

Ciencias de la Computación

מדעי המחשב

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 -	$(1 \times 10) + (1 \times 15) -$	25 puntos
Segunda parte -	$(2 \times 25) -$	50 puntos
Tercera parte -	$(1 \times 25) -$	25 puntos
Total	-	100 puntos

ג. Material auxiliar permitido:

Cualquier material auxiliar, salvo calculadoras programables.

ד. Instrucciones especiales:

- (1) En la primera 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 la que escribes: C# o Java.
- (3) **Indica en la cobertura externa** del cuadernillo el nombre de la **orientación según la cual estudiaste**, una de estas 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 en el cuadernillo del examen solamente. Escribe "טייטה" en el encabezado de toda página usada de borrador.
La escritura de borrador en páginas que no están en el cuadernillo del examen podría causar la anulación del examen.

¡Buena Suerte!

הוראות לנבחן

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח ההערכה:
בשאלון זה שלושה פרקים.

פרק ראשון -	$(10 \times 1) + (15 \times 1) -$	25 נקודות
פרק שני -	$(25 \times 2) -$	50 נקודות
פרק שלישי -	$(25 \times 1) -$	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 בפרקים הראשון והשני
כתוב בשפה אחת בלבד - Java או C#.
- (2) **רשום על הכריכה החיצונית** של המחברת באיזו **שפה** אתה כותב - Java או C#.
- (3) **רשום על הכריכה החיצונית** של המחברת את שם **המסלול שלמדת**.
המסלול הוא אחד מארבעת המסלולים האלה:
מערכות מחשב ואסמבלי, מבוא לחקר ביצועים, מודלים חישוביים,
תכנות מונחה עצמים.

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

כתוב במחברת הבחינה בלבד. רשום "טייטה" בראש כל עמוד המשמש טייטה.
כתובת טייטה בדפים שאינם במחברת הבחינה עלולה 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 exact en Java o Exact en C# que reciba un vector –arr de tipo cadena de caracteres, y un número entero – num. El método devolverá el número de cadenas de caracteres en el vector arr, cuya longitud es igual a num.

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

2. En el sistema de computación de la tienda de equipos de camping se da la clase **Flashlight** que tiene dos atributos:

model – nombre del modelo de linterna de tipo cadena de caracteres.

price – precio de la linterna número de tipo real.

- א. Escribe en Java o C# el título de la clase **Flashlight** y sus atributos, y escribe un método constructor que reciba valores para el modelo de la linterna y el precio de la linterna.

- ב. (1) Con vistas a una excursión debes equiparte con tres linternas diferentes.

Escribe un método externo threeFlashlights en Java o ThreeFlashlights en C# que reciba un vector arr de objetos diferentes de tipo **Flashlight**, y el número total de tipo real.

El método debe encontrar e imprimir por una sola vez los modelos de las tres linternas cuya suma de precios es el número total.

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

Debes suponer que existe una única terna de linternas cuya suma de precios es el número total.

- (2) ¿Cuál es la complejidad del tiempo de ejecución del método que escribiste en el subapartado ב(1). Justifica tu respuesta.

3. En el sistema de computación de la compañía de venta de automóviles usados se definió la clase de nombre **Car** que posee tres atributos:

licenseNum — número de registro de tipo cadena de caracteres.

hadAccident — una variable de tipo booleano que recibe true si el automóvil tuvo un accidente; de lo contrario recibe false .

price — precio del automóvil de tipo número entero.

Debes suponer que para la clase se definió un método constructor que recibe valores para cada uno de los atributos de la clase, y que para cada atributo se definieron en Java, métodos get y set, y en C# métodos Get y Set .

א. Escribe en la clase **Car** un método interno booleano de nombre range en Java o Range en C# , que reciba dos números de tipo entero min y max . El método devolverá true si el precio del automóvil se halla entre min y max (incluidos), en caso contrario el método devolverá false .

Debes suponer que max es mayor que min .

ב. Se da la clase **AllCars** que posee dos atributos:

cars — un vector unidimensional de tipo **Car**.

num — número actual de automóviles que hay en la compañía, de tipo número entero.

