Preguntas del documento ExamenJavaV3

Exercise 1.-

You are merging data from two sources connected to a network Access point to créate a new data packet.

You must merge strings a and b, then return a single merged string. A merge operation on two string is described as follows:

- Append alternating characters from a and b, respectively, to some new string, mergedString.
- Once all of characters in one of the string have been merged, append the remaining characters in the other string to mergedString.

As an example, asume you have two strings to merge: ‘abc and stuvwx’. Alternate between the first and second strings as long as you can:

‘a’ + ‘s’ + ‘b’ + ‘t’ + ‘c’ + ‘u’. At this point you have traversed the first string and have generated ‘asbtcu’. The remainder of the second string, ‘vwx’ is now added to the end of the string, creating ‘asbtcuvwx’.

Function Descriptiva

Complete the function mergeString in the editor below. The function must return the merged string.

mergeStrings has the following parameter(s):

a: first string

b: second string

Constraints

- 1 <= |a|,|b|<= 25000

Respuesta

```
public class Solution{
        static String mergeStrings(String a, String b){
        String abc="";
        if(a.length==b.length){
                for(int i=0;i<b.length;i++){
                abc.append(a[i]);
                abc.append(b[i]);
                }
        }else if(a.length>b.length){
                for(int i=0;i<a.length;i++){
                abc.append(a[i]);
```

```
                if(i<b.length)
                        abc.append(b[i]);
                }
        }else{
                for(int i=0;i<b.length;i++){
                if(i<a.length)
                        abc.append(a[i]);
                abc.append(b[i]);
                }
        }
        }
}
```

Exercise 2.-

You are designing a compiler for a C++ program and need to check that braces in any given file are balanced.

Braces in a string are considered to be balanced if the following criteria are met:

- All braces must be closed. Braces come in pairs of the form (), {} and []. The left brace opens the pair, and the right one closes it.

- In any set of nested braces, the braces between any pair must be closed.

For example, [{}] is a valid grouping of braces but [}]{} is not.

Function Description

Complete the function braces in the editor below. The function must return an array of strings where the string at each index i denotes whether or not the braces were balanced in a values. The array should consist of string “YES” or *NO* aligned with their indexes in values.

Braces has the following parameter(s):

Values[$values_{0,...,}values_{n-1}$] an array of strings to analyze

Constraints

- 1<= n<=15

- 1<= length of $values_i$ <= 100

- It is guaranteed that each $values_i$ consists of (,), {, }, [, and] only.

```
class BRaces{
        public String[] braceBalance(String[] args){

        }
}
```

//todo código

Exercise 3.-

Tables EMPLOYEES has 5 rows. Consider the following sequence of SQL statements.

SQL>CREATE TABLE myTable SELECT AS (SELECT* FROM EMPLOYEE);

SQL> INSERT INTO myTable SELECT * FROM myTable;

SQL> INSERT INTO myTable SELECT * FROM myTable;

SQL> INSERT INTO myTable SELECT * FROM myTable;

SQL> INSERT INTO myTable SELECT * FROM myTable;

If the SQL statement

SELECT COUNT (*) FROM myTable;

Is executed after executing all the statements listed above, what will be printed is.

Pick one of the choices:

- 80
- 25
- 20
- 5

resultado, 80

Exercise 4.-

Which testing is used for a program's individual components functionality testing?

Pick one of the choices:

- Functionality Testing
- Unit Testing
- Security Testing
- Smoke Testing
- Regression Testing

resultado, unit testing

Exercise 5.-

Consider the following Java code snippet:

```
public int divide(int a, int b){
        int c=-1;
        try{
                c=a/b;
        }catch(Exception e){
                System.err.print(“exception”);
        }finally{
                System.err.println(“finally”);
        }
        return c;
}
```

What will our code print when we call divide (4,0)?

Pick one of the choices:

- Exception Finally
- Finally Exception
- Exception

resultado, exception finally

Exercise 6.-

The feature which allows different methods to have the same name and arguments type, but the different implementation is called?

Pick one of the choices:

- Overloading
- Overriding

- Java does not permit methods with same and type signature
- None of the above

resultado, java does not permit methods with same name and type signature

Exercise 7.-

What does the following for loop output?

```
for(int i=10,j=1; i>j; --i, ++j)
        System.out.print(j%i);
```

Pick one of the choices:

- 12321
- 12345
- 11111
- 00000

resultado, 12345

Exercise 8.-

Identify all representation(s) that REST uses to represent resources:

Pick all the choices that aply:

- Text
- Xml
- Json
- None of these

resultado, text, xml y json

Exercise 9.-

There are multiple records in a table, and some of these records are duplicates. You need to fetch only one copy of each set of duplicate records. Which command will you use.?

Pick one of the choices:

- Select distinct

- Select unique
- Select different
- All the above

resultado, select distinct

Exercise 10.-

Which SQL keyword is used to retrieve a máximum value?

Pick one of the choices:

- TOP
- MOST
- UPPER
- MAX

resultado, max

Exercise 11.-

We perform the following sequence of actions:

1. Insert the following elements into a set: 1,2,9,1,2,3,1,4,1,5,7.
2. Convert the set into a list and sort it in ascending order.

Which option denotes the sorted list?

Pick one of the choices:

- {1, 2, 3, 4, 5, 7, 9}
- {9, 7, 5, 4, 3, 2, 1}
- {1, 1, 1, 1, 2, 2, 3, 4, 5, 7, 9}
- None of the above

resultado, {1,2,3,4,5,7,9}

Exercise 12.-

Identify any ítem(s) in the list below that are not HTTP verbs:

Pick one of the choices:

· GET

· POST

· HEAD

· DELETE

· REMOVE

· PUT

· PATCH

resultado, remove

Exercise 13.-

In this challenge, you will be asked to build on an abstract and initialize an instance of each class with a variable. The program will then test your implementation by retrieving the data you stored.

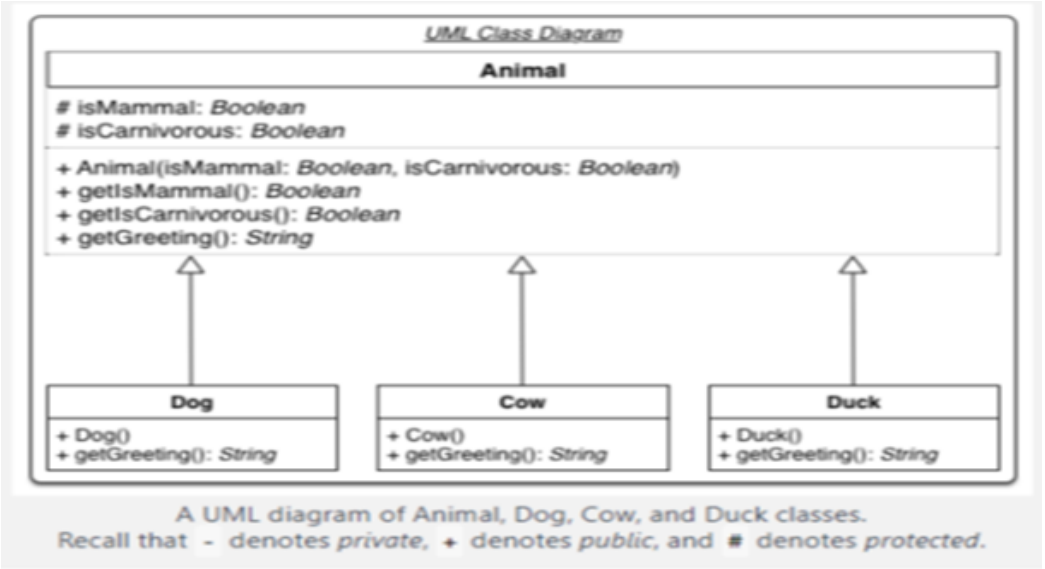
The locked code in the editor does the following:

1. Declares an abstract class named Animal with the implementations for getIsMammal() and getIsCarnivorous() methods, as well as an abstract method named getGreeting().

