

Evaluación final - Escenario 8

Fecha de entrega 18 de oct en 23:55

Puntos 100

Preguntas 20

Disponible 15 de oct en 0:00 - 18 de oct en 23:55

Límite de tiempo 90 minutos

Intentos permitidos 2

Instrucciones



Apreciado estudiante, presenta tus exámenes como **SERGIO EL ELEFANTE, quien con honestidad, usa su sabiduría para mejorar cada día.**

Lee detenidamente las siguientes indicaciones y minimiza inconvenientes:

1. Tienes dos intentos para desarrollar tu evaluación.
2. Si respondiste uno de los intentos sin ningún inconveniente y tuviste problemas con el otro, el examen no será habilitado nuevamente.
3. Cuando estés respondiendo la evaluación, evita abrir páginas diferentes a tu examen. Esto puede ocasionar el cierre del mismo y la pérdida de un intento.
4. Asegúrate de tener buena conexión a internet, cierra cualquier programa que pueda consumir el ancho de banda y no utilices internet móvil.
5. Debes empezar a responder el examen por lo menos dos horas antes del cierre, es decir, máximo a las 9:55 p. m. Si llegada las 11:55 p. m. no lo has enviado, el mismo se cerrará y no podrá ser calificado.
6. El tiempo máximo que tienes para resolver cada evaluación es de 90 minutos.
7. Solo puedes recurrir al segundo intento en caso de un problema tecnológico.
8. Si tu examen incluye preguntas con respuestas abiertas, estas no serán calificadas automáticamente, ya que requieren la revisión del tutor.
9. Si presentas inconvenientes con la presentación del examen, puedes crear un caso explicando la situación y adjuntando siempre imágenes de evidencia, con fecha y hora, para que Soporte Tecnológico pueda brindarte una respuesta lo antes posible.
10. Podrás verificar la solución de tu examen únicamente durante las 24 horas siguientes al cierre.
11. Te recomendamos evitar el uso de teléfonos inteligentes o tabletas para la presentación de tus actividades evaluativas.
12. Al terminar de responder el examen debes dar clic en el botón "Enviar todo y terminar" de otra forma el examen permanecerá abierto.

¡Confiamos en que sigas, paso a paso, en el camino hacia la excelencia académica!
¿Das tu palabra de que realizarás esta actividad asumiendo de corazón nuestro

PACTO DE HONOR?



Historial de intentos

	Intento	Hora	Puntaje
MANTENER	Intento 2	64 minutos	90 de 100
MÁS RECIENTE	Intento 2	64 minutos	90 de 100
	Intento 1	72 minutos	55 de 100

❗ Las respuestas correctas ya no están disponibles.

Puntaje para este intento: **90** de 100

Entregado el 17 de oct en 22:19

Este intento tuvo una duración de 64 minutos.

Incorrecto

Pregunta 1

0 / 5 pts

¿ Como se le llama la relación que define la solución de la situación particular y se basa en llamadas a la misma función?

- ☒ Relación de concurrencia
- ☐ Relación de recurrencia
- ☐ Relación directa
- ☐ Relación redundante

Pregunta 2

5 / 5 pts

Se dice que Leonardo de Pisa, más conocido como Fibonacci, descubrió la sucesión que lleva su nombre estudiando cómo crecía una población de conejos. El crecimiento de la población seguía las siguientes reglas:

- La población inicia con una pareja de conejos jóvenes.
- Una pareja de conejos jóvenes madura al cumplir un mes.
- Una pareja de conejos concibe una nueva pareja de conejos jóvenes cada mes, iniciando un mes después de su maduración.

Así, en los meses 1 y 2 se tendría únicamente una pareja. Para el mes 3, habría ya una nueva pareja, para un total de 2. En el mes 4, la pareja ya madura tendría una nueva pareja, completando 3 parejas. En el mes 5, la primera pareja y su primera pareja cría engendrarían dos nuevas parejas, siendo la población total 5 parejas.

