# Draw flowchart and implement the solution in any programming language.

Problem No. 1

*Add two given two digit numbers as long as their total is still two digits only.*

**Requirement gathering and analysis-**

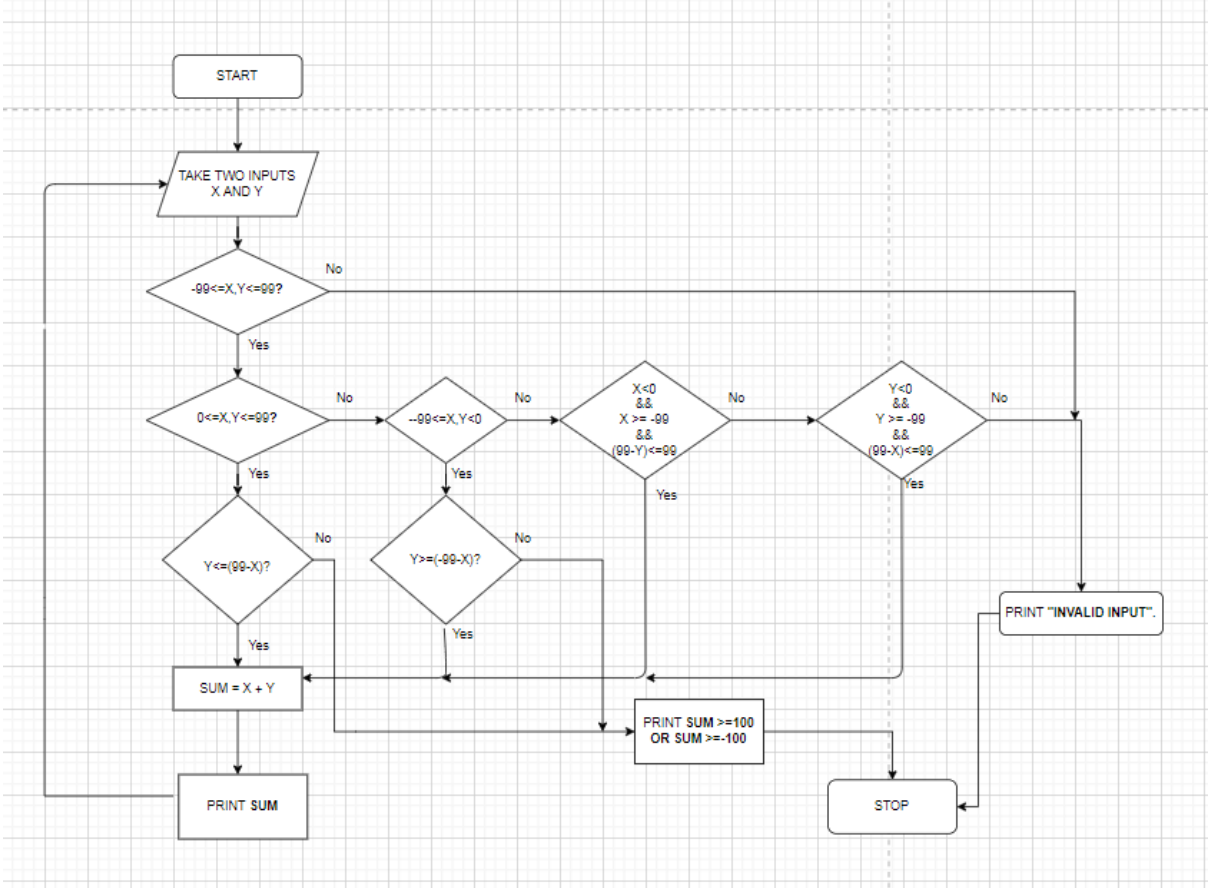
Constraints and conditions :

1) Both the numbers are two digit only.
2) Their sum should be  two digit only.
3) Suppose two numbers and sum are X , Y and S respectively. then  -99<=X,Y,S<=99.

**Approach Or Design :**

1) Take two inputs from the user. Check the constraints and conditions satisfied or not.
2) Check until and unless sum is of two digit only.
3) Finally output the results.

