1. **What is the output after the program is run?**

```
public class Exam01 {
    public static void main(String[] args) {
        int x = 5;
        while (x > 0) {
            System.out.print(x);
            x = x - 1;
        }
    }
}
```

answer

[redacted] 54321

2. **What is the output after the program is run?**

```
public class Exam02 {
    public static void main(String[] args) {
        for (int x = 5; x > 0; x--) {
            System.out.print(x);
        }
    }
}
```

answer

[redacted] 54321

3. **What is the output after the program is run?**

```
public class Exam03 {
    public static void main(String[] args) {
        int sum = 0, num = 0;
        do {
            num++;
            sum = sum + num;
        } while (num <= 5);
        System.out.println(sum);
    }
}
```

num sum
1 → 1
2 → 3
3 → 6
4 → 10
5 → 15
6 → 21

answer

[redacted] 21

4. **What is the output after the program is run?**

```
public class Exam04 {
    public static void main(String[] args) {
        int sum = 0;
        for (int i = 0; i <= 5; i++) {
            sum = sum + 2 * i;
        }
        System.out.println(sum);
    }
}
```

answer

[redacted] 30

5. Fill in the blank using the given choices to complete the balance calculation with the given interest rate for 12 months.

```
public class Exam05 {
    public static void main(String[] args) {
        double interest = 0.07;
        double balance = 5000;
        int months = 12;
        for (                                ) {
            balance += balance * interest / 12;
        }
        System.out.println(balance);
    }
}
```

answer

```
int i = 0; i < months; i++    หรือรูปแบบอื่นใดที่ทำให้ทำซ้ำ 12 รอบ
```

6. What is the output after the program is run?

```
public class Exam06 {
    public static void main(String[] args) {
        int scoreA = 30, scoreB = 45;
        char candidateA = 'D', candidateB = 'D', candidateC = 'D';
        if (scoreA > 20) {
            candidateA = 'B';
        } else if (scoreB > 30) {
            candidateB = 'A';
        }
        if (scoreB > 3) {
            candidateC = 'C';
        }
        System.out.println(candidateA);
        System.out.println(candidateB);
        System.out.println(candidateC);
    }
}
```

answer

```
B
D
C
```

7. What is the output after the program is run?

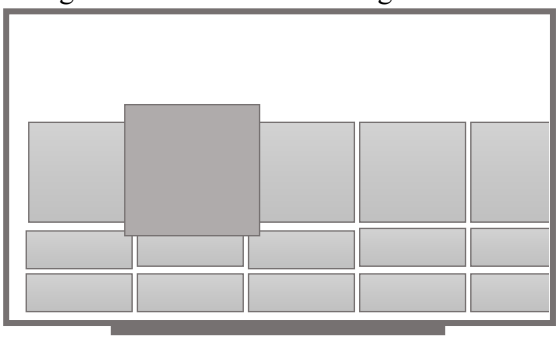
```
public class Regtangle {
    private double width;
    private double height;

    public Regtangle(double width, double height) {
        this.width = width;
        this.height = height;
    }

    public void setWidth(double width) {
        this.width = width;
    }

    public void setHeight(double height) {
        this.height = height;
    }

    public double getArea() {
        return this.width * this.height;
    }
}
--------------------------------------------------------------------------------
public class Exam07 {
    public static void main(String[] args) {
        double areaAA, areaBB, areaCC;
        Regtangle regA, regB;

        areaAA=0;
        areaCC=areaAA;
        regA = new Regtangle(5, 10);
        areaAA=regA.getArea();
        regB=regA;
        areaBB=regB.getArea();
        System.out.println("Part 1");
        System.out.println(areaAA);
        System.out.println(areaBB);
        System.out.println(areaCC);
        regB.setHeight(5);
        areaAA=regA.getArea();
        areaBB=regB.getArea();
        System.out.println("Part 2");
        System.out.println(areaAA);
        System.out.println(areaBB);
        System.out.println(areaCC);
    }
}
```

answer

```
Part 1
50.0
50.0
0.0
Part 2
25.0
25.0
0.0
```

8. **A remote control for a smart TV, as show below, consist of 5 navigation buttons, left, right, up, down, and ok button. Write a `Navigation2D` class that behaves like this navigation. For simplicity the `on/off` button is included in the class but it should be note that there is no interaction between the `Navigation2D` class and the `on/off` button.**

   - The ***left*** and ***right*** button navigate menu to the left and right side of the current column. It is a circular movement.



   - The ***up*** and ***down*** button navigate menu to the up and down side of the current row. It is a circular movement.
   - the ***Ok*** button return the value indicated the position of the current position

components: public class **`Navigation2D`**

- [2 points] **`Navigation2D(int row, int column)`**
  - Set the number of row and column
  - Initial the current position of cursor at row 0 and column 0
  - Initial the **`on`** button to false
- [2 points] **`moveLeft(): void, moveRight(): void`**
  - if it moves left beyond the first column, its value will become the last column
  - if it moves right beyond the number of columns, its value will become the first column
- [2 points] **`moveUp(): void , moveDown(): void`**
  - if it moves up beyond the first row, its value will become the last row
  - if it moves down beyond the last row, its value will become the first row
- [2 points] **`getCurrentPosition(): int`**
  - Return the current position which is computed by `currentRow* column+currentColumn +1`
- [2 points] **`turnOn(): void, turnOff(): void`**
  - Turn the switch on, off

answer

```
public class Navigation2D {

    private final int row, column;
    private int currentRow, currentColumn;
    private boolean on;

    public Navigation2D(int row, int column) {
        this.row = row;
        this.column = column;
    }
```

```
    public void moveLeft() {
        currentColumn--;
        if (currentColumn < 0) {
            currentColumn = column - 1;
        }
        // or
        //currentColumn=(currentColumn==0)?column - 1:currentColumn-1;
    }

    public void moveRight() {
        currentColumn++;
        if (currentColumn >= column) {
            currentColumn = 0;
        }
    }

    public void moveUp() {
        currentRow--;
        if (currentRow < 0) {
            currentRow = row - 1;
        }
    }

    public void moveDown() {
        currentRow++;
        if (currentRow >= row) {
            currentRow = 0;
        }
    }

    public int getCurrentPosition() {
        return currentRow * column + currentColumn + 1;
    }

    public void turnOn() {
        this.on = true;
    }

    public void turnOff() {
        this.on = false;
    }

    //(optional)
    public boolean isOn() {
        return on;
    }
}
```