El número de parejas que conforman la población en el mes n puede calcularse mediante la ecuación de recurrencia:

$$parejas(n) = \begin{cases} 1 & , n = 1 \text{ ó } n = 2 \\ parejas(n-1) + parejas(n-2), & n > 2 \end{cases}$$

De los siguientes, un método que no evalúa correctamente la ecuación de recurrencia presentada es:

☐

```
static long parejas(int n){  
    if (n==1 || n==2) return 1;  
    return parejas(n-1) + parejas(n-2);  
}
```

☐

```
static long parejas(int n){  
    return (n==1 || n==2)?1:parejas(n-1) + parejas(n-2);  
}
```

☒

```
static long parejas(int n){  
    if (n==1 || n==2) return 1;  
    if (n>2) return parejas(n-1) + parejas(n-2);  
}
```

La implementación

```
static long parejas(int n){  
    if (n==1 || n==2) return 1;  
    if (n>2) return parejas(n-1) + parejas(n-2);  
}
```

Generará un error de compilación ya que el compilador no sabrá que retornar cuando no se satisfaga ninguna de las dos condiciones (note que el compilador no tiene por qué saber que n nunca será menor a 1).

Las implementaciones restantes son equivalentes y correctas.

☐

```
static long parejas(int n){  
    if (n>2) return parejas(n-1) + parejas(n-2);  
    return 1;  
}
```

Pregunta 3

5 / 5 pts

El método recursivo

```
int metodo(int n)
```

```
{
```

```
if(n == 0) return 1;
```

```
else return 2 * metodo(n - 1);
```

```
}
```

Permite:

☐ Calcular $2n$

☐ Calcular 2^n

- ☐ Calcular la suma de los número pares menores que n
- ☐ Calcular el factorial de 2
- ☒ Ninguna de las anteriores

Pregunta 4**5 / 5 pts**

Los algoritmos recursivos son una alternativa al uso de estructuras iterativas.

- ☒ Verdadero
- ☐ Falso

Pregunta 5**5 / 5 pts**

David quiere desarrollar un programa de edición de código java que le permita hacer su código más legible y calcule algunos datos interesantes. Dado que el procesamiento de información textual no es su fuerte, David le ha pedido su ayuda para codificar algunas de las tareas básicas del editor.

David cree que si representa el texto como un arreglo de valores *char* en lugar de como un *String*, su programa será más eficiente. La instrucción que le permite convertir el código representado como *String* en un arreglo de *chars* es:

- ☐ `char[] cod_arr = (char[]) cod;`

☐ `char[] cod_arr = cod.toArray();`

☒ `char[] cod_arr = cod.toCharArray();`

No es posible hacer una conversión directa de String a char[], por lo que las instrucciones

`char[] cod_arr = cod;` y `char[] cod_arr = (char[]) cod;` generarán errores de compilación.

El método de la clase String que permite convertir una cadena a arreglo de chars es toCharArray.

☐ `char[] cod_arr = cod;`

Pregunta 6

5 / 5 pts

Cuál es la salida del siguiente programa en JAVA?

```
String a = "hola";
String b = 'h' + "o" + 'l' + "a" + "";
boolean c = a.equals (b);
boolean e = !c;
char f = 'f';
char g = 'f';
boolean h = g != f;
System.out.println (a);
System.out.println (b);
System.out.println (c);
System.out.println (e);
System.out.println (f);
System.out.println (g);
System.out.println (h );
```

☐

```
hola_  
hola  
false  
false  
g  
g  
false
```

☒

```
hola  
hola  
true  
false  
f  
f  
false
```

☐

```
hola  
hola  
false  
false  
f  
g  
false
```

☐

```
hola  
hola_  
false  
true  
g  
f  
false
```

Pregunta 7**5 / 5 pts**

Cuál es la manera correcta para convertir mi arreglo en String e imprimirlo en pantalla ?

☐

System.out.println(Arreglo);

☒

System.out.println(Arrays.toString(arreglo));

☐

System.out.println("Su arreglo es");

☐ `System.out.println("Su arreglo fue"+Arreglo);`

Pregunta 8**5 / 5 pts**

David quiere desarrollar un programa de edición de código java que le permita hacer su código más legible y calcule algunos datos interesantes. Dado que el procesamiento de información textual no es su fuerte, David le ha pedido su ayuda para codificar algunas de las tareas básicas del editor.

Una de las funcionalidades con que David desea que cuente su editor es la de decidir si un nombre de variable es válido o no. Recuerde que un nombre de variable válido puede contener únicamente letras, dígitos, guiones bajos (_) y signos de dólar (\$). Además, el primer símbolo del nombre no puede ser un dígito.

