**Algorithm for The Glass House problem**

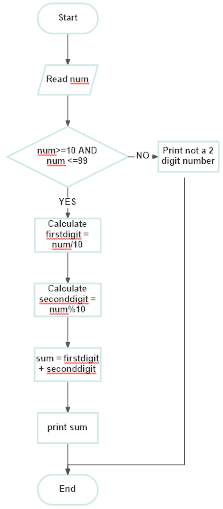
1. Declare Variables to store the first and second digit of the number.
2. Initialize standard input stream System.in for taking inputs.
3. Accept two digit number as input.
4. Check if the number between 10 to 99 i.e. of 2 digits. Else goto step 9.
5. Initialize the first digit of number by dividing the number by 10.
6. Initialize the second digit of number by getting the remainder, number mod 10.
7. Assign the sum of first digit and second digit to sum.
8. Print sum, goto step 10.
9. Print not a 2 digit number.
10. Stop.

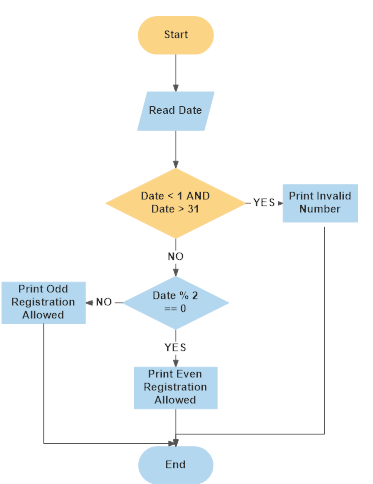
**Algorithm for The Traffic Congestion problem**

1. Initialize standard input stream System.in for taking inputs.
2. Accept the date value as input.
3. Check if the Date is valid i.e. between 1 to 31. Else goto Step 7
4. Check if the date is divisible by 2 i.e. if remainder is 0. Else goto step 6
5. Print even registered cars are permitted today , Goto step 8.
6. Print odd registered cars are permitted today, Goto step 8.
7. Print invalid input.
8. Stop.

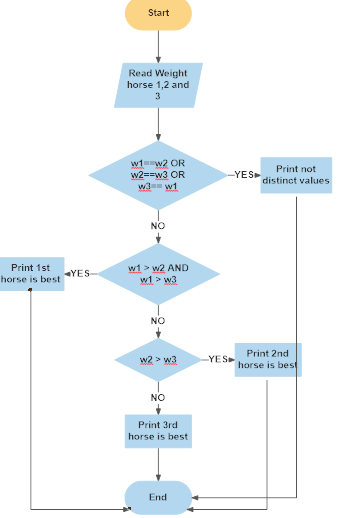
**Algorithm for Choosing the best Horse problem**

1. Initialize standard input stream System.in for taking inputs.
2. Accept the weight of the three horses as input.
3. Check if any two horses have same weights, Else goto step 5.
4. Print weights are not distinct values , goto step 10
5. Check if the weight of first horse is greater than both the other horses, Else goto step 7.
6. Print First horse is best, goto step 10.
7. Check if the weight of second horse is greater than weight of the third horse, Else goto step 9.
8. Print second horse is best, goto step 10.
9. Print Third horse is best.
10. Stop

**Flowchart for The Glass House problem**

**Flowchart for The Traffic Congestion problem**

**Algorithm for Choosing the best Horse problem**

****

**JAVA Code Conventions**

Java coding convention is a rule to follow as you decide what to name your identifiers such as class, package, variable, constant, method, etc.

But, it is not forced to follow. So, it is known as convention not rule. These conventions are suggested by several Java communities such as Sun Microsystems and Netscape.

All the classes, interfaces, packages, methods and fields of Java programming language are given according to the Java naming convention. If you fail to follow these conventions, it may generate confusion or erroneous code.

**Camel case in Java Programming :** It consists of compound words or phrases such that each word or abbreviation begins with a capital letter or first word with a lowercase letter, rest all with capital.

1. **Classes and Interfaces** :
   * Class names should be **nouns**, in mixed case with the **first** letter of each internal word capitalized. Interfaces name should also be capitalized just like class names.
   * Use whole words and must avoid acronyms and abbreviations.

Examples:

*interface Bicycle*

*class MountainBike implements Bicyle*

*interface Sport*

*class Football implements Sport*

1. **Methods :** 
   * Methods should be **verbs**, in mixed case with the **first letter lowercase** and with the first letter of each internal word capitalised.

Examples:

*void changeGear(int newValue);*

*void speedUp(int increment);*

*void applyBrakes(int decrement);*

1. **Variables :** Variable names should be short yet meaningful.
   * Should **not** start with underscore(‘\_’) or dollar sign ‘$’ characters.
   * Should be mnemonic i.e, designed to indicate to the casual observer the intent of its use.
   * **One-character variable names should be avoided** except for temporary variables.
   * Common names for temporary variables are i, j, k, m, and n for integers; c, d, and e for characters.

Examples:

// variables for MountainBike class

*int speed = 0;*

*int gear = 1;*

1. **Constant variables:** 
   * Should be **all uppercase** with words separated by underscores (“\_”).
   * There are various constants used in predefined classes like Float, Long, String etc.

Examples:

*static final int MIN\_WIDTH = 4;*

// Some Constant variables used in predefined Float *class*

*public static final float POSITIVE\_INFINITY = 1.0f / 0.0f;*

*public static final float NEGATIVE\_INFINITY = -1.0f / 0.0f;*

public static final float NaN = 0.0f / 0.0f;

1. **Packages:** 
   * The prefix of a unique package name is always written in **all-lowercase ASCII letters** and should be one of the top-level domain names, like com, edu, gov, mil, net, org.
   * Subsequent components of the package name vary according to an organisation’s own internal naming conventions.

Examples:

*com.sun.eng*

*com.apple.quicktime.v2*

*// java.lang packet in JDK*

*java.lang*