2. Creates Dog, Cow and Duck objects.

3. Calls the getIsMammal(), getIsCarnicorous(), and getGreeting() methods on each of these respective objects.

Consider the following UML diagram:



A UML diagram of Animal, Dog, Cow, and Duck classes.
Recall that - denotes *private*, + denotes *public*, and # denotes *protected*.

Complete the code in the editor below to implement the following:

1. Three clases named Dog, Cow and Duck that inherit from the Animal class.
2. No-argument constructors for each class that initialize the instance variables inherited from the superclases.
3. Each class must implement the getGreeting() method:
    a. For a Dog object, this must return the string ruff.
    a. For a Cow object, this must return the string moo.
    b. For a Duck object, this must return the string quack.

Input Format

There is no input for this challenge.

Output Format

The getGreeting() method must always return a string denoting the appropriate greeting for the implementing class.

Sample Output

A dog says ‘ruff’, is carnivorous, and is a mammal.

A cow says ‘moo’, is not carnivorous, and is a mammal.

A duck says ‘quack’, is not carnivorous, and is not a mammal

```
abstract class Animal{
        protected boolean isMammal;
        protected boolean isCarnivorous;
        public Animal(boolean isMammal, boolean isCarnivorous){
                this.isMammal=isMammal;
                this.isCarnivorous=isCarnivorous;
        }
        public boolean getIsMammal(){
                return this.isMammal;
        }
```

```
        public boolean getIsCarnivorous(){
                return this.isCarnivorous;
        }
        abstract public String getGreeting();
        public void printAnimal(String name){
                System.out.println(
                        "A" + name + " says '" +this.getGreeting()
            +   "' is " + (this.getIsCarnivorous()? "": "not")
            +   "carnivorous, and is "+(this.getIsMammal()? "": "not") + "a mammal"
        }
}
public class Solution{
        public static void main(String[] args){
                Animal dog=new Dog();
                dog.printAnimal("dog");
                Animal cow=new Cow();
                cow.printAnimal("cow");
                Animal duck=new Duck();
                duck.printAnimal("duck");
        }
}
//todo code
```

Exercise 14.-

What is the output for the below Java code?

```
public class Test{
        public static void main(String[] args){
```

```
        int i=010;
        int j=07;
        System.out.println(i);
        System.out.println(j);
    }
}
```

Pick one of the choices:

- 8 7
- 10 7
- Compilation fails with an error at line 3
- Compilation fails with an error at line 5

Exercise 15.-

A public data member with the same name is provided in both base as well as derived clases. Which of the following is true?

Pick one of the choices:

- It is a compiler error to provide a field with the same name in both base and derived class
- The program will compile and this feature is called overloading
- The program will compile and this feature is called overriding
- The program will compile and this feature is called as hiding or shadowing

resultado, the program will compile and this feature is called hiding or shadowing

Exercise 16.-

Given three clases A, B and C.

B is a subclass of A

C is a subclass of B

Which one of these boolean expressions is true only when an object denoted by reference o has actually been instantiated from class B, as opposed to from A or C?

Pick one of the choices:

- (o instanceof B) && (! (o instanceof A))
- (o instanceof B) && (! (o instanceof C))
- (o instanceof B)
- None of the above

resultado, (o instanceof B) && (! (o instanceof C))

Exercise 17.-

Which statement is true?

Pick one of the choices:

- Non-static member classes must have either default or public accessibility
- All nested classes can declare static member classes
- Methods in all nested classes can be declared static
- Static member classes can contain non-static methods

resultado, Static member classes can contain non-static methods

Exercise 18.-

A constructor is called whenever

Pick one of the choices:

- An object is declared
- An object is used
- A class is declared
- A class is used

resultado, an object is declared

Exercise 19.-

You are implementing a student registration and student information retrieval system for a school using a simple class roster in Java. When a stundent is registered, the system must assign an integer ID(enrollmentNumber), starting at 1 and adding 1 as

each student is registered. The stundent's name is stored with the assigned enrollmentNumber. The retrieval request should return a student's registration information.

The student class should implement:

- The constructor. Student (String name)
- The method String toString() to return the string "(enrollmentNumber):(name)"

The locked stub code in the editor validates the implementation of the Student class.

After each student is registered, the code stub requests and prints the student's information to test your code.

Constraints

- 1 <= numberOfStudents <=10^3

Sample input for custom testing

3

Erica

Bob

Maria

Sample output

1: Erica

2: Bob

3: Maria

Explanation

The three students are registered in the following order:

- The first student to be registered is Erica so she is assigned 1 as the enrollment number by the portal.
- Bob is second, so he is assigned the number2.
- Maria is third, so she es assigned the numer 3.

Now, the information of all the students is printed in the order in which they are registered.

//Todo, código

Exercise 20.-

Which of the following data types in Java are primitive?

Pick one of the choices:

· String

· Struct

· Boolean

· char

resultado, char

Exercise 21.-

Which of the following are true for Java Classes?

Pick one of the choices:

· The Void class extends the Class class

· The Float class extends the Double class

· The System class extends the Runtime class

· The Integer class extends the Number class

resultado, Integer class extends the Number class

Exercise 22.-

The following code snippet is a demostration of a particular design pattern. Which desing pattern is it?

```
public class Mystery{

    private static Mystery instance=null;

    protected Mystery(){
```

```
            public static Mystery getInstance(){
                if(instance==null){
                    instance=new Mystery();
                }
                return instance
            }
        }
}
```

Pick one of the choices:

- Factory Design Pattern
- Strategy Pattern
- Singleton
- Facade Design Pattern

resultado, singleton

Exercise 23.-

Which of the following Java declaration of the String array is correct?

Pick one of the choices:

- String temp [] = new String {“j”, “a”, “z”};
- String temp [] = {“j” “b” “c”};
- String temp = {“a”, “b”, “c”};
- String temp [] = {“a”, “b”, “c”};

resultado, String temp [] = {“a”, “b”, “c”};

Exercise 24.-

Which is true of the following program?

package exam.java;

```
public class TestFirstApp{
    static void doIt(int x,int y, int m){
        if(x==5) m=y;
            else m=x;
    }
    public static void main(String[] args){
        int i=6, j=4, k=9;
        TestFirstApp.doIt(i,j,k);
        System.out.println(k);
    }
}
```

Pick one of the choices

- Doesn't matter what the values of *i* and *j* are, the output will always be *5*.
- Doesn't matter what the values of *k* and *j* are, the output will always be *5*.
- Doesn't matter what the values of *i* and *j* are, the output will always be *9*.
- Doesn't matter what the values of *k* and *j* are, the output will always be *9*.

resultado, Doesn't matter what the values of *i* and *j* are, the output will always be *9*.

Exercise 25.-

Which of the following statements are correct. Select the correct answer.

- Each Java file must have exactly one package statement to specify where the class is stored.
- If a java file has both import and package statement, the import statement must come before package statement.
- A java file has at least one class defined
- If a java file has a package statement, it must be the first statement (except comments).

Resultado, If a java file has a package statement, it must be the first statement (except comments).