A continuación se halla el método constructor de la clase:

```
public AllCars ( int max )  
{  
    this.cars = new Car [max];  
    this.num = 0;  
}
```

(1) Escribe en la clase **AllCars** un método interno booleano de nombre addCar en Java o AddCar en C# , que recibe un objeto de tipo **Car** . El método lo agrega al vector de automóviles en el primer lugar libre y devuelve true .

Si el vector está lleno, el método devolverá false .

Debes suponer que no hay lugares libres desde el comienzo del vector hasta el número num (num no incluido).

(2) Escribe en la clase **AllCars** un método interno de nombre print en Java o Print en C# , que reciba dos números de tipo entero min y max (max mayor que min).

El método imprimirá el número del registro de cada automóvil del vector cars, que no tuvo accidentes y cuyo precio se halla entre los números min y max (incluidos)

Es obligatorio utilizar el método que fue definido en el apartado א.

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. **Presta atención:** Esta pregunta tiene una versión en Java (en las páginas 4 y 5) y una versión en C# (en las páginas 6 y 7). Resuelve según lo que hayas estudiado.

Para los que resuelven en Java

Se da el método secret1:

```
public static boolean secret1 (int num, int digit )  
{  
    if (num < 10)  
        return ( num % 2 == digit % 2 );  
    if ( num % 2 != digit % 2 )  
        return false;  
    return secret1 ( num / 10, digit );  
}
```

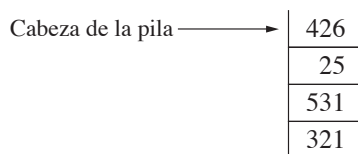
- ⌘. (1) Escribe qué devolverá la convocatoria del método `secret1 ( 937, 5 )`. Debes mostrar el seguimiento.
- (2) Ofrece un ejemplo del número `num` de 3 cifras, para el cual la convocatoria del método `secret1(num,5)` devuelva un valor diferente de lo que se obtuvo en el subapartado ⌘ (1). Debes mostrar el seguimiento.
- (3) Escribe en una oración que es lo que ejecuta el método booleano `secret1`, es decir cuál es la pregunta a la que el método devuelve `true` o `false`.

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

Se da el método secret2 :

```
public static boolean secret2 ( Stack <Integer> s )
{
    boolean ok;
    int x;
    if ( s.isEmpty () )
        ok = true ;
    else
    {
        x = s.pop () ;
        if ( ! (secret1 (x, x % 10)) )
            ok = false;
        else
            ok = secret2 (s);
    }
    return ok ;
}
```

- ב. (1) Para la pila s que se halla a continuación:



Escribe qué devolverá el método secret2 . Debes mostrar el seguimiento. (No es necesario mostrar el seguimiento del método secret1).

- (2) Escribe en una oración que es lo que ejecuta el método booleano secret2 , es decir cuál es la pregunta a la que el método devuelve true o false .

Para los que resuelven en C#

Se da el método Secret1:

```
public static bool Secret1 (int num, int digit )  
{  
    if ( num < 10 )  
        return ( num % 2 == digit % 2 );  
    if ( num % 2 != digit % 2 )  
        return false;  
    return Secret1 ( num / 10, digit );  
}
```

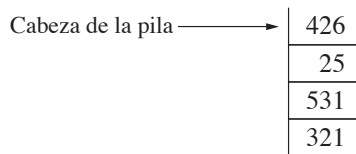
- ⌘. (1) Escribe qué devolverá la convocatoria del método Secret1 (937, 5) . Debes mostrar el seguimiento.
- (2) Ofrece un ejemplo del número num de 3 cifras, para el cual la convocatoria del método Secret1(num,5) devuelva un valor diferente del que se obtuvo en el subapartado ⌘ (1). Debes mostrar el seguimiento.
- (3) Escribe en una oración que es lo que ejecuta el método booleano Secret1 , es decir cuál es la pregunta a la que el método devuelve true o false .

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

Se da el método Secret2:

```
public static bool Secret2 ( Stack <int> s )
{
    bool ok;
    int x;
    if ( s.IsEmpty () )
        ok = true ;
    else
    {
        x = s.Pop () ;
        if ( ! (Secret1 (x, x % 10)) )
            ok = false;
        else
            ok = Secret2 (s);
    }
    return ok ;
}
```

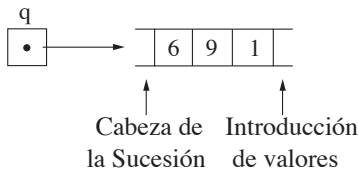
- ב. (1) Para la pila s que se halla a continuación:



Escribe qué devolvera el método Secret2 . Debes mostrar el seguimiento. (No es necesario mostrar el seguimiento del método Secret1).

