C#, págs. 22-25.

Responde a una pregunta de la orientación que estudiaste.

Sistemas de computación y Assembler

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

7. En esta pregunta hay dos apartados, א-ב, que no tienen relación entre sí. Responde a ambos.

א. Te presentamos un segmento de programa en Assembler tal que en AL y en AH hay números diferentes entre sí, no indicados.

```
        MOV  AX , 7F80H
        CMP  AH , AL
        JL   SMALL
BIG:    MOV  BL , AH
        JMP  SOF
SMALL:  MOV  BL , AL
SOF:
```

(1) ¿Cuál será el valor de BL al final de la ejecución del segmento de programa?

(2) ¿Qué ejecuta el segmento de programa?

(3) En el segmento de programa, en lugar de la instrucción JL SMALL se escribió la instrucción JB SMALL .
Finalizada la ejecución del segmento de programa después de esa sustitución, ¿será el valor de BL distinto del valor que se hubiera obtenido al finalizar la ejecución antes de la sustitución de la instrucción? Explica tu respuesta.

/continúa en la página 10/

ב. (Sin relación con el apartado א.)
A continuación seis proposiciones. Para cada una de ellas determina si es cierta o no. Si la proposición no es cierta explica por qué.

(1) $10011110101110_2 > 27AE_{16}$

(2) $27AE_{16} > 10159_{10}$

(3) Las funciones de los acumuladores BP y SP son:
BP determina en qué dirección se registrará una palabra en una pila, y de qué dirección en la pila será extraída la palabra.
SP habilita a un procedimiento para extraer parámetros de la pila, sin la intervención de BP .

(4) Una vez finalizada la lectura y la ejecución de cada instrucción, el valor de IP aumenta siempre en 1 .

(5) A continuación dos segmentos de programa en Assembler.

Segmento 2	Segmento 1
MOV DX , 0000H	MOV DX , 0000H
MOV AX , 0064H	MOV AX , 0064H
DIV AX	DIV AL

Finalizada la ejecución de los dos segmentos el contenido de AX será 0001H .

(6) A continuación dos segmentos de programa en Assembler.

Segmento 2	Segmento 1
MOV AL , 5BH	MOV AL , 5BH
MOV CL , 9	MOV CL , 8
ROL AL , CL	ROL AL , CL

Finalizada la ejecución del segmento 1 el valor de AL será igual al valor que se le sustituyó en la primera instrucción,
y también, finalizada la ejecución del segmento 2 el valor de AL será igual al valor que se le sustituyó en la primera instrucción.

/continúa en la página 11/

8. Se dan dos vectores ARR1 y ARR2 . Los dos vectores son de la misma magnitud y cada uno de sus términos tiene la magnitud de una palabra. En cada término, en cada uno de los vectores, se halla almacenado un número hexadecimal entero de cuatro cifras en el intervalo 1000H - FFFFH . Asimismo, se da la variable K de magnitud de un byte en el que se halla almacenada la magnitud de los vectores ARR1 y ARR2 .

Escribe en Assembler un segmento de programa que almacene en el acumulador DX el número de pares de números (uno del vector ARR1 y el otro del vector ARR2) que cumplen las siguientes condiciones:

(i) Los dos números del par tienen el mismo índice en el vector.

(ii) Los dos números del par están compuestos de las mismas cifras pero en orden invertido.

Ejemplo: para los vectores ARR1 y ARR2 de magnitud 5 que se hallan a continuación, se almacenará en el acumulador DX el número 2 .

ARR1	2025H	1061H	1492H	5777H	1948H
ARR2	1984H	1601H	2914H	9999H	8491H

/continúa en la página 12/

Introducción a la Investigación operativa [חקר ביצועים]

Si estudiaste en esta orientación, responde a una de las preguntas 9-10 (25 puntos).