Exercise 26.-

What is the output for the following program:

```
class Constructor{
    static String str;
    public void Constructor(){
        System.out.println(“in constructor”);
        str=”hello world”;
        }
        public static void main(String[] args){
                Constructor c= new Constructor();
                System.out.println(str);
        }
}
```

Pick one of the choices

- In Constructor
- Null
- Compilation Fails
- None of the above

resultado, null

Exercise 27.-

Given the following code, what is the most likely result.

```
import java.util.*;
public class Compares{
```

```
    public static void main(String[] args){
        String[] cities= {“Bangalore”,”pube”, ”san francisco”, “new york”};
        MySort ms=new MySort();
        Arrays.sort(cities.ms);
        System.out.println(Arrays.binarySearch(cities,”new york”));
        }
        static class MySort implements Comparator{
                public int compare(String a, Stringb){
                        return b.compareTo(a);
                }
        }
}
```

Pick one of the choices

- -1
- 1
- 2
- Compilation fails

resultado, compilations fails

Exercise 28.-

Anna has an array of n integers called num. She can reduce the array by 1 element by performing a move. Each move consists of the following three steps:

1. Pick two different elements numfij and numifi. i
2. Remove the two selected elements from the array.
3. Add the sum of the two selected elements to the end of the array.

Each move has a cost associated with it and this cost is equal to the sum of the two elements removed from the array during the move. Anna wishes to calculate the minimum total cost of reducing her array to one element.

For example, consider the array num = (4.6,8). Anna removes 4 and 6 in her first move at a cost of 4 + 6 = 10. and the resultant array is num_1 = [18,10]. She then removes 8 and 10 in her second move at a cost of 8 + 10 = 18, and the resultant array is num_2 = [18]. The total cost of reducing this array to one element using this sequence of moves is 10 + 18 = 28. This is just one set of possible moves. For instance, she could have started with 4 and 8.

Function Description

Complete the function reductionCost in the editor below. The function must return the minimum total cost of reducing the input array to one element

reductionCost has the following parameter(s):

num[num[0],...num[n-1]]: an array of integers

Constraints

· 2<= n <= 10^4

· 0 <= num[i] <= 10^5

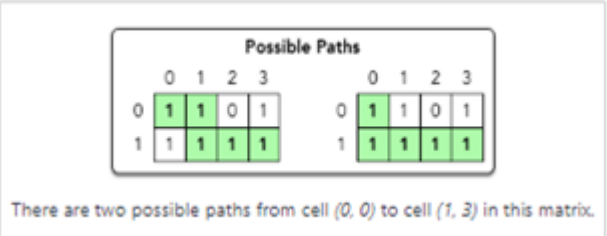
//todo code

Exercise 29.-

Consider a maze mapped to a matrix with an upper left comer at coordinates (row, column) = (0, 0). Any movement must be in increasing row or column direction. Determine the number of distinct paths through the maze. Always start at position (0, 0), the top left. and end up at (max(row), max(column)), the bottom right.

As an example, consider the following diagram where 1 indicates an open cell and 0 indicates blocked. It is only possible to travel through open cells, so no path can go through the cell at (0. 2). There are two distinct paths to the goal.

Possible Paths



Function Description

Complete the function number0fParhs in the editor below. The function must return the number of paths through the matrix, modulo (10^9 + 7).

numberOfeaths has the following parameter(s):

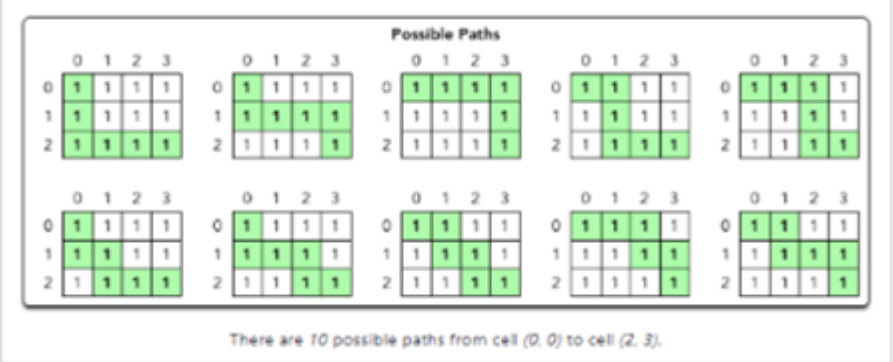
a [a[0],…a[n-1]]: an array of strings, each a row of the matrix where each character represents a column

Constraints

· 1 <=n, m <= 1000

· Each cell in matrix a contains either a 0 or a 1.

Explanation



//todo code

Exercise 30.-

Given an array of words representing your dictionary, you test each word to see if it can be made into another word in the dictionary. This will be done by removing characters one at a time. Each word represents its own first element of its string chain, so start with a string chain length of 1. Each time you remove a character, increment your string chain by 1. In order to remove a character, the resulting word must be in your original dictionary. Your goal is to determine the longest string chain achievable for a given dictionary.

For example, given a dictionary [a, and, an, bear], the word and could be reduced to an and then to a. The single character a cannot be reduced any further as the null string is not in the dictionary. This would be the longest string chain, having a length 3. The word bear cannot be reduced at all.

Function Description

Complete the function longestChain in the editor below. The function must return a single integer representing the length of the longest string chain. longestChain has the following parameter(s):

words[words[0]....words[n-1]]: an array of strings to test

Constraints

· 1 <= n <= 50000

· 1 <= lword[i] <= 60, where 0 <= i < n

· Each words[i] is composed of lowercase letters in ascii[a-z].

Sample Input 0

6, a, b, ba, bca, bda, bdca

Sample Output 0

4

Explanation 0

Sample Case 1: words = {“a”, “b”, “ba”, “bca”, “bda”, “bdca”}

Because 'a' and 'b' are single-character words, we cannot remove any characters from them as that would result in the empty string, so the length for both of these string chains is 1.

The word “ba” can create two different string chains each with a length of 2: ("ba” — “a” and “ba”— 'b'). This means our current longest string chain is 2.

The word “bca” can create two different string chains of length 3: ('bca” ---“ba” — "a" and “bca”— “ba”— “b”). This means our current longest string chain is now 3.

The word 'bda' can create two different string chains of length 3: ('bda" — 'ba' — “a” and “bda" — 'ba' — “b”. At this point, our current longest string chain is still 3.

The word 'bdca' can create four different string chains of length: 4 (“bdca” — 'bda' — "ba" — “a”, 'bdca' — "bda” — "ba” — 'b', 'bdca' — “bca” — “ba” — 'a', and 'bdca' — “bca” — "ba” — 'b'). This means our current longest string chain is now 4.

The longest string chain is 4.

//todo code

Exercise 31.-

You are implementing a student registration and student information retrieval system for a school using a simple class roster in Java. When a student is registered, the system must assign an integer ID (enrollmentNumber), starting at 1 and adding 1 as each student is registered. The student's name is stored with the assigned enrollmentNumber. The retrieval request should return a student's registration information.

The Student class should implement:

· The constructor.Student(String name)

· The method String toString() to return the string "{enrollmentNumber}: {name}"

The locked stub code in the editor validates the implementation of the Student class.

After each student is registered, the code stub requests and prints the student's information to test your code.

Constraints

· 1 <= numberOfStudents <= 10^3

Input Format For Custom Testing

The first line contains the value of numberOfStudents describing the total number of students being registered. Each of the next numberOfStudents lines contains the student name.