La expresión que permite verificar si el primer símbolo del nombre es un dígito es:

☒ `length() > 0 && nombre.charAt(0)>='0' && nombre.charAt(0)<='9'`

Para verificar que un carácter corresponda a un dígito basta con preguntar si su código se encuentra entre el código del carácter '0' (48) y el código del carácter '9' (57), inclusive.

☐ `length() > 0 && nombre.charAt(0)>='9' && nombre.charAt(0)<='0'`

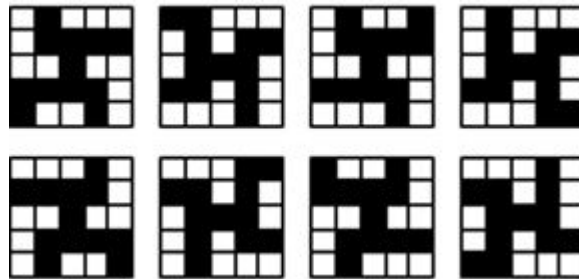
☐ `length() > 0 && nombre.charAt(0)>=0 && nombre.charAt(0)<=9`

☐ `nombre.length() > 0 && nombre.charAt(0)>=9 && nombre.charAt(0)<=0`

Pregunta 9**5 / 5 pts**

Se desea implementar el sistema de visión de un robot clasificador de piezas. El robot ve a través de una cámara que transforma la imagen de cada pieza en una matriz de 5 filas y 5 columnas. Cada celda de la matriz contiene un 1 o un 0, de acuerdo a si el color en dicho píxel es negro o blanco, respectivamente.

Una pieza puede ser rotada 90, 180 o 270 grados. Además, puede volcarse. Como ejemplo de lo anterior, las siguientes imágenes corresponden a un mismo tipo de pieza:



Su tarea es implementar un programa que reciba las matrices de dos imágenes e indique si las imágenes corresponden a piezas del mismo tipo.

Como parte del programa que resuelve el problema enunciado, se debe implementar un método que tome como entradas dos matrices correspondientes a dos imágenes y retorne un valor booleano indicando si las imágenes son idénticas (sin rotaciones ni reflejo).

Para esto, se tiene el siguiente código:

```
static boolean son_iguales(byte[][] img1, byte[][] img2){
    for (int i = 0; i < img1.length; i++)
        if (<condicion>) return false;
    return true;<condicion>
}
```

Para que el método funcione correctamente, <condicion> debe reemplazarse por:

☐ `img1[i] != img2[i]`

☐ `Arrays.not_equals(img1[i], img2[i])`

☒ `!Arrays.equals(img1[i], img2[i])`

Al comparar dos arreglos (o matrices) usando el operador `!=` se está interrogando sobre si los arreglos (o matrices) son el mismo, mas no sobre sus contenidos. Así, dos arreglos distintos pero de idéntico contenido darían como resultado verdadero, lo cual es incorrecto.

El método `Arrays.equals` permite comparar dos arreglos por contenido. Para preguntar si sus contenidos son diferentes basta negar el resultado de este método.

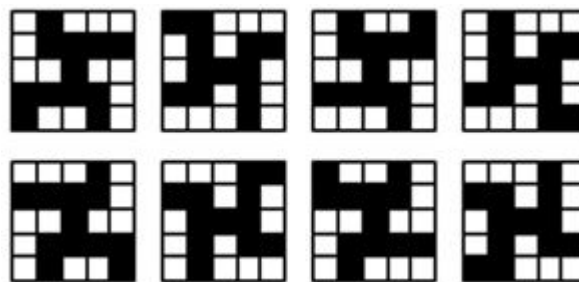
El método `Arrays.not_equals` no existe.

☐ `img1 != img2`

Pregunta 10

5 / 5 pts

Se desea implementar el sistema de visión de un robot clasificador de piezas. El robot ve a través de una cámara que transforma la imagen de cada pieza en una matriz de 5 filas y 5 columnas. Cada celda de la matriz contiene un 1 o un 0, de acuerdo a si el color en dicho píxel es negro o blanco, respectivamente.



Una pieza puede ser rotada 90, 180 o 270 grados. Además, puede volcarse. Como ejemplo de lo anterior, las siguientes imágenes corresponden a un mismo tipo de pieza:

Su tarea es implementar un programa que reciba las matrices de dos imágenes e indique si las imágenes corresponden a piezas del mismo tipo.