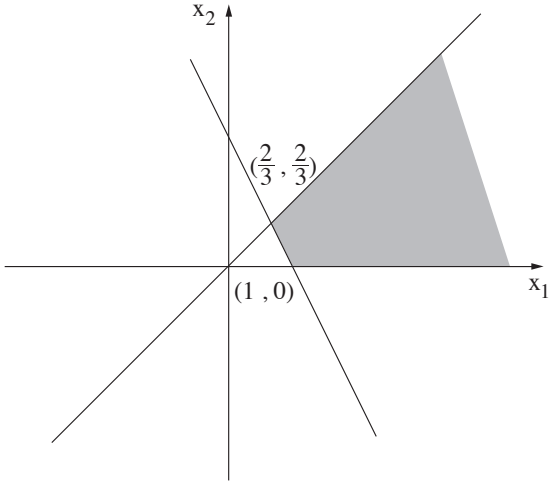
9. Se da un problema de programación lineal:

$\max\{z = 5x_1 - x_2\}$

Sometido a las restricciones siguientes:

$$x_2 \leq x_1$$
$$2x_1 + x_2 \geq 2$$
$$x_1 \geq 0$$
$$x_2 \geq 0$$

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



Cada uno de los apartados א-ה que se encuentran a continuación se refieren al problema de programación lineal dado.

Los apartados א-ה son independientes entre sí. Responde a todos los apartados.

Para los apartados א- ד se dan cuatro proposiciones i-iv, y para cada apartado sólo una de las proposiciones es cierta.

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.

/continúa en la página 13/

En cada uno de los apartados א- ד determina cuál es la proposición verdadera, cópiala en tu cuadernillo y fundamenta tu afirmación.

- Si elegiste la proposición i en un apartado cualquiera, debes encontrar la solución óptima única y el valor de la función objetivo en dicha solución.
- Si elegiste la proposición ii en un apartado cualquiera, debes anotar la solución óptima general del problema, y el valor de la función objetivo en el intervalo de soluciones óptimas.

א. ¿Qué proposición es cierta para el problema de programación lineal dado al comienzo de la pregunta? Justifica tu respuesta.

ב. Se cambia solamente la función objetivo del problema dado al comienzo de la pregunta a: $\max\{z = -5x_1 + x_2\}$
¿Qué proposición es cierta después del cambio de la función objetivo? Justifica tu respuesta.

ג. Se cambia solamente la función objetivo del problema dado al comienzo de la pregunta a: $\min\{z = -5x_1 + x_2\}$.
¿Qué proposición es cierta después del cambio de la función objetivo? Justifica tu respuesta.

ד. Se cambia solamente la función objetivo del problema dado al comienzo de la pregunta a: $\max\{z = -2x_1 + 2x_2\}$
¿Qué proposición es cierta después del cambio de la función objetivo? Justifica tu respuesta.

ה. Se agrega una restricción adicional al problema dado al comienzo de la pregunta, y es: $2x_1 - x_2 \leq 2$, y se cambia la función objetivo del problema dado al comienzo de la pregunta a: $\max\{z = ax_1 + x_2\}$ siendo $(-1) < a < 2$, y el valor máximo que alcanza la función objetivo es 4 en el intervalo de soluciones posibles. Calcula el valor de a.

/continúa en la página 14/

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

א. $G = (V, E)$ es un grafo **no orientado** representado por la siguiente lista de adyacencias:

a	→	b	→	c	→	d	→	‖
b	→	a	→	c	→	‖		
c	→	a	→	b	→	‖		
d	→	a	→	f	→	e	→	‖
e	→	d	→	f	→	‖		
f	→	e	→	d	→	‖		

(1) Dibuja el grafo G representado por la lista de adyacencias.

(2) ¿Es el grafo G dado un grafo conexo? Justifica tu respuesta.

(3) Opera un algoritmo de búsqueda de profundidad (DFS) sobre el grafo dado comenzando en el vértice a.
Dibuja en tu cuadernillo de examen solo el árbol de expansión (DFS) [העץ הפורש] que se obtiene.
Básate en la representación dada en la lista de adyacencias.