Sample Case 0

Sample Input For Custom Testing

3, Erica, Bob, Maria

Sample Output

1: Erica

2: Bob

3: Maria

Exercise 32.-

To delete all pairs of keys and values in a given HashMap, which of the following methods should be used?
Pick one of the choices

a) clearAll()

b) empty()

a) remove()

b) clear()

resultado, b, clear()

Exercise 33.-

Which pattern do you see in the code below:
java.util.Calendar.getInstance();
Pick one of the choices

a) Singleton Pattern

b) Factory Pattern

c) Facade Pattern

d) Adaptor Pattern

resultado, factory pattern

Exercise 34.-

What is the output of the following program:

```
interface BaseI { void method (); }

	class BaseC

	{

		public void method()

		{

			System.out.printIn(“Inside BaseC: :method”);

		}

	}

	class ImplC extends BaseC implements BaseI

	{

		public static void main (String [ls)

		{
```

```
                (new ImplC()).method();
        }
    }
```

Pick one of the choices

a) Null

b) Complication fails

c) Inside BaseC::method

d) None of the above

resultado, b, compilation fails, falta implementación de método de interfaz en implC

Exercise 35.-

Consider the following three classes:

```
class A {}

    class B extends A {}

    class C extends B {}
```

Consider an object of class B is instantiated, i.e.,

```
    B b = new B();
```

Which of the following Boolean expressions evaluates to true:
Pick one of the choices

a) (b instanceof B)

b) (b instanceof B) && (!(b instanceof A))

c) (b instanceof B) && (!(b instanceof C))

d) None of the above

resultado, tanto a cómo c son correctas

Exercise 36.-

What is the output of the following program:

```
class Constructor
    {
```

```
        static String str;
        public void Constructor(){
                System.out.printIn("In constructor");
                str = "Hello World";
        }
        public static void main (String [] args){
                Constructor C = new Constructor ();
                System.out.printIn(str);
        }
}
```

Pick one of the choices

a) In Constructor

b) Null

c) Compilation fails

d) None of the above

resultado, null

Exercise 37.-

☆ **The Robot Class**

We need a *Robot* class that can move around on a two dimensional plane. It needs to be able to change its position, report its position and report its last move as described below. Implement a *Robot* class per the following specifications:

Fields		
Data Type	**Name**	**Description**
`integer`	`currentX`	The robot's current x-coordinate in the *2D* plane.
`integer`	`currentY`	The robot's current y-coordinate in the *2D* plane.
`integer`	`previousX`	The robot's x-coordinate in the *2D* plane prior to its most recent movement.
`integer`	`previousY`	The robot's y-coordinate in the *2D* plane prior to its most recent movement.
Note: The robot's initial location is at *(x, y)* coordinate *(0, 5)*.		

Parametrized constructor

Data Type	Param. Name	Description
integer	x	The value of *currentX* for the new Robot.
integer	y	The value of *currentY* for the new Robot.
Note: The robot created by this constructor is considered to have spawned at *(0, 5)* and moved to *(currentX, currentY)* so *(previousX, previousY)* starts as *(0, 5)*.		

Methods				
Return Type	**Method Name**	**Param. Type**	**Param. Name**	**Description**
void	moveX	integer	dx	Move the robot from current position *(x, y)* to new position *(x + dx, y)*. Remember to maintain *previousX*.
void	moveY	integer	dy	Move the robot from current position *(x, y)* to new position *(x, y + dy)*. Remember to maintain *previousY*.
void	printCurrentCoordinates	no parameters		Print two space-separated integers describing the robot's current *x* and *y* coordinates.
void	printLastCoordinates	no parameters		Print two space-separated integers describing the robot's *previousX* and *previousY* coordinates. This will be called after the robot has moved from position *(0, 5)* at least once.
void	printLastMove	no parameters		Print two space-separated values describing the robot's most recent movement: • If the last move was *moveX(dx)*, print *x dx* where *x* is the actual character *x* and *dx* is the distance moved in the x-direction during the last call to *moveX*. • If the last move was *moveY(dy)*, print *y dy* where *y* is the actual character *y* and *dy* is the distance moved in the y-direction during the last call to *moveY*.

The code provided has a complete definition for a main method that creates *Robot* objects and tests the class' methods. Implement the *Robot* class according to the criteria above to pass all test cases.

Constraints

- *-100 ≤ x, y, dx, dy ≤ 100*

▸ **Input Format for Custom Testing**

▾ **Sample Case 0**

Sample Input 0

```
2
1
1
1
```

Sample Output 0

```
0 5
2 1
x 1
3 1
3 1
x 2
5 2
5 2
```

Explanation 0

- The *firstRobot* object is initially at position *(0, 5)*, so the call to *firstRobot.printCurrentCoordinates()* prints *0 5*.
- For the *secondRobot* object created with the parameterized constructor, it was created and moved from *(0, 5)* → *(2, 1)* so *secondRobot.printCurrentCoordinates()* prints *2 1*. Next, we call the following sequence of methods on the *secondRobot* object:
  1. *secondRobot.moveX(1)* moves the robot *1* unit from *(2, 1)* → *(3, 1)*.
  2. *secondRobot.printLastMove()* prints *x 1* as its last movement was *moveX(1)*.
  3. *secondRobot.printCurrentCoordinates()* prints *3 1* because it moved from *(2, 1)* → *(3, 1)*.
  4. *secondRobot.moveY(1)* moves the robot *1* unit from *(3, 1)* → *(3, 2)*.
  5. *secondRobot.printLastCoordinates()* prints *3 1* because its last movement was *(3, 1)* → *(3, 2)*, so the coordinates of its last location prior to the movement was *(3, 1)*.
     - At this point, test code adds *1* to *dx* ⇒ *dx = 2* and subtracts *1* from *dy* ⇒ *dy = 0*.
  6. *secondRobot.moveX(2)* moves the robot from the *(3, 2)* → *(5, 2)*.
  7. *secondRobot.printLastMove()* prints *x 2* as its last movement was *moveX(2)*.
  8. *secondRobot.printCurrentCoordinates()* prints *5 2* because it moved from *(3, 2)* → *(5, 2)*.
  9. *secondRobot.moveY(0)* moves the robot *0* units from *(5, 2)* → *(5, 2)*.
  10. *secondRobot.printLastCoordinates()* prints *5 2* because its last movement was *(5, 2)* → *(5, 2)*.

YOUR ANSWER

```java
import java.util.Scanner;
/*
 * Create the class Robot. Do not use the public access modifier in your class declaration.
 */
public class RobotMoves {
    private static final Scanner scan = new Scanner(System.in);

    public static void main(String[] args) {
        int x = scan.nextInt();
        int y = scan.nextInt();
        int dx = scan.nextInt();
        int dy = scan.nextInt();

        Robot firstRobot = new Robot();
        firstRobot.printCurrentCoordinates();

        Robot secondRobot = new Robot(x, y);
        secondRobot.printCurrentCoordinates();

        for (int i = 1; i < 3; i++) {
            secondRobot.moveX(dx);
            secondRobot.printLastMove();
            secondRobot.printCurrentCoordinates();
            secondRobot.moveY(dy);
            secondRobot.printLastCoordinates();

            dx += i * i;
            dy -= i * i;
        }
    }
}
```

Exercise 38.-

☆ Separating Students

There are a number of students standing in a single-file line. Each student is numbered sequentially from *0*. Each student also has a binary digit associated with them, where *0* indicates the student's performance is above average and *1* indicates the student's performance is below average.