Como solución definitiva al problema enunciado, se propuso el siguiente método:

```
static boolean comparar_imagenes(byte[][] img1, byte[][] img2){
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 4; j++) {
            if (son_iguales(img1, img2)) return true;
            img2 = rotar_imagen(img2);
        }
        img2 = reflejar(img2);
    }
    return false;
}
```

- La función `rotar_imagen` retorna una copia de la imagen de entrada rotada 90 grados en sentido contrario a las manecillas del reloj.
- La función `reflejar` retorna una copia de la imagen de entrada invirtiendo el orden de sus columnas.
- La función `son_iguales` retorna un valor booleano indicando si las dos matrices son idénticas o no.

Sobre el método `comparar_imágenes` es posible afirmar que:

☐

No resuelve el problema ya que *rotar_imagen* debería rotar la imagen en el sentido de las manecillas del reloj, y no en sentido contrario.

☐

No resuelve el problema ya que solamente se rota una de las imágenes, no las dos.

☒

Resuelve el problema de manera correcta.

El método toma como referencia la imagen `img1` y modifica la imagen `img2` intentando hacerlas idénticas. La rotación de `img2` ocurre 4 veces, probando todas las rotaciones posibles. Esto hace que sea indiferente si se rota en el sentido de las manecillas del reloj o en sentido contrario. Adicionalmente, la imagen es reflejada y rotada 4 veces nuevamente, por lo que todas las distintas versiones de una misma pieza son verificadas.

En conclusión, el método es correcto.



No resuelve el problema ya que solamente se refleja una de las imágenes, no las dos.

Pregunta 11

5 / 5 pts

Las calles de la ciudad de Bacatá forman una cuadrícula perfecta, como se muestra en la figura 1.

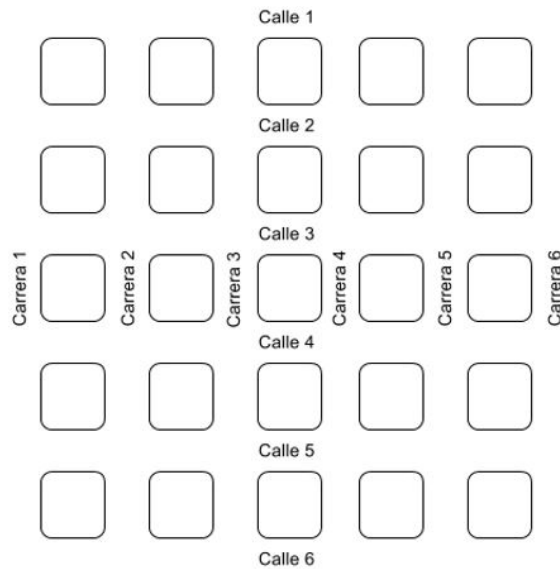


Figura 1. Área de la ciudad de Bacatá entre las calles 1 a 6 y las carreras 1 a 6

Fuente: Elaboración propia

Las vías horizontales son llamadas calles y están enumeradas iniciando en 1. Las vías verticales son llamadas carreras y están numeradas de igual forma.

El alcalde de Bacatá ha decidido modificar la forma en que las tarifas de los taxis son calculadas, considerando la geometría de la ciudad. Las nuevas tarifas se calcularán de acuerdo con la calle y carrera de la intersección donde se toma el taxi, así como a la calle y carrera donde finaliza el viaje. Específicamente, el costo de un viaje se calcula como:

$$\text{Costo} = \$100 \times \left(\left| \text{calle}_{\text{origen}} - \text{calle}_{\text{destino}} \right| + \left| \text{carrera}_{\text{origen}} - \text{carrera}_{\text{destino}} \right| \right)$$

donde $|x|$ representa el valor absoluto de x .

El alcalde le ha pedido personalmente a usted que construya la aplicación que calculará las nuevas tarifas y reemplazará a los taxímetros actuales. La aplicación debe solicitar la información de los puntos de inicio y finalización del viaje, e informar al usuario cuánto deberá pagar al llegar a su destino.

En el problema enunciado, los valores de salida (que definen la solución de la instancia) son:

☐ El número total de calles y carreras que recorre el taxi durante el viaje.

☐ Los números de la calle y la carrera del punto de origen, y los números de la calle y la carrera del punto de destino.

☒ El valor que debe pagar el usuario al finalizar el viaje.