(4) Opera un algoritmo de búsqueda en anchura (BFS) sobre el grafo dado comenzando en el vértice a.
Dibuja en tu cuadernillo de examen solo el árbol de expansión (BFS) [העץ הפורש] que se obtiene.
Básate en la representación dada en la lista de adyacencias.

/continúa en la página 15/

ב. (Sin relación con el apartado א.)

(1) En la tabla que está a continuación se da una solución básica posible de un problema de transporte y se dan los valores de u_1 , u_2 , u_3 , v_1 , v_2 , v_3 .

Orígenes	**Destinos**			**Oferta**	u_i
	1	2	3		
1	2 (20)	5	7	20	2
2	1	1 (10)	4	10	0
3	0	1	8 (10)	15	0
Demanda	20	15	10		
v_j	0	1	8		

Copia la tabla en tu cuadernillo y complétala teniendo en cuenta los valores de los u_i y los v_j de modo que se obtenga una solución básica posible.

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

/continúa en la página 16/

(2) En la tabla que está a continuación se da una solución básica posible de un problema de transporte y se dan los valores de u_1 , u_2 , u_3 , v_1 , v_2 , v_3 .

Orígenes	**Destinos**			**Oferta**	u_i
	1	2	3		
1	14	15 (130)	17 (50)	180	0
2	10 (80)	8 (20)	14	100	−7
3	15	20	18 (80)	80	1
Demanda	80	150	130		
v_j	17	15	17		

¿Se trata de una solución óptima? Justifica tu respuesta.

(3) En la tabla que está a continuación se da una solución básica posible de un problema de transporte, y se da $v_1 = 0$

Orígenes	**Destinos**			**Oferta**	u_i
	1	2	3		
1	10 (20)	25	30	20	
2	10 (30)	22	14 (20)	50	
3	18	20 (40)	20 (20)	60	
Demanda	50	40	40		
v_j	0				

Copia la tabla en tu cuadernillo y completa en ella los valores de u_1 , u_2 , u_3 , v_2 , v_3 .

/continúa en la página 17/

Modelos computacionales

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

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

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

Java

```
boolean foo(String str) {
    int cntrA = 0;
    int cntrC = 0;
    for (int i= 0; i < str.length; i++) {
        if (str[i] == 'a') cntrA++;
        if (str[i] == 'c') cntrC++;
    }
    if ((cntrA % 2 == 0) &&
        (cntrC % 3 == 0))
        return true;
    return false;
}
```

C#

```
bool Foo(string str) {
    int cntrA = 0;
    int cntrC = 0;
    for (int i= 0; i < str.Length; i++) {
        if (str[i] == 'a') cntrA++;
        if (str[i] == 'c') cntrC++;
    }
    if ((cntrA % 2 == 0) &&
        (cntrC % 3 == 0))
        return true;
    return false;
}
```

(1) Escribe el lenguaje L sobre el alfabeto {a , c} que es la colección de todas las palabras para las cuales el método dado devuelve true.

(2) Construye un autómata finito determinístico que acepte el lenguaje L.

ב. (Sin relación con el apartado א.)
Construye un autómata finito no determinístico sobre el alfabeto {a , b} que acepte todas las palabras que contienen por lo menos una aparición de cada una de las secuencias: bbb , aaba , ababa .

12. Se da el lenguaje L sobre el alfabeto {a, b, c}.

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

א. Construye un autómata de pila [מחסנית] que acepte el lenguaje L.

ב. Construye una máquina de Turing que acepte el lenguaje L.

/continúa en la página 18/

Programación orientada a objetos

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

13. A continuación las interfaces IOne, ITwo, IThree y IFour y las clases Alpha, Beta, Gamma y Program.

```
public interface IOne {
  public boolean firstA(Object x);
  public void firstB(int num);
}

public interface ITwo {
  public int second();
}

public interface IThree {
  public int third();
}
public  interface IFour {
  public int fourth();
}

public class Alpha implements IOne {}
public class Beta implements ITwo, IFour {}
public class Gamma implements ITwo, IThree {}
public class Program  {
  public static void main(String[] args) {}
}
```

א. Escribe los nombres de los métodos que hay que implementar en cada una de las clases Alpha, Beta y Gamma y explica por qué.