The teacher wants to group the students by performance on opposite ends of the line such that the number of adjacent pairs of students where one student is a *0* and the other student is a *1* is *minimized*. To accomplish this, any student can swap places in line with the student located immediately in front of or behind them. Each time a pair of students swaps places, it counts as a *move*. Determine the minimum number of moves needed to create an optimal configuration.

For example, there are *n=4* students arranged as *avg=[0,1,0,1]*. With *1* move, switching students *1* and *2*, we get the array *[0,0,1,1]* which is optimal.

Function Description

Complete the function *minMoves* in the editor below. The function must return an integer denoting the minimum number of moves necessary to achieve an optimal configuration.

minMoves has the following parameter(s):

avg[avg[0],...avg[n-1]]: an array of binary digits

Constraints

- $1 \le n \le 10^5$
- $avg[i] \in [0,1)$

▼ Sample Case 0

Sample Input 0

```
8
1 1 1 1 0 0 0 0
```

Sample Output 0

```
0
```

Explanation 0

There is only one pair of above and below average students standing next to each other. This cannot be reduced, so no moves are necessary.

Draft saved 08:39 am

Original Code | Java 8

```
1  import java.io.*;
14
15 class Result {
16
17     /*
18      * Complete the 'minMoves' function below.
19      *
20      * The function is expected to return an INTEGER.
21      * The function accepts INTEGER_ARRAY avg as parameter.
22      */
23
24     public static int minMoves(List<Integer> avg) {
25     // Write your code here
26
27     }
28
29 }
30
31 public class Solution {
```

```java
public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));
        BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter(System.getenv("OUTPUT_PATH")));

        int avgCount = Integer.parseInt(bufferedReader.readLine().trim());

        List<Integer> avg = IntStream.range(0, avgCount).mapToObj(i -> {
            try {
                return bufferedReader.readLine().replaceAll("\\s+$", "");
            } catch (IOException ex) {
                throw new RuntimeException(ex);
            }
        })
            .map(String::trim)
            .map(Integer::parseInt)
            .collect(toList());

        int result = Result.minMoves(avg);

        bufferedWriter.write(String.valueOf(result));
        bufferedWriter.newLine();

        bufferedReader.close();
        bufferedWriter.close();
    }
}
```

Exercise 39.-

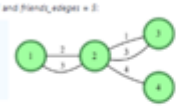
☆ Shared Interests

Given a graph of friends who have various interests, determine which groups of friends have the most interests in common. You will then use a little math to determine a value to return.

The graph will be represented as a series of nodes numbered consecutively from 1 to *friends_nodes*. Friendships have evolved based on interests which will be represented as weights in the graph. Any members who share the same interest are said to be connected by that interest. Once you have determined the node pairs with the maximum number of shared interests, return the maximal product of the node pairs' labels.

For example, you are given a graph with *friends_nodes* = 4 and *friends_edges* = 3:

```
From  To   Weight
1     2    2
1     2    3
2     3    1
2     3    3
2     4    4
```



If we look at each interest, we have the following connections:

```
  Weight
(Interest)    Connections
    1           2,3
    2           1,2
    3           1,2,3
    4           2,4
```

In this case, we see the pairs (1, 2) (interests 2 and 3) and (2, 3) (interests 1 and 3) each share two interests which is maximal. We take the products of the node numbers: $1 * 2 = 2$ and $2 * 3 = 6$. The maximal value 6 is returned.

Function Description

Complete the function *maxShared* in the editor below. The function must return the maximal integer product of all node pairs sharing the most interests. Indexes align *friends_from*, *friends_to* and *friends_weight*.

maxShared has the following parameter(s):
- *friends_nodes*: integer number of nodes
- *friends_edges*: integer number of edges
- *friends_from*[from[0]...from[edges-1]]: an array of integers
- *friends_to*[to[0]...to[edges-1]]: an array of integers
- *friends_weight*[weight[0]...weight[edges-1]]: an array of integers

Constraints

- $2 \le friends\_nodes \le 100$
- $1 \le friends\_edges \le min(200, \frac{friends\_nodes * (friends\_nodes - 1)}{2})$
- $1 \le friends\_weight[i] \le 100$
- $1 \le friends\_from[i], friends\_to[i] \le friends\_nodes$
- $1 \le friends\_weight[i] \le friends\_edges$
- $friends\_from[i] \ne friends\_to[i]$
- Each pair of friends can be connected by zero or more interests.

▸ **Input Format for Custom Testing**

▾ **Sample Case 0**

Sample Input 0

```
4 5
1 2 1
1 2 2
2 3 1
2 3 3
2 4 3
```

Sample Output 0

```
6
```

preguntas del documento “preguntas java apx segunda vuelta.pdf”

1. The standard API for accessing databases in Java is:

a. JPA/Hibernet

b. JDBC

c. ODBC

respuesta, b, la jdbc es la estandar

2. Valid argument for catch clause:

A. Throwable

B. Exception but NOT including RuntimeException

C. CheckedException

D. RunTimeException

E. Exception

respuesta, a, d y e, estas tres son posibles de atrapar con un catch

3. Asume que este switch acepta datos de tipo entero. ¿Qué tipo de dato no se podría colocar en el espacio en blanco?

```
class App{
        public static void main(String[] args) {
                _____ dayOfWeek= 1;
                switch (dayOfWeek) {
                        case 1:
                                System.out.println("Monday");
                        case 2:
```

```
                    System.out.println("Tuesday");
                case 3:
                    System.out.println("Other day");
                    break;
            }
        }
}
```

A. int

B. long

C. char

D. None of the above

respuesta, b, porque se puede poner int en el switch, d queda descartada, char tambien se puede debido a que se puede evaluar cómo int

4. Elige todas las excepciones no verificadas

a. IllegalArgumentException

b. NumberFormatException

c. IOException

d. NullPointerException

e. Exception

f. ArrayIndexOutOfBoundsException

respuesta, a, b, d, f, estas son excepciones que se ven al momento de correr el programa, de ahi el nombre unchecked

5. ¿Cuál es el resultado?

```
public class Main{
    public static void main(String[] args) {
        int[] ii = null;
        for (int xx : ii) {
```

```
                System.out.println(xx);
            }
        }
}
```

a. java.lang.NullPointerException

b. null

respuesta, a, se intento hacer alguna operacion, en este caso iterar, sobre un null

6. ¿Cuál es la salida?

```
public class Movie {
    public static void main(String[] args) {
        String movie = "Thriller";
        switch (movie) {
            case "Thriller":
                System.out.println("Movie Thriller");
            case "Comedy":
                System.out.println("Movie Comedy");
            case "Romance":
                System.out.println("Movie Romance");
                break;
            case "Action":
                System.out.println("Movie Action");
        }
    }
}
```

a. Movie Thriller

b. Movie Thriller Movie Comedy

c. Movie Thriller Movie Comedy Movie Romance

respuesta, c, cómo no hay breaks, se sigue ejecutando hasta que encuentra el primero, en este caso es el que esta en romance

7. ¿Cuál es el resultado?

```
public class strings {
	public static void main(String[] args) {
		String s = "ABCD";
		s.trim();
		s.toUpperCase();
		s+= " 123";
		System.out.println(s.length());
	}
}
```

respuesta, 8

8. Usando StringBuilder, elige las opciones que den el mismo resultado que la

concatenación original. (Elige todas las opciones aplicables).

```
public class Cadena {
	public static void main(String[] args) {
		String name = "Joe";
		int age = 31;
		String result = "My name is " + name + ", my age is " + age;
		System.out.println(result);
		StringBuilder sb = new StringBuilder();
	}
}
```

a. sb.append("My name is " + name + ", my age is " + age);

b. sb.insert("My name is " + name).append(", my age is "+ age);

c. sb.insert("My name is").insert(name).insert(", my age is ").insert(age);

d. sb.append("My name is ").append(name).append(", my age is ").append(age);

respuesta, a y d, ambos usan correctamente el append y manejan la string de forma correcta, con insert se ocupa pasar un offset primero