- (2) Escribe en una línea que es lo que ejecuta el método booleano Secret2 , es decir cuál es la pregunta a la que el método devuelve true o false .

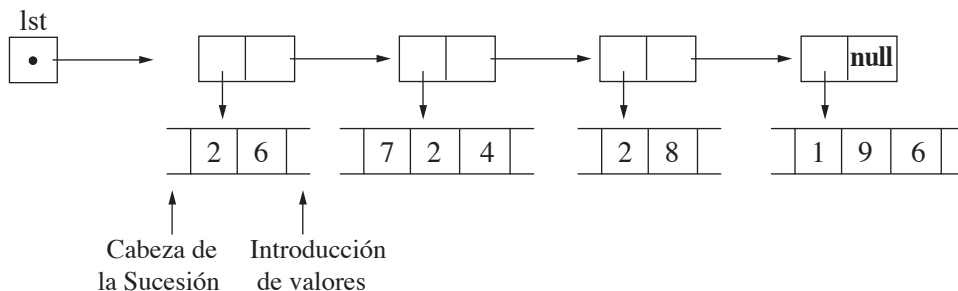
5. א. "Sucesión numérica" es una sucesión de cifras del 1 al 9 (incluido), que representa un número entero — el primer miembro (cabeza de la sucesión) es la cifra de las unidades, el segundo miembro es la cifra de las decenas y así sucesivamente.
- Debes suponer que el número posible de cifras en la sucesión no es mayor que el número de cifras que puede incluir una variable de tipo `int`.
- Por ejemplo: la sucesión que se halla a continuación representa al número 196.



Escribe un método de nombre `toNumber` en Java o `ToNumber` en C#, que reciba una sucesión numérica — `q`, y devuelva el número representado en la sucesión.

Nota: No es necesario conservar la estructura de la sucesión.

- ב. Se da una cadena de eslabones en la que cada eslabón contiene una sucesión numérica. Por ejemplo: la cadena de eslabones que se halla a continuación representa los números, 62, 427, 82, 691.



Escribe un método de nombre `bigNumber` en Java o `BigNumber` en C#, que recibe un indicador `lst` para la cadena de eslabones y devuelve el mayor número representado en la cadena de eslabones.

Para la cadena de eslabones descrita en el ejemplo el método devuelve el número 691.

Es obligatorio utilizar el método que se definió en el apartado א.

6. Se da la clase **Range** que posee dos atributos:

low — número de tipo entero.

high — número de tipo entero.

high es mayor que low .

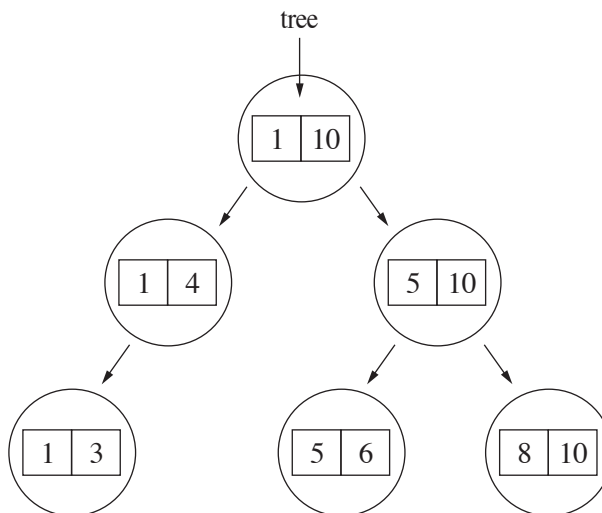
Debes suponer que para cada atributo se definieron en Java, métodos get y set, y en C# métodos Get y Set.

Un **árbol de intervalos** es un árbol en el cual sus miembros son de tipo **Range** .