**Flowchart :**




**Solution Implementation :**

```
import java.util.*;

public class Main{

   public static void main(String args[]){
      Scanner input = new Scanner(System.in);
      while(true){
         int X = input.nextInt();
         int Y = input.nextInt();
         if(X>=-99 && X<=99 && Y>=-99 && Y <= 99){
            if(X>=0 && X<=99 && Y>=0 && Y<=99){
               if(Y<=(99-X)){
               System.out.println(X+Y);
               continue;
               }
               else
               {
                  System.out.println("Sum is more than two digit...");
                  break;
               }
            }
            else if (X<0 && X>=-99 && Y<0 && Y>=-99)
            {
               if(Y>=(-99-X))
               {
                  System.out.println(X+Y);
                  continue;
               }
               else{
                  System.out.println("Sum is more than two digit...");
                  break;
               }
            }
            else if(X<0 && X>=-99 && (99-Y)<=99){
               System.out.println(X+Y);
               continue;
            }
            else if(Y<0 && Y>=-99 && (99-X)<=-99){
               System.out.println(X+Y);
               continue;
            }
            else
            {
               System.out.println("Sum is more than two digit...");
               break;
            }
         }
```

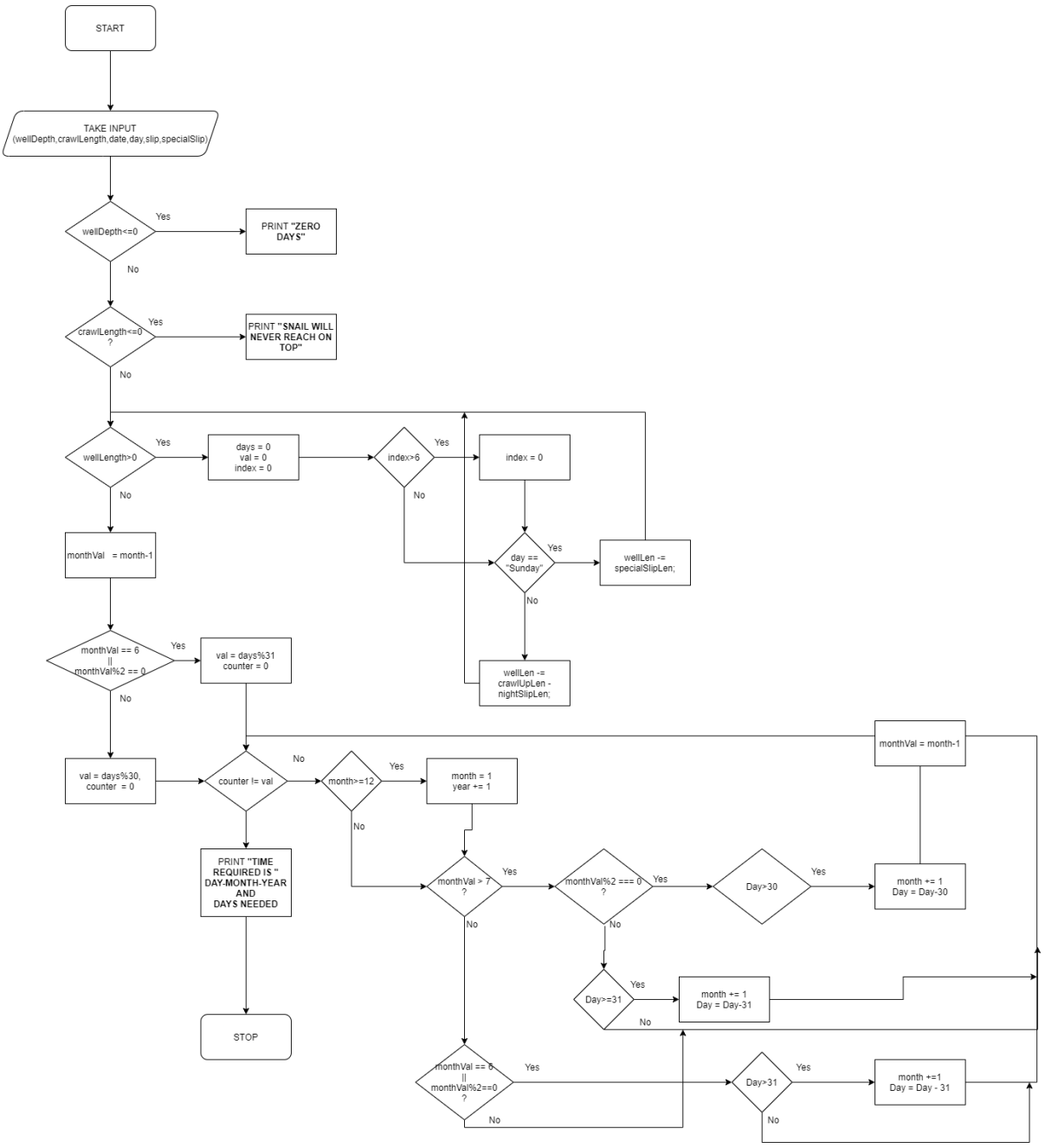
```
        else
        {
           System.out.println("Inputs are invalid");
           break;
        }

     }
  }
}
```

