

```

class Ex1Baril {
    // pour stocker les séquences binaires
    static String[] ts;
    // indice du tableau
    static int z;

    // gen_bin stockant les chaines dans le tableau ts
    public static void tab_bin(int n, String chaine){
        if(n==0) ts[z++] = chaine;
        else {
            tab_bin(n-1,chaine+"0");
            tab_bin(n-1,chaine+"1");
        }
    }

    // affiche un tableau de String
    public static void displayS(String h[]) {
        for (int i = 0 ; i < h.length ; i++)
            System.out.println(h[i]);
    }

    // affiche en ligne le n premières lignes d'un tableau de String
    public static void displaySl(String h[], int n) {
        for (int i = 0 ; i < n ; i++)
            System.out.print(h[i]+" ");
    }

    // test de périodicité
    public static boolean isPer(String a) {
        boolean res = false;
        for (int u=1; u<a.length(); u++)
            if (a.equals(a.substring(u)+a.substring(0,u)))
                res = true;
        return res;
    }

    public static void main(String[] args){
        int l, j;
        int c; //nb mots de Lyndon de chaque taille
        int e; //nb cl eq de chaque taille
        // stocke les mots du cycle universel
        String[] tcs;
        int x = 0; // indice de tcs

        System.out.println("Donner la taille n des séquences");
        int n = Lire.i();
        tcs = new String[(int)Math.pow(2,n)];

        // pour chaque taille inférieure à n
        for (int p=1; p<=n; p++) {
            c = 0; e = 0;
            l = (int)Math.pow(2,p); // nb de mots de taille p
            ts = new String [l];
            z = 0;
            // stocke les séquences binaires de taille p dans ts
            tab_bin(p,"");
            System.out.println("\ntaille "+p+" :");

```

```

        System.out.println("plus petits mots de chq cl eq : ");
        for (int i=0; i<l; i++) { //pour chq mot
            if (!ts[i].equals("")) { // si le mot n'appartient pas encore
à une cl eq
                e++; // on incrémente le nb cl eq
                System.out.println(ts[i]); // on affiche le mot

                if (!isPer(ts[i])) { // s'il n'est pas périodique
                    System.out.println("mot de Lyndon");
                    c++;
                    if (n%p == 0) { // s'il divise n
                        // stockage des mots de lyndon dans tcs pour le
cycle universel
                            tcs[x++] = ts[i];
                        }
                    }
                for (int k=1; k<p; k++) { // pour chq décalage de ce mot
                    // on recherche un mot + grand ds la même cl eq
                    j=i+1;
                    boolean trouve = false;
                    while(j<l && !trouve) {
                        // si un mot + grand est ds la même cl eq
                        if (ts[j].equals(ts[i].substring(k)+ts[i].substring(0,k)))
{
                            ts[j] = ""; // on élimine ce mot
                            trouve = true;
                        }
                        j++;
                    }
                }
            }
            System.out.println("nb de classes d'equivalence : "+e);
            System.out.println("nb de mots de Lyndon : "+x);
        }
        // tri par ordre lexicographique du tableau tcs
        // tri à bulle simpliste
        for (j=0; j<x-1; j++) {
            for (int i=1; i<x; i++) {
                if(tcs[i].compareTo(tcs[i-1]) < 0) {
                    String tmp = tcs[i-1];
                    tcs[i-1] = tcs[i];
                    tcs[i] = tmp;
                }
            }
        }
        System.out.println("\ncycle universel :");
        displaySl(tcs,x);
    }
}

```

```

/*
nb de classes d'equivalence
(séquence A000031 de l'L'Encyclopédie en ligne des suites de nombres
entiers, oeis.org)
n = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13
    1, 2, 3, 4, 6, 8, 14, 20, 36, 60, 108, 188, 352, 632

```

nb de mots de Lyndon (séquence A001037 de l'oeis)
n = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12
1, 2, 1, 2, 3, 6, 9, 18, 30, 56, 99, 186, 335

cycle universel pour n = 4 :
0 0001 0011 01 0111 1
*/