Un **árbol de intervalos ordenados** es un árbol que o bien es vacío o es un árbol de intervalos tal que para cada nodo se cumplen las condiciones siguientes:

- Si existe un vástago izquierdo, entonces el low del nodo es igual al low del vástago izquierdo, y el high del nodo es mayor o igual al high del vástago izquierdo.
- Si existe un vástago derecho, entonces el high del nodo es igual al high del vástago derecho, y el low del nodo es menor o igual al low del vástago derecho.
- Si existen dos vástagos, el high del vástago izquierdo es menor que el low del vástago derecho.

Ejemplo de un **árbol de intervalos ordenados**:



Escribe un método externo booleano de nombre `order` en Java o `Order` en C# , que reciba un **árbol de intervalos** o un **árbol vacío** y devuelva `true` si el árbol es un **árbol de intervalos ordenado**, en caso contrario el método devolverá `false` .

Tercera parte (25 puntos)

Esta parte consta de preguntas según cuatro orientaciones:

Sistemas de computación y Assembler, págs. 10-11

Introducción a la investigación operativa, págs. 12-15

Modelos computacionales, pág. 16.

Programación orientada a objetos en Java, págs. 17-20.

Programación orientada a objetos en C#, págs. 21-24.

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 cuatro apartados, א-ד, que no tienen relación entre sí. Debes responder a todos.

א. Te presentamos un segmento de programa en Assembler.

```
MOV BX, 100H
```

```
MOV SI, 2
```

```
MOV CX, 3
```

L1:

```
MOV AL, CL
```

```
MOV [BX], AL
```

```
INC BX
```

```
LOOP L1
```

L2:

```
DEC AL
```

```
MOV [BX], AL
```

```
INC BX
```

```
DEC SI
```

```
JNZ L2
```

```
NOP
```

Sigue mediante una tabla de seguimiento la ejecución del segmento de programa. En la tabla de seguimiento debes incluir una columna para cada uno de los acumuladores presentados en el segmento. Además de eso se debe dibujar un mapa de memoria adecuado.

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

/continúa en la página 11/

- ב. A continuación un segmento de programa escrito en lenguaje Java o en C# . Escribe el segmento que corresponda en lenguaje Assembler.

if (a > 0 || b > 0) c ++ ;

Debes suponer que las variables a , b , c se hallan almacenadas en los acumuladores AX, BX , y CX respectivamente.

Debes suponer que los números están orientados (Signed) .

- ג. Se dan dos segmentos Segmento 1 y Segmento 2 .

¿Puede darse el caso de que una vez finalizada la ejecución de cada uno de los dos segmentos los valores en el acumulador CX sean iguales? Justifica tu respuesta.

Segmento 2	Segmento 1
MOV CX, 10	MOV AX, 10
SHL CX, 1	MOV CX, AX
MOV DX, CX	MUL CX
SHL CX, 2	MOV CX, AX
ADD CX, DX	

- ד. En el acumulador AX hay un número cualquiera. Escribe instrucciones que intercambian el bit inferior con el bit superior del acumulador . Por ejemplo: si el valor inicial de AX es 5A9Ch, después del intercambio su valor será 9C5Ah.

No se deben utilizar las instrucciones ROR, ROL.

8. "Número frecuente" en el vector es el número que aparece más veces en el vector.
Por ejemplo en el vector que se halla a continuación:

index:	0	1	2	3	4	5	6	7	8	9
value	-3	6	0	11	8	-9	5	6	33	6

El número frecuente es el 6, pues aparece en el vector 3 veces, y no hay ningún otro que aparezca en el vector más de 3 veces.

- א. Se da el vector ARR de magnitud 100 que contiene números de tipo bit.
Escribe un procedimiento [שגרה] que reciba de una pila un número entero, y almacene en el acumulador AL el número de veces que ese número aparece en el vector ARR.
- ב. Escribe un programa que extraiga el número frecuente del vector ARR , y lo almacena en la variable VAL .
Debes utilizar el procedimiento que escribiste en apartado א.
Debes suponer que en el vector hay un único número frecuente.

