```
/*code by Satyam Pandey*/
//I have provided comments whereever necessary to decipher my code, the
code is executable
import java.util.*;
import java.lang.*;
import java.io.*;
import java.text.ParseException;
import java.text.SimpleDateFormat;

/* Here object of Location(village) Class will have a registered id which
can be recognized by controlling station, the id will be specified by the
user */
class Location {
    int identity;
    int rating;
    Location(int id, int rat) {
      identity=id;//Registered Id
      rating=rat;//in KVA
   }

   public static void main(String[] args) throws IOException,
ParseException{
       BufferedReader inp = new BufferedReader(new
InputStreamReader(System.in));
       //input reg id and rating of transformer
           int id= Integer.parseInt(inp.readLine());
           int rat= Integer.parseInt(inp.readLine());
        /*for request to be valid here I am taking a condition for the
sake of understanding that id>=1&&id<=20000, we have only 20000 villages
currently registered*/
        System.out.println("Generate Request for Location ID: "+id+" for
procurement of transformer of "+rat+" KVA rating");
        Location a = new Location(id,rat);
        String reason=inp.readLine();//to Input your reason for
transformer amongst three categories storm, fault, relayfail, burnout
only if anything else will be given it will not be accepted
        System.out.println("Given Reason: "+reason);
        TransProc b = new TransProc(id,rat,reason);// Creating object of
Transformer Procurement authority class
        if(b.validregid()==true&&b.validreason()==true)
        {
            System.out.println("Your request for transformer procurement
is Valid and shall be processed soon");
        }
        else
        {
            System.out.println("Sorry! Our team found your request as
invalid and it will not be processed as of now");
            System.out.println("Now you need to seek approval from SDM,
please enter the date when last time transformer was installed in your
location: ");
            String datein=inp.readLine();//in dd/mm/yyyy format only
            Date datefor=new
SimpleDateFormat("dd/MM/yyyy").parse(datein);
            System.out.println(datein+"\t"+datefor);  //in GMT


            AdminApprov c= new AdminApprov(datein);
```

```
            if(c.Canapprove()==true)
            {
                System.out.println("Yes, Ministry has approved your
request because it was installed last time quite back, so your request
will be processed now!!");
            }
            else
            System.out.println("Sorry! Your request cant be processed
further as it is also rejected by ministry since it was installed within
2 years back only");

        }

   }
}
class TransProc {
    int regid;
    int reqrating;
    String reason;
   TransProc(int id, int reqrat, String reas) {
       regid=id;
       reqrating=reqrat;
       reason=reas;
      System.out.println("Request for Transformer Procurement Received,
Controlling Authority to attend the request for the following
specifications:\n"+"Location Reg ID: "+regid+"\n"+"Transformer Rating in
KVA: "+reqrating+"\n"+"Reason given by Location Incharge: "+reason);
   }
   boolean validregid()
   {

       if(regid>=1&&regid<=20000)
       {
           return true;
       }
       else
       return false;

   }
   boolean validreason()
   {

if(reason.equals("storm")||reason.equals("relayfail")||reason.equals("bur
nout")||reason.equals("fault")||reason.equals("naturalhazard"))
       return true;
       else
       return false;
   }

}
class AdminApprov {
    String start;
    String end;
    Date datestart;
    Date dateend;
    String dateinn;
    Date datein1;
    AdminApprov(String datein) throws ParseException
```

```
    {
        start="11/10/2007";
         end="11/10/2018";
        datestart=new SimpleDateFormat("dd/MM/yyyy").parse(start);
        dateend=new SimpleDateFormat("dd/MM/yyyy").parse(end);

        dateinn=datein;
        datein1=new SimpleDateFormat("dd/MM/yyyy").parse(dateinn);
    }
    boolean Canapprove()
    {
        if(datein1.after(datestart) && datein1.before(dateend))
        return true;
        else
        return false;
    }


}
//end of program
```