Las salidas corresponden a los valores finales que debe calcular y reportar el programa al usuario. De acuerdo con el enunciado del problema, la salida sería entonces el valor que debe pagar el usuario al finalizar el viaje.

Los números de las calles y carreras de los puntos origen y destino son proporcionados por el usuario. Por tanto, corresponden a las entradas del problema.

Las distancias entre las calles y las carreras de los puntos origen y destino, así como el número total de calles recorridas, son datos que debe calcular el programa pero no son relevantes para el usuario; son datos que usa el programa para calcular las salidas.

☐ Las distancias entre las calles y las carreras de los puntos origen y destino.

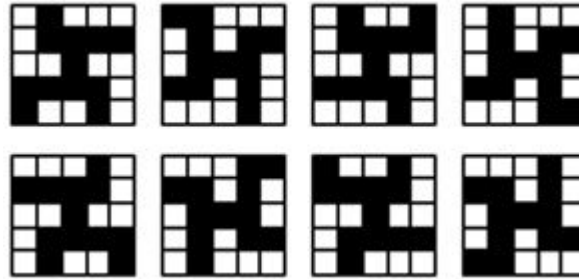
Pregunta 12

5 / 5 pts

Se desea implementar el sistema de visión de un robot clasificador de piezas. El robot ve a través de una cámara que transforma la imagen de cada pieza en una matriz de 5 filas y 5 columnas. Cada celda de la matriz

contiene un 1 o un 0, de acuerdo a si el color en dicho píxel es negro o blanco, respectivamente.

Una pieza puede ser rotada 90, 180 o 270 grados. Además, puede volcarse. Como ejemplo de lo anterior, las siguientes imágenes corresponden a un mismo tipo de pieza:



Su tarea es implementar un programa que reciba las matrices de dos imágenes e indique si las imágenes corresponden a piezas del mismo tipo.

Antes de procesar una imagen, debe verificarse que ésta contenga únicamente los valores 0 y 1. Se desea un método que retorne verdadero si la imagen es correcta o falso si contiene colores no aceptados (valores diferentes a 0 ó 1).

El método que hace correctamente esta verificación es:

☐

```
static boolean verificar_colores(byte[][] img){
    for (int i = 0; i < 5; i++)
        for (int j = 0; j < 5; j++)
            if (img[i][j]==1 &&
img[i][j]!=0) return true;
    return false;
}
```

☒

```
static boolean verificar_colores(byte[][] img){
    for (int i = 0; i < 5; i++)
        for (int j = 0; j < 5; j++)
            if (img[i][j]!=1 &&
img[i][j]!=0) return false;
    return true;
}
```

Debe verificarse para cada elemento de la matriz si es 1 o 0. En caso contrario, sabremos que la condición verificada no se cumple. Solamente es posible afirmar que la condición se cumple cuando se ha recorrido en su totalidad la matriz.

Las opciones *if (img[i][j]==1 || img[i][j]==0) return true;* y *if (img[i][j]==1 && img[i][j]==0) return true;* terminan el método prematuramente sin verificar toda la matriz.

En *if (img[i][j]!=1 || img[i][j]!=0) return false;*, la condición siempre se satisface ya que la posición no puede contener simultáneamente un 1 y un 0. Esto hace que el método siempre retorne falso, independientemente del contenido de la matriz.

La opción correcta es *if (img[i][j]!=1 && img[i][j]!=0) return false;*, ya que solamente afirma que la condición verificada no es satisfecha cuando se encuentra una posición que no contiene un 1 y tampoco contiene un 0.



```
static boolean verificar_colores(byte[][] img){
    for (int i = 0; i < 5; i++)
        for (int j = 0; j < 5; j++)
            if (img[i][j]!=1 ||
img[i][j]!=0) return false;
    return true;
}
```



```
static boolean verificar_colores(byte[][] img){
    for (int i = 0; i < 5; i++)
        for (int j = 0; j < 5; j++)
            if (img[i][j]==1 ||
img[i][j]==0) return true;
    return false;
}
```

Pregunta 13

5 / 5 pts

Qué hace el siguiente fragmento de código?

```
boolean respuesta = false;
for (int i = 0; i < s.length (); i++) {
    if (c == s.charAt (i))
        respuesta = true;
}
System.out.println (respuesta);
```

☐

El programa busca una letra **s** en una palabra **c**, cambia el valor de la variable **respuesta** a **true** al encontrar la letra, de lo contrario la deja en **false**.