Introducción a la Investigación operativa [חקר ביצועים]

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

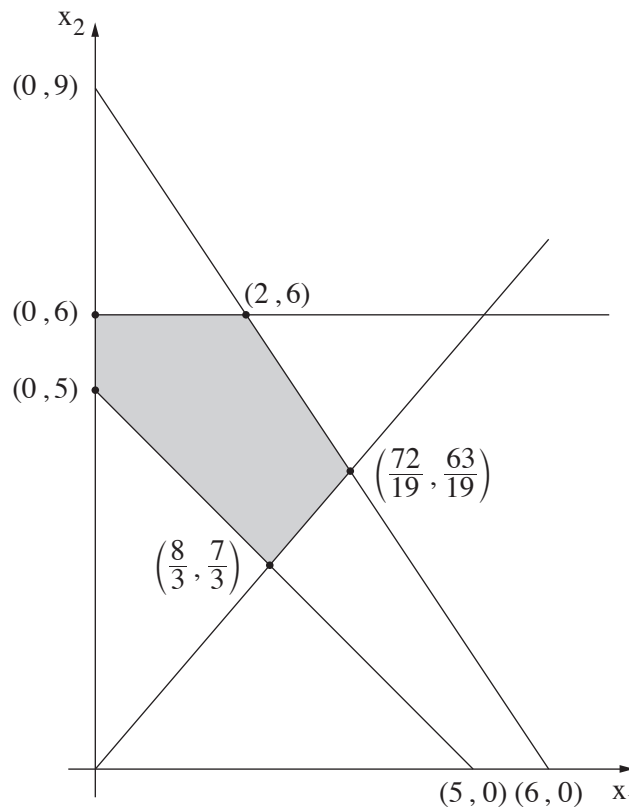
9. Se da un problema de programación lineal:

$$\min \{z = 3x_1 + 5x_2\}$$

Sometido a las restricciones siguientes:

- (1) $x_1 + x_2 \geq 5$
- (2) $x_1 \geq 0$
- (3) $0 \leq x_2 \leq 6$
- (4) $7x_1 \leq 8x_2$
- (5) $3x_1 + 2x_2 \leq 18$

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



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

/continúa en la página 13/

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

Los apartados (κ) - (τ) son independientes entre sí. Responde a todos los apartados.

Se dan cuatro proposiciones i-iv. Para cada uno de los apartados κ - τ que se encuentran a continuación solo una de las proposiciones es verdadera.

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

En cada uno de los apartados κ - τ siguientes determina cuál de las proposiciones i-iv es verdadera.

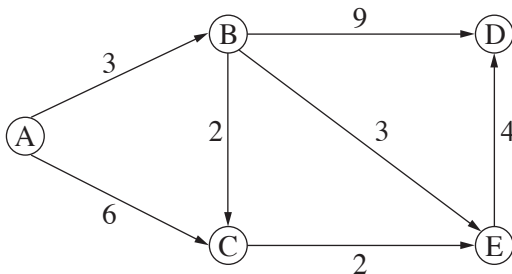
Escribe en tu cuadernillo el número de apartado κ - τ , copia la proposición verdadera en tu cuadernillo y fundamenta tu respuesta.

- 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 escribir la solución óptima general del problema, y el valor de la función objetivo en el intervalo de soluciones óptimas.
- κ ¿Qué proposición es cierta para el problema de programación lineal dado al comienzo de la pregunta? Justifica tu respuesta.
- κ Se cambia solamente la función objetivo del problema dado al comienzo de la pregunta por $\max\{z = 3x_1 + 5x_2\}$.
¿Qué proposición es cierta después del cambio? Justifica tu respuesta.
- λ Se cambia solamente la función objetivo del problema dado al comienzo de la pregunta por $\min\{z = 5x_1 + 5x_2\}$.
¿Qué proposición es cierta después del cambio? Justifica tu respuesta.
- τ Se agrega otra restricción al problema dado al comienzo de la pregunta, y es: $x_1 > 2x_2$.
¿Qué proposición es cierta después de agregar la restricción? Justifica tu respuesta. .

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

Debes responder a ambos.

א. $G = (V, E)$ es un grafo **orientado**:



(1) Dibuja el grafo valiéndote de una matriz de adyacencias.

(2) Halla los recorridos más cortos desde el vértice A a cada uno de los nodos del grafo. Describe cada uno de los recorridos de manera esquemática.

ב. (1) En la tabla que está a continuación se da una solución básica posible obtenida a partir de k iteraciones para un problema de transporte, y se da el valor de u_2 .