/continúa en la página 19/

Debes suponer que en cada una de las clases se implementó el método que escribiste en el apartado א.

ב. ¿Es la definición public class Omega implements IFour extends Beta { } correcta?

Si no es correcta explica por qué.

ג. Para cada uno de los segmentos i-iv que se hallan a continuación, si se los escribe con el método main, determina si el método main que se obtiene es correcto o no lo es. Justifica tu respuesta.

i
```
ITwo  x1 = new ITwo();
```

ii
```
Beta  b = new Beta();
```

iii
```
Alpha  a = new Alpha();
IOne  x2 =  a;
```

iv
```
Gamma  c = new Gamma ();
IOne  x3  = c ;
```

ד. Para cada una de las exigencias i-ii que siguen determina si es posible ejecutarla por medio de la escritura de una instrucción o instrucciones del método main. En caso de que fuere posible, escribe la instrucción adecuada o las instrucciones adecuadas. Si no fuere posible, explica por qué.

i Operar el método second sobre un objeto de tipo Beta .

ii Transformación de un objeto de tipo Gamma para que sea un objeto de tipo Alpha .

/continúa en la página 20/

14. En esta pregunta hay cuatro apartados א - ד independientes uno del otro. Responde a todos los apartados.

א. Se da la clase A y la clase B, hereditaria de A.

La instrucción A a1 = new B(); sufre una compilación y se ejecuta de manera correcta.

Para cada una de las proposiciones i-iii que se hallan a continuación, determina si la oración es verdadera o falsa. Justifica tus respuestas.

i Es imposible escribir la instrucción Object obj = a1; puesto que se requiere una transformación explícita (casting).

ii Es imposible escribir la instrucción A a2 = a1; puesto que se requiere una transformación explícita (casting).

iii Es imposible escribir la instrucción B b1 = a1; puesto que se requiere una transformación explícita (casting).

ב. Se da el método main en la clase Program :

```
public static void main(String[] args) {
  C c = new A();
  B bl = (B) (new A());
  B  b2 = new D();
  A  a = new D();
}
```

Determina cuál de las posibilidades siguientes i - v describe una relación entre las clases, según el método main dado.

i C hereda de A , B hereda de A , D hereda de A.

ii A hereda de C , B hereda de A , D hereda de A .

iii C hereda de A , B hereda de A , D hereda de B .

iv A hereda de C , C hereda de B , D hereda de A .

v A hereda de C , B hereda de A , D hereda de B , C hereda de B .

/continúa en la página 21/

ג. Se dan las definiciones de las clases School y City :

```
public class City{      (1)     int students;  }
public  class  School extends City {
  void poll() { System.out.println("There are "+ students + " in the school ") ; }
}
```

Para que no se produzcan errores de compilación se puede completar la parte indicada (1) por medio de una de las posibilidades que se hallan a continuación. Elige la posibilidad adecuada y cópiala en tu cuadernillo.

- Escribe public , private , o protected .
- Escribe solamente public o private .
- Escribe solamente public o protected .
- Escribe solamente protected .
- Escribe solamente public .
- Escribe solamente private .

ד. A continuación se halla la definición de la clase Quest :

```
public class Quest {
   private int  num;
   private static int count = 0;

   public Quest()
   {
      count++;
      num = count;
   }
   public void printNow()
   {
      System.out.println (num + "" + count);
   }
}
```

Determina cuántos objetos hay que producir de la clase Quest , y sobre cuál de ellos hay que ejecutar el método PrintNow , para que la salida sea 35 .