☐

El programa revisa si una letra **c** se encuentra en una palabra **s** dejando **true** en la variable **respuesta** si la encuentra **i** veces.

☐

El programa recorre una palabra contando el número de coincidencias de una letra **c** en una palabra **s**.

☒

El programa busca una letra **c** en una palabra **s**, cambia el valor de la variable **respuesta** a **true** al encontrar la letra, de lo contrario la deja en **false**.

☐

El programa busca una letra **s** en una palabra **c**, cambia el valor de la variable **respuesta** a **false** al encontrar la letra, de lo contrario la deja en **true**.

☐

El programa busca una letra **c** en una palabra **s**, cambia el valor de la variable **respuesta** a **false** al encontrar la letra, de lo contrario la deja en **true**.

Pregunta 14

5 / 5 pts

Si las siguientes instrucciones contenidas en un programa en lenguaje JAVA fueran ejecutadas, definiendo previamente los valores:


```
double monto=120;
```

```
int meses = 12;
```

```
double interes=0.1;
```

¿Qué valor se imprimiría en pantalla?

```
for ( int n = 1; n <= meses; n++){
```

```
    monto = monto + monto * interes / 100;
```

```
    System.out.println (" Interes del mes"+n+" es de: "+monto);
```

```
}
```

Interes del mes1 es de: 10.01

Interes del mes2 es de: 10.02001

Interes del mes3 es de: 10.030030009999999

Interes del mes4 es de: 10.04006004001

Interes del mes5 es de: 10.05010010005001

Interes del mes6 es de: 10.06015020015006

Interes del mes7 es de: 10.07021035035021

Interes del mes8 es de: 10.08028056070056

Interes del mes9 es de: 10.09036084126126

Interes del mes10 es de: 10.100451202102521

Interes del mes11 es de: 10.110551653304624

☐ Interes del mes12 es de: 10.12066220495793

Interes del mes1 es de: 110.11

Interes del mes2 es de: 110.22011

Interes del mes3 es de: 110.33033011

Interes del mes4 es de: 110.44066044011001

Interes del mes5 es de: 110.55110110055011

Interes del mes6 es de: 110.66165220165067

Interes del mes7 es de: 110.77231385385232

Interes del mes8 es de: 110.88308616770618

Interes del mes9 es de: 110.99396925387389

Interes del mes10 es de: 111.10496322312775

Interes del mes11 es de: 111.21606818635088

☐ Interes del mes12 es de: 111.32728425453723

Interes del mes1 es de: 120.12
Interes del mes2 es de: 120.24012
Interes del mes3 es de: 120.36036012000001
Interes del mes4 es de: 120.48072048012001
Interes del mes5 es de: 120.60120120060013
Interes del mes6 es de: 120.72180240180073
Interes del mes7 es de: 120.84252420420253
Interes del mes8 es de: 120.96336672840674
Interes del mes9 es de: 121.08433009513514
Interes del mes10 es de: 121.20541442523027
Interes del mes11 es de: 121.3266198396555
☒ Interes del mes12 es de: 121.44794645949516

Interes del mes1 es de: 90.09
Interes del mes2 es de: 90.18009
Interes del mes3 es de: 90.27027009000001
Interes del mes4 es de: 90.36054036009001
Interes del mes5 es de: 90.45090090045011
Interes del mes6 es de: 90.54135180135056
Interes del mes7 es de: 90.6318931531519
Interes del mes8 es de: 90.72252504630507
Interes del mes9 es de: 90.81324757135137
Interes del mes10 es de: 90.90406081892272
Interes del mes11 es de: 90.99496487974164
☐ Interes del mes12 es de: 91.08595984462139

Como podemos comprobar ejecutándolo en el programa Eclipse.