Orígenes	Destinos				Oferta	u_i
	A	B	C	D		
1	8 (30)	7	3	2	30	
2	4 (10)	9 (70)	4 (10)	4	90	0
3	3	2	7 (40)	8 (40)	80	
Demanda	40	70	50	40		
v_j						

- Copia la tabla en tu cuadernillo y completa en ella los valores $u_1, u_3, v_1, v_2, v_3, v_4$.
- Explica por qué la solución dada no es la óptima.
- Debes realizar una iteración adicional, o sea la iteración $k + 1$. ¿Cuál es la variable que resulta de la base en esta iteración?
- Traza en tu cuadernillo una nueva tabla, y escribe en ella la solución que se obtendrá después de esta iteración.
- ¿Ha bajado el costo total del problema de transporte?

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

(2) En la tabla que está a continuación se presenta un cierto problema de transporte.

Orígenes	Destinos			Oferta
	A	B	C	
1	10	5	7	10
2	12	8	4	18
3	8	2	5	10
Demanda	16	12	10	

- i. Se cambia la demanda del destino A de 16 a 10 , sin cambiar el resto de las demandas y de las ofertas de la tabla.

Dibuja en tu cuadernillo una nueva tabla que incluya todos los datos de la tabla dada, y posibilite encontrar una solución básica posible, según el método de la esquina noroeste. En la tabla se deben incluir los precios para cada destino.

- ii. Halla la solución básica posible según el método de la esquina noroeste.

Modelos computacionales

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

11. A continuación se hallan los lenguajes L_1 y L_2 sobre el alfabeto $\{1, 0\}$:

$$L_1 = \{0^i 1^{2n} \mid i \geq 1, n = i \bmod 3\}$$

$$L_2 = \{0^i 1^{n+i} \mid i \geq 1, n = i \bmod 3\}$$

- א. Escribe una palabra de longitud 6 que pertenezca al lenguaje L_1 y una palabra de longitud 6 que pertenezca al lenguaje L_2 .
- ב. Si el lenguaje L_1 es regular — construye un autómata finito determinístico no completo que admita el lenguaje, y si el lenguaje no es regular — construye un autómata pila que admita el lenguaje.
- ג. Si el lenguaje L_2 es regular — construye un autómata finito determinístico no completo que admita el lenguaje, y si el lenguaje no es regular — construye un autómata pila que admita el lenguaje.

12. Construye una máquina de Turing que admita como entrada palabras del alfabeto $\{0, 1\}$.

La máquina devolverá la palabra que se obtenga sin ceros, es decir, que la palabra que se devuelva contendrá solamente cifras 1, entre dos signos \$ en un lugar cualquiera de la cinta.

Por ejemplo, si la cinta se ve del siguiente modo:

	1	1	0	0	1	0	1	△	△	
--	---	---	---	---	---	---	---	---	---	------

La salida puede tener, por ejemplo, esta apariencia:

....	△	\$	1	1	1	1	\$	△	
------	---	----	---	---	---	---	----	---	-------

Programación orientada a objetos en Java

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

13. Se dan las clases A, B, Driver.

```
public class A {  
    private int val;  
    public A () {  
        this.val = 1;  
    }  
  
    public A (int val) {  
        this.val = val;  
    }  
  
    public int getVal () {  
        return this.val;  
    }  
  
    public boolean equals (Object other) {  
        System.out.println ("AObject");  
        if (other instanceof A)  
            return (this.val == ((A) other).val);  
        return false;  
    }  
}
```

```
public class B extends A {  
    private String st;  
    public B () {  
        this.st = "B" ;  
    }  
  
    public B (String st, int val) {  
        super (val);  
        this.st = st;  
    }  
  
    public String getSt () {  
        return this.st;  
    }  
  
    public boolean equals (Object other) {  
        System.out.println ("BObject") ;  
        if (other instanceof B)  
            return (this.st.equals (((B) other).st)  
                && this.getVal () == ((B) other).getVal ()) ;  
        return false;  
    }  
  
    public boolean equals(A other) {  
        System.out.println ("BA") ;  
        if (other instanceof B)  
            return (this.st.equals (((B) other).st)  
                && this.getVal () == ((B) other).getVal ()) ;  
        return false;  
    }  
  
    public boolean equals (B other) {  
        System.out.println ("BB") ;  
        return (this.st.equals (other.st)  
            && this.getVal () == other.getVal ()) ;  
    }  
}
```

```
public class Driver {  
    public static void main (String[] args) {  
        A a1 = new A () ;  
        A a2 = new A (5) ;  
        A ab = new B () ;  
        B b1 = new B ("B", 1) ;  
        B b2 = new B ("B", 5) ;  
        // *** //  
    }  
}
```

