

Nombre: _____ **Matrícula:** _____

1. La función *bres* mostrada en la figura adjunta, implementa el algoritmo de trazado de una línea *Bresenham*.

```
1  bres
2      int x, y, dx, dy, p
3      input (int x1, y1, x2, y2)
4      dx = x2 - x1
5      dy = y2 - y1
6      p = 2*dx - dy
7      display (x, y)
8      while x<x2
9          if p<0
10             p = p+2*dy
11             x = x+1
12          else
13             p = p+2*(dy-dx)
14             x = x+1
15             y = y+1
16          endif
17          display (x, y)
18      endwhile
19  end bres
```

- Elabore el *Data Flow Graph* (DFG) -grafo de flujo de datos-para la función ***bres*** [8%]
 - Defina du-pair. Identifique todos los du-pairs para las variables ***x1***, ***x2*** y ***y*** en la función ***bres*** especificando su uso (***p- o c-***) [8%]
 - Defina el tipo de cobertura 100% All-definitions. Usando esta definición, especifique los casos de prueba para la variable ***p***. Explique los principios que usted utilice para generar estos casos de prueba [8%]
 - Escriba los pares secuencia-tiempo para las variables ***x1***, ***x2*** y ***y***. Basado en estos, ¿a qué conclusiones puede usted llegar? [8%]
2. Considere una máquina dispensadora en base a monedas que prepara y vende vasos de café cargado o ligero, con o sin leche. Para comprar una bebida, el cliente debe insertar monedas y seleccionar la bebida. La máquina contiene dispensadores para monedas, café y leche; y los usa para preparar la bebida que el cliente haya seleccionado. Todas las bebidas tienen el mismo precio y la máquina da el cambio. La máquina valida el monto de monedas insertadas. Cuando suficientes monedas han sido insertadas, la bebida seleccionada será preparada y, de ser

necesario, el cambio será entregado. Asuma que la máquina sirve un tamaño de bebida y que el stock es infinito.

- a. Usando el modelo de componentes de UML 2.0, identifique los componentes apropiados para el sistema. Para cada componente, describa brevemente qué hace y liste los servicios que provee y requiere. **[6%]**
 - b. Componga los componentes identificados para implementar el sistema y especifique la interfaz del sistema. **[12%]**
 - c. Describa en detalle cómo se lleva a cabo la lógica de control del sistema. Utilice un diagrama de secuencia. **[5%]**
3. Muestre las expresiones *XPath* que resuelvan las siguientes consultas en el archivo en XML a continuación: **[20%]**
- a. Direcciones de todos los restaurantes
 - b. Nombres de todos los restaurantes
 - c. Dirección del restaurante *Kalpna*
 - d. Números de teléfono de todos los restaurantes que sirven comida india
 - e. Nombre de todos los restaurantes que sirven comida italiana

```
▼<restaurants>
  ▼<restaurant name="The Olive Tree">
    <address>27 Marshall Street, Edinburgh, EH8 9BJ</address>
    <cuisine>Mediterranean</cuisine>
    <phoneno>0131 667 9997</phoneno>
  </restaurant>
  ▼<restaurant name="Nawroz">
    <address>26 Potterow, Edinburgh, EH8 9BT</address>
    <cuisine>Kurdish</cuisine>
    <phoneno>0131 667 2299</phoneno>
  </restaurant>
  ▼<restaurant name="Kebab Mahal">
    <address>7 Nicolson Square, Edinburgh, EH8 9BH</address>
    <cuisine>Indian</cuisine>
    <phoneno>0131 622 7228</phoneno>
  </restaurant>
  ▼<restaurant name="Sylvesters">
    <address>55 West Nicolson Street, Edinburgh, EH8 9DB</address>
    <cuisine>Scottish</cuisine>
    <phoneno>0131 662 4493</phoneno>
  </restaurant>
  ▼<restaurant name="Buffalo Grill">
    <address>12 Chapel Street, Edinburgh, EH8 9AY</address>
    <cuisine>North American</cuisine>
    <phoneno>0131 667 7427</phoneno>
  </restaurant>
  ▼<restaurant name="Nile Valley">
    <address>6 Chapel Street, Edinburgh, EH8 9AY</address>
    <cuisine>Cafe</cuisine>
    <phoneno>0131 667 8200</phoneno>
  </restaurant>
  ▼<restaurant name="Kalpna">
    <address>2 St Patrick's Square, Edinburgh, EH8 9EZ</address>
    <cuisine>Indian</cuisine>
    <phoneno>0131 667 9890</phoneno>
  </restaurant>
  ▼<restaurant name="Suruchi">
    <address>14A Nicholson Street</address>
    <cuisine>Indian</cuisine>
    <phoneno>0131 556 6583</phoneno>
  </restaurant>
  ▼<restaurant name="Ciao Roma">
    <address>64 South Bridge</address>
    <cuisine>Italian</cuisine>
    <phoneno>0131 557 3777</phoneno>
  </restaurant>
  ▼<restaurant name="Ti Amo">
    <address>16 Nicolson Street, Old Town, Edinburgh, EH8 9DH</address>
    <cuisine>Italian</cuisine>
    <phoneno>0131 556 5678</phoneno>
  </restaurant>
</restaurants>
```

4. Para cada uno de los fragmentos de código fuente determine si usted puede aplicar *refactoring*. En caso de ser posible, nombre la técnica de *refactoring* que usted usará y muestre el resultado luego de aplicarla. **[25%]**

a.

```
public class Employee
{
    private String name;
    private String jobTitle;
}

public class Student
{
    private String name;
    private Course course;
}
```

b.

```
int withdraw(int amount)
{
    if (amount > balance)
        return -1;
    else {
        balance -= amount;
        return 0;
    }
}
```

c.

```
void setValue (String name, int value) {
    if (name.equals("height")) {
        height = value;
        return;
    }
    if (name.equals("width")) {
        width = value;
        return;
    }
    Assert.shouldNeverReachHere();
}
```

d.

```
double getValueForPeriod (int periodNumber)
{
    try
    {
        return values[periodNumber];
    }
    catch (ArrayIndexOutOfBoundsException e)
    {
        return 0;
    }
}
```

e.

```
public abstract class Party { }

public class Person extends Party
{
    private String firstName;
    private String lastName;
    private Date dob;
    private String nationality;
    public void printNameAndDetails()
    {
        System.out.println("Name: " + firstName + " " + lastName);
        System.out.println("DOB: " + dob.toString() + ", Nationality: " + nationality);
    }
}

public class Company extends Party
{
    private String name;
    private String companyType;
    private Date incorporated;
    public void PrintNameAndDetails()
    {
        System.out.println("Name: " + name + " " + companyType);
        System.out.println("Incorporated: " + incorporated.toString());
    }
}
```