Problem No. 2

*A snail is inside a well of 22 ft. it crawls up 4 feet everyday during day.*
*And while it is sleeping in the night, it slips by 2 feet. On a Sunday it does not crawl up but simply rests where it is. However, while it sleeps, on Sunday night, it slips by 1 foot.*

***Requirement Gathering and analysis:***
***Constraints and conditions -***
*Given inputs -*
*Start Date,Start Day,Well depth,crawl up length, nightSlip, specialNightSlip)*

*Flowchart :*




***Implementation in java***

```
import java.util.*;

public class Main{
    String Answer = "";
    String arr[]   = {"sunday","monday","tuesday","wednesday","thursday","friday","saturday"};
```

```
   String months[]  =
{"january","february","march","april","may","june","july","august","september","october","nove
mber","december"};

   int index = 0;
   int days = 0;
   private  String timeRequired(String day, int wellDepth,int crawlUpLen,int nightSlipLen,int
specialSlipLen,String date){
      if(wellDepth<=0 )
      return Integer.toString(days)+Answer;
      if(crawlUpLen <=0)
      return "Snail will never reach to the top of well!";

      int wellLen = wellDepth;
      while(wellLen>0){
         days++;
         if(index >6)
         index = 0;
         if(arr[index++].equals("sunday")){
            wellLen -= specialSlipLen;
         }
         else{
            wellLen -= crawlUpLen - nightSlipLen;
         }
      }
      Answer += arr[index-1];
      String arr[]  = date.split("-");
      int Day  = Integer.parseInt(arr[0]);
      int month = Integer.parseInt(arr[1]);
      int year  =Integer.parseInt(arr[2]);
      int val  = 0;
      int monthVal = month-1;
      System.out.println(Day+" : "+month+" :"+year);

         if(monthVal == 6 || (monthVal)%2 == 0)
            val = days%31;
         else
            val = days%30;
         Day += days;
         for(int i = 0;i<=val;i++){

            if(month>=12)
            {
               month = 1;
               year += 1;
            }
               if(monthVal>7)
               {
```

```
                if((monthVal)%2 == 0)
                {
                    if(Day>30){
                        month+=1;
                        Day=Day-30;
                    }
                }
                else{
                    if(Day>=31)
                    {
                        month+=1;
                        Day=Day-31;
                    }
                }

            }
            else{
                if(monthVal == 6 || (monthVal)%2 == 0){
                  if(Day>31){
                  month+=1;
                  Day = Day-31;
                  }
                }
                else{
                  if(Day>=30){
                  month+=1;
                  Day = Day - 30;
                  }
                }
            }
          monthVal = month-1;
        }

      String finalAns =
Integer.toString(Day)+"/"+Integer.toString(month)+"/"+Integer.toString(year);

      return "final Date and Day is  "+finalAns+", "+Answer+" AND Days required
"+Integer.toString(days);
   }


   public static void main(String args[]){
      Scanner input = new Scanner(System.in);
      System.out.println("Enter date (note:enter null otherwise) ");
      String date = input.next();
      System.out.println("Enter day (note: enter 'monday' otherwise) ");
      String day = input.next();
      day = day.toLowerCase();
```

```
        System.out.println("Enter wellDepth ");
        int wellDepth = input.nextInt();
        System.out.println("Enter crawlUpLen ");
        int crawlUpLen = input.nextInt();
        System.out.println("Enter nightSlipLen ");
        int nightSlipLen = input.nextInt();
        System.out.println("Enter specialSlipLen ");
        int specialSlipLen = input.nextInt();
        Main main = new Main();
        String Ans =
main.timeRequired(day,wellDepth,crawlUpLen,nightSlipLen,specialSlipLen,date);
        System.out.println(Ans);

    }
}
```