- א. Dibuja objetos que se formaron en el método `main`. Para cada objeto debes señalar los valores de sus atributos.
- ב. A continuación te presentamos ocho instrucciones 1-8:
- Coloca cada una de las instrucciones en el lugar de `/***/` en el método `main` de la clase `Driver`.
 - Escribe cuál será la salida como consecuencia de la sustitución con cada una de dichas instrucciones.

1. `if (a1.equals (b1)) System.out.println (1) ;`
2. `if (b1.equals (a1)) System.out.println (2) ;`
3. `if (a1.equals (ab)) System.out.println (3) ;`
4. `if (ab.equals (a1)) System.out.println (4) ;`
5. `if (b1.equals (ab)) System.out.println (5) ;`
6. `if (ab.equals (b1)) System.out.println (6) ;`
7. `if (a1.equals (a2)) System.out.println (7) ;`
8. `if (b1.equals (b2)) System.out.println (8) ;`

14. Se dan las clases Driver , First , Second , Third

```
public class First {  
    public static int count = 0 ;  
    private String str ;  
  
    public First (String str) {  
        count ++ ;  
        this.str = str ;  
    }  
  
    public First (First g) {  
        count ++ ;  
        this.str = "Copy" + g.str ;  
    }  
  
    public void setStr (String str) {  
        this.str = str ;  
    }  
  
    public void print () {  
        System.out.println ("First" + this.str) ;  
    }  
}
```

```
public class Second extends First {  
    private First f;  
    public Second (First fx , First fy) {  
        super (fx) ;  
        this.f = fy ;  
    }  
  
    public void setStr (String str) {  
        super.setStr (str) ;  
        this.f = new First (str) ;  
    }  
  
    public void print () {  
        System.out.println ("Second") ;  
        super.print () ;  
        this.f.print () ;  
    }  
}
```

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

```
public class Third {
    private int curr;
    private First[] arr ;
    public Third (int count ) {
        curr = 0 ;
        arr = new First [count] ;
    }

    public void print () {
        for ( int i = 0 ; i < curr ; i++ )
            arr [i].print () ;
    }

    public void add ( First s ) {
        if ( curr < arr.length ) {
            arr [curr] = s ;
            curr ++ ; }
    }
}
```

```
public class Driver {
    public static void main (String[] args) {
        First f1 = new First ("One") ;
        First f2 = new First ("Two") ;
        First f3 = new First (f2) ;
        Second s1 = new Second (f1, f3) ;
        Third t = new Third (First.count) ;
        t.add (f1) ;
        t.add (f2) ;
        t.add (s1) ;
        System.out.println (First.count) ;
        t.print () ;
        System.out.println (" -----") ;
        f1.setStr ("Five") ;
        System.out.println (First.count) ;
        t.print () ;
    }
}
```

- א. Dibuja un diagrama de jerarquía entre las clases First , Second , Third . Se debe señalar la herencia [ירושה] mediante la flecha  y la composición [הכללה] mediante el signo .
- ב. Realiza el seguimiento del método main de la clase Driver:
- (1) Muestra el contenido de todos los objetos que se formaron.
 - (2) Escribe la salida que se obtuvo.

Programación orientada a objetos en C#:

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

15. Se dan las clases A, B, Driver.

```
public class A {  
    private int val;  
  
    public A () {  
        this.val = 1;  
    }  
  
    public A (int val) {  
        this.val = val;  
    }  
  
    public int GetVal () {  
        return this.val;  
    }  
  
    public virtual bool Equals (Object other) {  
        Console.WriteLine ("AObject");  
        if (other is A)  
            return (this.val == ((A) other).val);  
        return false;  
    }  
}
```

```
public class B : A {  
    private string st ;  
  
    public B () {  
        this.st = "B" ;  
    }  
  
    public B (string st, int val)  
        : base (val) {  
        this.st = st ;  
    }  
  
    public string GetSt () {  
        return this.st ;  
    }  
  
    public override bool Equals (Object other) {  
        Console.WriteLine ("BObject");  
        if (other is B)  
            return (this.st.Equals (((B) other).st)  
                && this.GetVal () == ((B) other).GetVal ());  
        return false ;  
    }  
  
    public bool Equals (A other) {  
        Console.WriteLine ("BA");  
        if (other is B)  
            return (this.st.Equals (((B) other).st)  
                && this.GetVal () == ((B) other).GetVal ());  
        return false;  
    }  
  
    public bool Equals (B other) {  
        Console.WriteLine ("BB");  
        return (this.st.Equals (other.st)  
            && this.GetVal () == other.GetVal ());  
    }  
}
```