Pregunta 15

5 / 5 pts

Ejercicio de Compilación:

1. Ingrese a eclipse.
2. Cree un proyecto.
3. Cree una clase.
4. Realize el encabezado public static void main (String[] args){
5. Declare variables (DE SER NECESARIO)

¿Cuál es la salida del siguiente programa en JAVA?

```
for (int i=10;i>=0;i--){
```

```
System.out.println (i);
```

```
}
```

☐

```
1.0  
9.0  
1.0  
2  
3  
-1
```

☒

```
10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
0
```

☐

```
Error de compilación
```

☐

```
20  
19  
18  
17  
16  
15  
14  
13  
12  
11  
10
```

Pregunta 16**5 / 5 pts**

Si las siguientes instrucciones contenidas en un programa en lenguaje JAVA fueran ejecutadas, ¿Qué valor se imprimiría en pantalla?

```
int i=0;

while(i<=10) {

System.out.println(i);

i++;

}
```

- ☐ No pasa nada porque la condición no se cumple.
- ☒ Los números del 0 al 10
- ☐ Los números del 1 al 10
- ☐ Ninguna de las anteriores

Recordemos que el ciclo while, se cumplirá siempre que sea verdadero.

Incorrecto**Pregunta 17****0 / 5 pts**

Se dice que dos números naturales son *primos relativos* si no tienen ningún factor primo en común o, dicho de otra manera, si y sólo si su máximo común divisor es igual a 1. Dados dos números naturales **a** y **b**, cuál de los siguientes fragmentos de código dice correctamente si son o no primos relativos?

☒

```
int i = 2;
boolean resultado = true;
while (i < a) {
    if (a % i != 0 && b % i != 0)
        resultado = false;
    i = i + 1;
}
```

☐

```
int c = a;
if (a > b)
    c = b;
int i = 2;
boolean resultado;
while (i < c) {
    if (a % i == 0 && b % i == 0)
        resultado = false;
    else
        resultado = true;
    i = i + 1;
}
```

☐

```
int c = a;
if (a > b)
    c = b;
int i = 2;
boolean resultado = true;
while (resultado && i < c) {
    if (a % i == 0 || b % i == 0)
        resultado = false;
    i = i + 1;
}
```

☐

```
int c = a;
if (a > b)
    c = b;
int i = 2;
boolean resultado = true;
while (resultado && i < c) {
    if (a % i == 0 && b % i == 0)
        resultado = false;
    i = i + 1;
}
```

Pregunta 18**5 / 5 pts**

A continuación, encontrará el enunciado de un problema matemático simple. Entre las opciones que se le presentarán posteriormente,

seleccione aquella que corresponde al encabezado de un método, en lenguaje Java, que permitiría resolver correctamente el problema.

Problema: Dadas las coordenadas (x_1, y_1) del punto P_1 sobre el plano cartesiano y las coordenadas (x_2, y_2) del punto P_2 sobre el plano cartesiano, calcule la pendiente de la recta que pasa por los puntos P_1 y P_2 .

Nota: preste atención al hecho de que solamente se le pide que identifique el encabezado apropiado, no se está hablando de las instrucciones del método.

☐

```
static double calcular_pendiente(double P1, double x1, double y1, double P2, double x2, double y2) { }
```

☐

```
static int calcular_pendiente(double P1, double x1, double y1, double P2, double x2, double y2) { }
```

☐

```
static int calcular_pendiente(double P1, x1, y1, P2, x2, y2) { }
```

☒

```
static double calcular_pendiente(double x1, double y1, double x2, double y2) { }
```

☐

```
static int calcular_pendiente(double x1, y1, x2, y2) { }
```

Pregunta 19

5 / 5 pts

Ejercicio de Compilación:

1. Ingrese a eclipse.
2. Cree un proyecto.
3. Cree una clase.
4. Realize el encabezado `public static void main (String[] args){`

5. Declare variables (DE SER NECESARIO)

¿Cuál es la salida del siguiente programa en JAVA?

```
String a = "hola";  
String b = 'h' + "o" + 'l' + "a" + "";  
boolean c = a.equals (b);  
boolean e = !c;  
char f = 'f';  
char g = 'f';  
boolean h = g != f;  
System.out.println (a);  
System.out.println (b);  
System.out.println (c);  
System.out.println (e);  
System.out.println (f);  
System.out.println (g);  
System.out.println (h );
```

☐

```
hola  
hola_  
false  
true  
g  
f  
false
```

☐

```
hola_  
hola  
false  
false  
g  
g  
false
```

☐

```
hola  
hola  
false  
false  
f  
g  
false
```

☒

```
hola  
hola  
true  
false  
f  
f  
false
```

Pregunta 20**5 / 5 pts**

Los tipos de datos reales, se clasifican en: (Seleccione 2 respuestas)

☒ float☒ double☐ char☐ boolean

Puntaje del examen: **90** de 100

✕