/continúa en la página 22/

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

15. A continuación las interfaces IOne, Itwo, IThree y IFour y las clases Alpha, Beta, Gamma y Program.

```
interface IOne {
  bool FirstA(Object x);
  void FirstB(int num);
}

interface ITwo {
  int Second();
}

interface IThree {
  int Third();
}
interface IFour {
  int Fourth();
}

public class Alpha : IOne {}
public class Beta : ITwo, IFour {}
public class Gamma : ITwo, IThree {}
public class Program  {
  public static void Main() {}
}
```

א. Escribe los nombres de los métodos que hay que implementar en cada una de las clases Alpha, Beta y Gamma y explica por qué.

/continúa en la página 23/

Debes suponer que en cada una de las clases se implementó el método que escribiste en el apartado א.

ב. ¿Es la definición public class Omega IFour, Beta { } correcta?
Si no es correcta explica por qué.

ג. Para cada uno de los segmentos i-iv que se hallan a continuación, si se los escribe con el método Main, determina si el método Main que se obtiene es correcto o no lo es. Justifica tu respuesta.

i
```
ITwo  x1 = new ITwo();
```

ii
```
Beta  b = new Beta();
```

iii
```
Alpha  a = new Alpha();
IOne  x2 =  a;
```

iv
```
Gamma  c = new Gamma ();
IOne  x3  = c ;
```

ד. Para cada una de las exigencias ii-i que siguen determina si es posible ejecutarla por medio de la escritura de una instrucción o instrucciones del método Main. En caso de que fuere posible, escribe la instrucción adecuada o las instrucciones adecuadas. Si no fuere posible, explica por qué.

i Operar el método Second sobre un objeto de tipo Beta .

ii Transformación de un objeto de tipo Gamma para que sea un objeto de tipo Alpha .

/continúa en la página 24/

16. En esta pregunta hay cuatro apartados א - ד independientes uno del otro. Responde a todos los apartados.

א. Se da la clase A y la clase B, hereditaria de A.

La instrucción A a1 = new B(); sufre una compilación y se ejecuta de manera correcta.

Para cada una de las proposiciones i-iii que se hallan a continuación, determina si la oración es verdadera o falsa. Justifica tus respuestas.

i Es imposible escribir la instrucción Object obj = a1; puesto que se requiere una transformación explícita (casting).

ii Es imposible escribir la instrucción A a2 = a1; puesto que se requiere una transformación explícita (casting).

iii Es imposible escribir la instrucción B b1 = a1; puesto que se requiere una transformación explícita (casting).

ב. Se da el método Main en la clase Program :

```
public static void  Main(string[] args) {
  C c = new A();
  B bl = (B) (new A());
  B  b2 = new D();
  A  a = new D();
}
```

Determina cuál de las posibilidades siguientes i - v describe una relación entre las clases, según el método Main dado.

i C hereda de A , B hereda de A , D hereda de A.

ii A hereda de C , B hereda de A , D hereda de A .

iii C hereda de A , B hereda de A , D hereda de B .

iv A hereda de C , C hereda de B , D hereda de A .

v A hereda de C , B hereda de A , D hereda de B , C hereda de B .

/continúa en la página 25/

ג. Se dan las definiciones de las clases School y City :

```
public class City{    (1)    int students;  }
public  class  School : City {
  void Poll() { Console.WriteLine("There are "+ students + " in the school ") ; }
}
```

Para que no se produzcan errores de compilación se puede completar la parte indicada (1) por medio de una de las posibilidades que se halla a continuación. Elige la posibilidad adecuada y cópiala en tu cuadernillo.

- Escribe public , private , o protected .
- Escribe solamente public o private .
- Escribe solamente public o protected .
- Escribe solamente protected .
- Escribe solamente public .
- Escribe solamente private .

ד. A continuación se halla la definición de la clase Quest :

```
public class Quest  {
        private int num;
        private static int count = 0;

        public Quest()
        {
                count++;
                num = count;
        }
        public void PrintNow()
        {
                Console.
WriteLine(num+""+count);
        }
}
```

Determina cuántos objetos hay que producir de la clase Quest , y sobre cuál de ellos hay que ejecutar el método PrintNow , para que la salida sea 35 .

¡Buena Suerte! **בהצלחה!**