9. ¿Qué afirmación es correcta sobre el siguiente código?

```
public class MyStuff {
        String name;
        MyStuff(String n) {name = n;}
        public static void main(String[] args) {
                MyStuff m1 = new MyStuff("guitar");
                MyStuff m2 = new MyStuff("tv");
                System.out.println(m2.equals(m1));
        }
        public boolean equals(Object o) {
                MyStuff m = (MyStuff) o;
                if (m.name != null) {return true;}
                return false;
        }
}
```

a. The output is true and MyStuff fulfills the Object.equals() contract

b. The output is false and MyStuff fulfills the Object.equals() contract

c. The output is true and MyStuff does NOT fulfill the Object.equals() contract.

d. The output is false and MyStuff does NOT fulfill the Object.equals() contract.

respuesta, c, aunque da verdadero, esto no indica que sean iguales en contenido, ya que siempre y cuando el resultado no sea nulo seran iguales ante los ojos de ese método

10. Which two statments are true?

A. An interface CANNOT be extended by another interface.

B. An abstract class can be extended by a concrete class.

C. An abstract class CANNOT be extended by an abstract class.

D. An interface can be extended by an abstract class.

E. An abstract class can implement an interface.

F. An abstract class can be extended by an interface.

resultado, b, e, una clase abstracta puede ser extendida por una clase normal, al igual que una clase abstracta puede implementar una interfaz

11. ¿Cuál es el resultado?

```
public class DoWhile {
	public static void main(String[] args) {
		int ii = 2;
		do {
			System.out.println(ii);
		} while (--ii);
	}
}
```

a. 2 1 0

b. 1 0

c. null

d. An infinite loop

e. Compilation fails

resultado, e, while solo acepta boolean statements

12. ¿Cuál es el resultado?

```
public class Bucle {
	public static void main(String[] args) {
		int[] at = { 1, 2 };
		for (int x : at) {
```

```
                    System.out.println(x + ", ");
                    if (x < at.length) {
                        break;
                    }
                }
            }
        }
```

a. 1, 2,

b. No compila

c. 1,

d. 1, 2

resultado, c, ya que 1 es menor que la longitud total de la cadena (2), imprime solamente uno y sale del bucle

13.

```
Int a = 10; int b = 37; int z = 0; int w = 0;
If (a==b) {z=3; } else if (a>b) {z=6;}
w = 10 * z;
```

a. 30

b. 0

c. 60

resultado, b, no entra a ninguno de los ifs y por ende no hace ninguna de las operaciones

14. ¿Cuál es el resultado?

```
class Control {
    public static void main(String[] args) {
        int i = 42;
        String s = (i < 40) ? "Greater than" : false;
        System.out.println(s);
```

```
        }
}
```

a. Error de compilación

b. Greater then

c. Greater then false

d. false

resultado, a, no se permite asignarle un valor que no sea string a un string

15. ¿El siguiente código compila?

```
public class estructura {
        public static void main(String[] args) {
                int num = 1;
                switch (num) {
                        case 1:
                        case 2:
                                System.out.println("Caso 2");
                        case 3:
                                System.out.println("Caso 3");
                                break;
                }
        }
}
```

Si, compila, no hay ningun problema con el código

16. ¿Cuál de los siguientes códigos podría estar presente en un método que no retorna ningún valor? (Elige dos opciones)

a. return null;

b. return void;

c. return;

d. Omitir el uso de la palabra clave return

resultado, c y d, return solamente se recomienda cuando la ejecución del método fuera interrumpida antes, mientras que lo normal es no usar return

17. What two keywords may precede the word ‘class’ in a class declaration?

a. Volatil

b. Public

c. Local

d. Synchronized

e. Static

resultado, b y e

18. Which three are bad practices?

a. Checking for an IOException and ensuring that the program can recover if

one occurs.

b. Checking for ArrayIndexOutOfBoundsException and ensuring that the

program can recover if one occurs.

c. Checking for FileNotFoundException to inform a user that a filename entered

is not valid.

d. Checking for Error and, if necessary, restarting the program to ensure that

users are unaware problems.

e. Checking for ArrayIndexOutOfBoundsException when iterating through an

array to determine when all elements.-have been visited.

resultado, b, d y e, b es mala practica porque no deberia pasar y aun si pasara no seria buena idea meter runtimes dentro de un try-catch, d es mala practica porque reiniciar el programa podria causar que se volviera a repetir el problema y causarle frustracion al usuario y e es mala idea simplemente porque se estaria esperando a un mal funcionamiento del programa para que termine de ejecutarse

19. _____ occurs when a subclass redefines a method from its superclass, while

_____ happens when multiple methods in the same class share the same name but differ in their parameters.

a. Overloading / Overriding

b. Overriding / Overloading

c. Inheriting / Overriding

d. Overloading / Inheriting

resultado, b

20. What changes will make this code compile?

```
class X{
        X(){ }
        private void one(){ }
}
public class Y extends X{
        Y(){}
        private void two(){ one();}
        public static void main(String[] args){
                new Y().two();
        }
}
```

A. Adding the public modifier to the declaration of class X.

B. Adding the protected modifier to the class X() constructor.

C. Changing the private modifier on the declaration of the one() method to protected.

D. Removing the Y() constructor.

E. Removing the private modifier from the two() method

resultado, c, esto porque no encuentra el método de la clase x ya que solo la clase x puede llamarla, pero no estamos en x sino en su hijo y, haciendo que sea inaccesible y causando un error

21. ¿Cuáles de las siguientes opciones son instanciaciones e inicializaciones válidas de

un arreglo multidimensional? Elige dos

A. int[][] array2D = {{0, 1, 2, 4}{5, 6)};

B. int[][] array2D = new int[][2];

array2D[0][0] = 1;

array2D[0][1] = 2;

array2D[1][0] = 3;

array2D[1][1] = 4;

int[] array3D = new int[2][2][2];

C. int[][][] array3D = {{{0, 1}, {2, 3}, {4, 5}}};

int[] array = {0, 1};

D. array3D[0][0] = array;

array3D[0][1] = array;

array3D[1][0] = array;

array3D[1][1] = array;

resultado, a y c

22. ¿Cuál es la salida?

```
public class Main{
        public static void main(String[] argos){
                int x=2;
                if(x==2) System.out.println(“A”);
                else System.out.println(“B”);
                else System.out.println(“C”);
        }
```

}

resultado, no compila, hay un else sin nada mas

23. Concatenación de strings e ints

```
public static void main(String[] argos){
        System.out.println(“result” + 3+5);
        System.out.println(“result”+ (3+5));
}
```

A. Result: 8 Result: 8

B. Result: 35 Result: 8

C. Result: 8 Result: 35

D. Result: 35 Result: 35

resultado, b, esto porque en la primera esta sumando todo cómo cadena, en la segunda suma 3+5 cómo enteros para luego sumarlos cómo cadena

Preguntas del documento “Preguntas que vinieron apx.docx”