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

/continúa en la página 22/

```
public class Driver {  
    public static void Main (string[] args) {  
        A a1 = new A ();  
        A a2 = new A (5);  
        A ab = new B ();  
        B b1 = new B ("B", 1);  
        B b2 = new B ("B", 5);  
        // *** //  
    }  
}
```

- א. Dibuja objetos que se formaron en el método Main . Para cada objeto debes señalar los valores de sus atributos.
- ב. A continuación te presentamos ocho instrucciones 1-8:
 - Coloca cada una de las instrucciones en el lugar de `//***//` en el método Main de la clase Driver .
 - Escribe cuál será la salida como consecuencia de la sustitución con cada una de dichas instrucciones.

1. `if (a1.Equals (b1)) Console.WriteLine (1) ;`
2. `if (b1.Equals (a1)) Console.WriteLine (2) ;`
3. `if (a1.Equals (ab)) Console.WriteLine (3) ;`
4. `if (ab.Equals (a1)) Console.WriteLine (4) ;`
5. `if (b1.Equals (ab)) Console.WriteLine (5) ;`
6. `if (ab.Equals (b1)) Console.WriteLine (6) ;`
7. `if (a1.Equals (a2)) Console.WriteLine (7) ;`
8. `if (b1.Equals (b2)) Console.WriteLine (8) ;`

16. Se dan las clases Driver , First , Second , Third

```
public class First
{
    public static int count=0 ;
    private string str ;

    public First (string str) {
        count ++ ;
        this.str = str ;
    }

    public First (First g) {
        count ++ ;
        this.str = "Copy" + g.str ;
    }

    public virtual void SetStr (string str) {
        this.str = str ;
    }

    public virtual void Print () {
        Console.WriteLine ("First" + this.str ) ;
    }
}
```

```
public class Second : First {
    private First f;

    public Second (First fx, First fy)
    : base (fx) {
        this.f = fy ;
    }

    public override void SetStr (string str) {
        base.SetStr (str) ;
        this.f = new First (str) ;
    }

    public override void Print () {
        Console.WriteLine ("Second") ;
        base.Print () ;
        this.f.Print () ;
    }
}
```

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

```
public class Third {
    private int curr;
    private First[] arr ;

    public Third (int count) {
        curr = 0 ;
        arr = new First [count] ;
    }

    public void Print () {
        for (int i = 0 ; i < curr ; i++)
            arr [i].Print () ;
    }

    public void Add (First s) {
        if (curr < arr.Length) {
            arr [curr] = s ;
            curr++ ; }
    }
}
```

```
public class Driver
{
    public static void Main (string[] args) {
        First f1 = new First ("One") ;
        First f2 = new First ("Two") ;
        First f3 = new First (f2) ;
        Second s1 = new Second (f1, f3) ;
        Third t = new Third (First.count) ;
        t.Add (f1) ;
        t.Add (f2) ;
        t.Add (s1) ;
        Console.WriteLine (First.count) ;
        t.Print() ;
        Console.WriteLine (" -----") ;
        f1.SetStr ("Five") ;
        Console.WriteLine (First.count) ;
        t.Print () ;
    }
}
```

- א. Dibuja un diagrama de jerarquía entre las clases First , Second , Third . Se debe señalar la herencia [ירושה] mediante la flecha $\longrightarrow$ y la composición [הכלה] mediante el signo $\blacklozenge$.
- ב. Realiza el seguimiento del método main de la clase Driver:
- (1) Muestra el contenido de todos los objetos que se formaron.
 - (2) Escribe la salida que se obtuvo.

¡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.