1. String

Write a Java program to replace each substring of a given string that matches the given regular expression with the given replacement.

Ans.

```
import java.util.Scanner;

public class Autest1 {
    public static String replace(String s1, String match, String
replace){
        StringBuilder s= new StringBuilder();
        String k[]=s1.split(" ");
        for(int i=0;i<k.length;i++){
            if(k[i].equals(match))
                s.append(replace);
            else
                s.append(k[i]);
            s.append(" ");
        }
        String m= s.toString();
        if(m.charAt(m.length()-1)==' '){
            return m.substring(0,m.length()-1);
        }
        else
            return m;
    }
    public static void main(String[] args) {
        Scanner s= new Scanner(System.in);
        String s1=s.nextLine();
        System.out.println("a");
        String match=s.nextLine();
        String replace=s.nextLine();
        System.out.println(replace(s1,match,replace));
    }
}
```

2. Collection

Write a Java program to get a reverse order view of the keys contained in a given map

Ans.

```
import java.util.*;


public class Autest2 {
    public static <K extends Comparable,V> Map<K,V>
sortbykey(Map<K,V> a){
        Map<K,V> treemap= new
TreeMap<>(Collections.reverseOrder());
        treemap.putAll(a);
        return treemap;
    }
    public static void main(String[] args) {
        Map<String,String> h= new TreeMap<>();
        h.put("Ashish","1");
        h.put("Chetan","2");
        h.put("Bowmkcik","1");
        h.put("Zeera","1");
        h=sortbykey(h);
        System.out.println(h);
    }
}
```

3 Exception

Write your own unchecked Exception and throw it from you counter programme which counts 1 to 100.When you get Prime no while counting then throw this Exception and catch this to print you exception message.

Ans.

```
public class PrimeNumberException extends Exception{

    PrimeNumberException(String a){
      super(a);
    }

}

public class Autest3 {
    public static void main(String[] args) {
        int n=100;
        for(int i=1;i<=n;i++){
            try{
                if (primary(i)){
                    throw new
PrimeNumberException(Integer.toString(i)+" Prime number exception
occured");
                }
            }
            catch (Exception e){
                System.out.println(e.getMessage());
            }
        }

    }
    public static boolean primary(int n){
       if(n<=1)
           return false;
       if(n<=3)
           return true;
       if(n%2==0 || n%3==0)
           return false;
       for(int i=5 ; i*i<=n;i=i+6){
           if(n%i==0 || n%(i+2)==0)
               return false;
       }
       return true;
    }
}
```

4. Serialization

Write a programme to serialize 3 fields out of 5 and deserialize it. Use UUID to prvent object mutation.

Ans.

```
import java.io.*;
import java.util.UUID;

public class School implements Serializable {
    private static final UUID SerialVerisonUID= UUID.randomUUID();
    public int rollno;
    public String name;
    public int classn;
    public transient char section;
    public transient int grade;
}


public class Serializingdata {
    public static void main(String[] args) {
        School s=new School();
        s.rollno=1;
        s.name="arpit";
        s.classn=5;
        s.section='a';
        s.grade=9;
        try{
            FileOutputStream fileout= new FileOutputStream("/home/
arpit/Desktop/school.txt");
            ObjectOutputStream out = new
ObjectOutputStream(fileout);
            out.writeObject(s);
            out.close();
            fileout.close();
            System.out.println("data saved");
        }
        catch (Exception e){
            e.printStackTrace();
        }
    }
}




public class Deserializing {
    public static void main(String[] args) throws
IOException,ClassNotFoundException{
        School s =null;
        try{
            FileInputStream file=new FileInputStream("/home/arpit/
Desktop/school.txt");
            ObjectInputStream obj=new ObjectInputStream(file);
            s=(School) obj.readObject();
            obj.close();
            file.close();
        }
        finally {
```

```
            System.out.println(s.name);
            System.out.println(s.rollno);
            System.out.println(s.classn);
            System.out.println(s.section);
            System.out.println(s.grade);
        }

    }
}
```

A. Print American Flag

```
public class AuClass1 {
    public static void main(String[] args) {
        for(int i=0;i<16;i++){
            if(i<9){
                for(int j=0;j<40;j++){
                    if(j<11){
                        if(i%2==0 && j%2==0){
                            System.out.print("* ");
                        }
                        else if(i%2!=0 && j%2!=0){
                            System.out.print("* ");
                        }
                        else{
                            System.out.print("  ");
                        }
                    }
                    else{
                        if(i%2==0){
                            System.out.print("r ");
                        }
                        else
                            System.out.print("w ");
                    }
                }
            }
            else{
                for(int j=0;j<40;j++){
                    if(i%2==0){
                        System.out.print("r ");
                    }
                    else
                        System.out.print("w ");
                }
            }
            System.out.println();
        }
    }
}
```

B. Evaluate Expression

Ans.

```
import java.util.Stack;

public class AuClass2 {
    public static int calculate(String s){
        char[] tokens= s.toCharArray();
        Stack<Integer> values= new Stack<Integer>();
        Stack<Character> ops= new Stack<Character>();
        for(int i=0;i<tokens.length;i++){
            if(tokens[i]==' ')
                continue;
            if(tokens[i]>='0' && tokens[i]<='9'){
                StringBuffer sbuf= new StringBuffer();
                while(i<tokens.length && tokens[i]>='0' &&
tokens[i]<='9')
                    sbuf.append(tokens[i++]);
                values.push(Integer.parseInt(sbuf.toString()));
            }
            else if (tokens[i]=='(')
                ops.push(tokens[i]);
            else if (tokens[i]==')'){
                while(ops.peek()!='(')

values.push(applyOp(ops.pop(),values.pop(),values.pop()));
                ops.pop();
            }
            else if (tokens[i]=='+' || tokens[i]=='-' ||
tokens[i]=='*'|| tokens[i]=='/' || tokens[i]=='%')
            {
                while(!ops.empty() &&
hasPrecedence(tokens[i],ops.peek()))

values.push(applyOp(ops.pop(),values.pop(),values.pop()));
                ops.push(tokens[i]);
            }
        }
        while(!ops.empty())

values.push(applyOp(ops.pop(),values.pop(),values.pop()));
        return values.pop();
    }
    public static int applyOp(char op , int a, int b){
        switch (op){
            case '+':
                return a+b;
            case '-':
                return a-b;
            case '*':
                return a*b;
            case '/':
                return a/b;
            case '%':
                return a%b;


        }
        return 0;
```

```
    }
    public static boolean hasPrecedence(char op1, char op2){
        if(op2=='(' || op2==')')
            return false;
        if((op1=='*'|| op1=='/') && (op2=='+' || op2=='-'))
            return false;
        else
            return true;
    }


}
```