1.- cuantos errores tiene el siguiente código?

```
public class MyClass{
        static final String s1=”Hola”;
        static final String s2=”mundo”;
        static final String s3=””;
        static{
                s1=”Hello”;
        }
        static{
                s2=”world”;
        }
        public static void main(String argos[]){
                s3=”despues”;
```

```
        }
}
```

a) 0
b) 1
c) 3

respuesta, c, no se pueden modificar finals ya declarados

2.-

```
public class MyClass{
        public static void main(String argos[]){
                String hola=”hola mundo”;
                System.out.println(hola.charAt(11));
        }
}
```

resultado, lanza indexOutOfbounds

3-

```
public class MyClass{
        public static void main(String argos[]){
                int x=10;
                int y=5;
                int z=111;
                switch(z){
                        case 111:
                                System.out.println(“todo bien”);
                        case x:
                                System.out.println(“valor de x”+x);
                                break;
                        case y:
```

```
                    System.out.println("valor de y"+y);
                    break;
                }
            }
}
```

resultado, error de compilacion, solo se pueden poner constantes en un switch

4.- cuales de las siguientes clases tienen un constructor por default?

```
class X{}
class Y{
    public Y(){}
}
class Z{
    public Z(String s){}
}
```

resultado, solo x, aunque y tenga un constructor parecido al default, no cuenta cómo el default

5.- which of the following java operators can be used with boolean variables? (choose all that apply)

a) ==
b) +
c) --
d) !
e) %
f) <=

respuesta, a y d, a es un operador de igualdad el cual puede aceptar desde primitivos hasta objetos y d es exclusivo de booleanos

6.- Which two will compile, and can be run successfully using the following command?

java fred1 hello walls

a) class Fred1{public static void main(String argos){System.out.println(args[1])}}

b) class Fred1{public static void main(String[] args){System.out.println(args)}}
c) class Fred1{public static void main(String[] args){System.out.println(args[1])}}
d) abstract class Fred1{public static void main(String[] argos){System.out.println(args[2])}}

respuesta, b y c, b imprimiria la direccion de memoria de args, mientras que c imprimiria walls

6.-Which three implementations are valid?

```
interface SampleCloseable{

        public void close() throws java.io.IOException;

}
```

a) class Test implements SampleCloseable{public void close() throws java.io.IOException{//do something}}
b) class Test implements SampleCloseable{public void close() throws Exception{//do something}}
c) class Test implements SampleCloseable{public void close() throws FileNotFoundException{//do something}}
d) class Test extends SampleCloseable{public void close() throws java.io.IOException{//do something}}
e) class Test implements SampleCloseable{public void close(){//do something}}

resultado, a, c y e, a porque lanza la misma excepcion, c porque lanza una excepcion hija y e porque no lanza ninguna

7.- Which two possible outputs?

```
public class Main{

        public static void main(String[] args) throws Exception{

                doSomething();

        }

        private static void doSomething() throws Exception{

                System.out.println(“before if clause”);

                if(Math.random()>0.5){throw new Exception();}

                System.out.println(“after if clause”);

        }

}
```

a) before if clause exception in thread main java.lang.exception at main.doSomething (Main:java21) at main.main (main: java:15)
b) before if clause exception in thread main java.lang.exception at main.doSomething (Main:java21) at main.main (main: java:15) after if clause
c) exception in thread main java.lang.exception at main.doSomething (Main:java21) at main.main (main: java:15)
d) before if clause after if clause

respuesta, a y d, solo llegaria al primer print si es que lanza la excepcion, mientras que d es lo que pasaria si no lanza ninguna excepcion

8.- What is the result?

```
import java.text.*;

public class Align{

	public static void main(String[] argos) throws ParseException{

		String[] sa={“111.234”,”222.5678”};

		NumberFormat nf=NumberFormat.getInstance();

		nf.setMaximumFractionDigits(3);

		for(String s:sa){

			System.out.println(nf.parse(sa));

		}

	}

}
```

a) 111.234 222.5678
b) an exception is thrown at runtime
c) 111.234 222.568
d) 111.234 222.567

respuesta, a, parse solo transforma la string en numero, no la formatea

9.-

Given:

```
class MySort implements Comparator<Integer>{

	public int compare(Integer x,Integer y){

		return y.compareTo(x);
```

```
        }
}
```

and the code fragment:

```
Integer[] primes={2,7,5,3}

MySort ms=new MySort();

Arrays.sort(primes, ms);

for(Integer p2:primes){System.out.print(p2+" ");}
```

what is the result?

a) 2 3 5 7
b) 2 7 5 3
c) 7 5 3 2
d) compilation fails

respuesta, c, el sort solo cambia de forma ascendente a descendente

10.-

Given

```
public static void main(String[] args){
        String[] table={"aa","bb","cc"};
        int ii=0;
        for(String ss:table){
                while(ii<table.length){
                        System.out.println(ii);ii++;
                        break;
                }
        }
}
```

how many times is 2 printed?

a) zero
b) once
c) twice

d) thrice
e) it is not printed because compilation fails

resultado, b, solo se imprime una vez 2, pero se imprime 0, 1 y 2

11.-

What is the result?

```
int a=10;

int b=37;

int z=0;

int w=0;

if(a==b){z=3;} else if (a>b){z=6;}

w=10*z;
```

a) 0
b) 30
c) 60

resultado, a, no entra a ningun if

12.- The following are the complete contents of TestClass.java file. Which packages are automatically imported?

```
class TestClass{

        public static void main(String[] argos){

                System.out.println(“hello”);

        }

}
```

a) java.util
b) System
c) java.lang
d) java.io
e) String
f) the package with no name

respuesta, c y f, java.lang se importa automaticamente en todos los programas por default y el paquete sin nombre se importa tambien automaticamente si un programa no tiene un package definido

13.-

```
public static void main(String[] args){
        System.out.println(“hola mundo”);
        int [] arr = new int[];//1
        int [][] d = new int[];//2
        int a[][]=new int[7][6];//3
        Object [][][] objects=new Object[4][0][5];//4
}
```

respuesta, 1 y 2 no compilan, se necesita que tengan la longitud o que esten declarados

14.-

```
public static void main(String[] args) {
        int a=0;
        System.out.println(a++ + 2);
        System.out.println(a);
}
```

en el primer print a la variable “a” se le suma 2 e imprime 2 después de hacer ese proceso

a la variable “a” se incrementa en 1 entonces en el segundo print imprimer 1

Resultado 2 1

15.- Que se va a imprimir al ejecutar el método main()

```
public class Spider {
        void spider(int a){
                System.out.print("Spider");
        }
}
public class Genero {
```

```
	void genero(double b){
		System.out.print("Genero");
	}
}
public class SpiderMain {
	public static void main(String[] args) {
		Spider spider = new Spider();
		Genero genero = new Genero();
		spider.spider(4);
		genero.genero(9.0);
	}
}
```

resultado, spider genero

16.- Con cuál de línea imprimera Equals

```
//boolean r1 = a.equals(b);
//boolean r2 = a==b;
//boolean r3 = a.equalsIgnoreCase(b);
String a = "Hola";
String b = "hola";
boolean resultado = false;
if (.....){
System.out.println("Equals");
}else {
System.out.println("Not Equals");
}
```

resultado, boolean r3 = a.equalsIgnoreCase(b); ya que ignoraria las mayusculas

17.-

```
import java.util.*;

public class Main{

        public static void main(String[] argos) throws Exception

                short kk=11;

                short ii;

                short jj=0;

                for(ii=kk; ii>6; ii-=1)

                        {jj++;}

                System.out.println("jj="+jj);

        }

}
```

a) compilation fails
b) jj=5
c) jj=6
d) jj=0

resultado, b

18.-

¿Qué se imprime al ejecutar el código?

```
public class EjemploArray {

        public static void main(String[] args) {

                List p= new ArrayList();

                p.add(7);

                p.add(1);

                p.add(5);

                p.add(1);

                p.remove(1);

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

```
        }
}
```

resultado, [7,5,1] quita el primero que encuentre con el valor

19.- which three methods inserted individually at line, will correctly complete class Two (choose three)?

```
class One{
        void foo(){}
}
class Two extends One{
        //insert method here
}
```

a) int foo(){/*insert code here*/}
b) void foo(){/*insert code here*/}
c) public void foo(){/*insert code here*/}
d) private void foo(){/*insert code here*/}
e) protected void foo(){/*insert code here*/}

respuesta, b, c y e, no se puede int porque estaria sobreescribiendo de forma erronea

20.-which two statements are true?

a) an abstract class can implement an interface
b) an interface can be extended by an abstract class
c) an interface cannot be extended by another interface
d) an abstract class cannot be extended by an abstract class
e) an abstract class can be extended by an interface
f) an abstract class can be extended by a concrete class

resultados, a y f

21.- ejemplos de Varargs

public void suma(int...g,int []q){} //no compila porque los varargs debe ir al final de la lista de parametros

public void div(int[], ...int r){} //no compila porque los varargs van después del tipo de dato

```
public int suma(int..d,float...o){
return d+o;
```

}//no compila porque no sabe que es la variable d de los parametros si se desea agregar varargs deben ser tres puntos*/

public void t (int...h, double...t){}//no compila porque solo se permite una vez usar varargs

public void ad(int b,double s, String...a){}// si funciona

22.-

```
public class VaribaleFinal {

	public class Persona extends Carro{

		final String nombre;

		public Persona() {

			Carro carro = new Carro();

			nombre = "andres";

		}

	}

	public class Carro{}

	public static void main(String[] args) {

		Persona persona = new Persona(); // error aqui

	}

}
```

resultado, la clase la toma cómo una anidada de variable final, esta misma hace que no pueda ser referenciada, ya que no es una static y ocupa una instancia del padre para que funcione, si se saca tanto carro cómo persona de la clase funcionaria

23.-

```
public static void main(String[] args) {

	boolean x = false;

	boolean z = false;

	int y= 20;

	System.out.println(x);

	//true false |
```

```
        x = (y!=10) ^ (z!=false);
        System.out.println(x + " " + y + " " + z);
}
```

imprime

false

true 20 false

esto por la impresión rara que tiene antes de la operacion y luego el print final

Problemas del documento ejercicios-propuestos.docx

1.-

```
import java.text.NumberFormat; import java.text.ParseException;
import java.util.Scanner;
import java.util.ArrayList;
import java.util.List;
public class Ejemplos {
        public static void main(String[] args) {
                Animal uno=new Animal();
                Animal dos=new Dog();
                uno.makeSound();
                dos.makeSound();
                Dog tres=(Dog)new Animal();
                tres.makeSound();
        }
}
class Animal {
        void makeSound() {
        System.out.println("Animal sound");
```

```
        }
}
class Dog extends Animal {
        void makeSound() {
                System.out.println("Wau Wau");
        }
}
```

1) Animal sound Wau Wau compilation error

2) Compilation Error

3) Animal sound Wau Wau Animal sound

4) Animal sound

respuesta, 1, no se puede castear a un padre para que sea un hijo, se puede castear a un hijo para que sea padre

2.-

```
import java.text.NumberFormat; import java.text.ParseException;
import java.util.Scanner;
import java.util.ArrayList;
import java.util.List;
import java.lang.*;
public class Ejemplos {
        public static void main(String[] args) {
                Cambios uno=new Cambios();
                int x=1;
                String hola="hola";
                StringBuilder hola2=new StringBuilder("hola2");
                Integer x2=4;
                uno.makeSound(x, hola);
```

```
                uno.makeSound(x2, hola2);
                System.out.println("Cambios?: "+x+","+hola+","+x2+","+hola2);
        }
}
class Cambios{
        void makeSound(int x, String s) {
                s="cambiando string";
                x=5;
        }
        void makeSound(Integer x,StringBuilder s) {
                x=9;
                s=s.delete(0,s.length());
        }
}
```

1) Compilation error

2) Cambios?: 1,hola,4,

3) Cambios?: 1,hola,4,hola2

4) Cambios?: 5,cambiando string,9,

respuesta, 2, string builder es el unico mutable que fue afectado

3.-

```
interface i1{
        public void m1();
}
interface i2 extends i1 {
        public void m2();
```

```
}
class animal implements i1,i2 {
//¿Qué métodos debería implementar la clase animal en este espacio?
}
```

1) solo m1

2) m1 y m2

3) ninguno

4) error compilación

respuesta, 2, ambos deben ser implementados

4.-

```
public class Main {
	public static void main(String[] args) {
		Padre objetoPadre = new Padre();
		Hija objetoHija = new Hija();
		Padre objetoHija2 = (Padre) new Hija();
		objetoPadre.llamarClase();
		objetoHija.llamarClase();
		objetoHija2.llamarClase();
		Hija objetoHija3 = (Hija) new Padre();
		objetoHija3.llamarClase();
	}
}
public class Hija extends Padre {
	public Hija() {
		// Constructor de la clase Hija
	}
```

```
	@Override
	public void llamarClase() {
		System.out.println("Llame a la clase Hija");
	}
}
public class Padre {
	public Padre() {
		// Constructor de la clase Padre
	}
	public void llamarClase() {
		System.out.println("Llame a la clase Padre");
	}
}
```

resultado

a) Llame a la clase Padre

Llame a la clase Hija

Llame a la clase Hija

Error: java.lang.ClassCastException

b) Llame a la clase Padre

Llame a la clase Hija

Llame a la clase Hija

Llame a la clase Hija

c) Llame a la clase Padre

Llame a la clase Hija

Llame a la clase Hija

Llame a la clase Padre

resultado, a, las primeras dos son normales, la tercera castea cómo padre pero sigue siendo hija, la cuarta causa excepcion ya que un hijo no puede ser casteado cómo padre

5.-

```
interface Movable {
	void move();
}
abstract class Vehicle {
	abstract void fuel();
}
class Car extends Vehicle implements Movable {
	void fuel() {
		System.out.println("Car is refueled");
	}
	public void move() {
		System.out.println("Car is moving");
		}
	}
public class Main {
	public static void main(String[] args) {
		Vehicle myCar = new Car();
		myCar.fuel();
		((Movable) myCar).move();
	}
}
```

resultado, imprime

car is refueled

car is moving

6.-

```
class Animal {
	void makeSound() throws Exception {
		System.out.println("Animal makes a sound");
		}
	}
class Dog extends Animal {
	void makeSound() throws RuntimeException {
		System.out.println("Dog barks");
	}
}
public class Main {
	public static void main(String[] args) {
		Animal myDog = new Dog();
		try {
			myDog.makeSound();
		}
		catch (Exception e) {
			System.out.println("Exception caught");
		}
	}
}
```

Cuál sería la salida en consola al ejecutar este código?

1- Dog barks

2- Animal makes a sound

3- Exception caught

4- Compilation error

respuesta, 1, solo imprime dog barks

7.-

```
import java.util.*;

import java.lang.*;

import java.io.*;

class Main {

	public static void main(String[] args) {

		String str = "1a2b3c4d5e6f";

		String []splitStr = str.split("//D");

		for(String elemento : splitStr){

			System.out.println(elemento);

		}

	}

}
```

resultado, imprime la misma string pero en el array splitStr, ya que ocupa una regex pero la regex dada no hace nada