

## SIMPLE SIEVE

```
public class SieveMain{

    public static void simpleSieve(int limit)
    {
        boolean prime[] = new boolean[limit + 1];
        for (int i = 2; i < limit; i++)
        {
            prime[i] = true;
        }

        for(int p = 2; p*p < limit; p++)
        {
            if(prime[p] == true)
            {
                for(int i = p*p; i < limit; i += p)
                {
                    prime[i] = false;
                }
            }
        }

        for(int i = 2; i < limit; i++)
        {
            if(prime[i] == true)
            {
                System.out.print(i + " ");
            }
        }
    }

    public static void main(String[] args)
    {
        simpleSieve(30);
    }
}
```

## SEGMENTED\_SIEVE

```
public class Segmented {
    public static void SegSieve(int l, int h) {
        boolean prime[] = new boolean[h - l + 1];

        for (int p = 2; p * p <= h; p++) {
            int sm = Math.max(p * p, (l + p - 1) / p * p);

            for (int i = sm; i <= h; i += p) {
                prime[i - l] = true;
            }
        }

        for (int i = l; i <= h; i++) {
            if (!prime[i - l] && i > 1) {
                System.out.print(i + " ");
            }
        }
    }

    public static void main(String[] args) {
        SegSieve(10, 30);
    }
}
```

## EULERS PHI ALGO

```
import java.util.*;
```

```

public class EulerPhiAlgorithm
{
    public static int phi(int n)
    {
        int result = n;

        for(int p =2; p*p <=n; p++)
        {
            if(n%p == 0)
            {
                while(n%p == 0)
                {
                    n /= p;
                }
                result -= result/p;
            }
        }

        if(n > 1)
        {
            result -= result/n;
        }

        return result;
    }

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number: ");
        int n = sc.nextInt();
        System.out.println("Euler's Totient Function for " + n + " is " + phi(n));
    }
}

```

CRT

```

import java.util.Scanner;

public class remainder_theorem
{
    public static int findmin(int num[], int rem[], int K)
    {
        int x = 1;
        while(true)
        {
            int j;
            for(j=0; j <K; j++)
            {
                if(x%num[j] != rem[j])
                {
                    break;
                }
            }
            if(j==K)
            {
                return x;
            }
            x++;
        }
    }

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
    }
}

```

```

        //congruence relations
        int size = sc.nextInt();
        int a[] = new int[size];
        int m[] = new int[size];

        //a
        for(int i=0; i <size; i++)
        {
            a[i] = sc.nextInt();
        }

        //m
        for(int i=0; i<size; i++)
        {
            m[i] = sc.nextInt();
        }

        System.out.println(findmin(m,a,size));
    }
}

```

AAT

```

import java.util.*;

public class AliceAppleTree
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int apple = sc.nextInt();
        int count = 0, sum =0;
        while(sum<apple)
        {
            count++;
            sum += (12*count*count);
        }

        System.out.println((8*(count)));
    }
}

```

STROBOGRAMMATIC

```

import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

public class Strobo
{
    public static boolean isStrobogrammatic(String num)
    {
        Map<Character, Character> map = new HashMap<Character, Character>();
        map.put('0', '0');
        map.put('1', '1');
        map.put('6', '9');
        map.put('8', '8');
        map.put('9', '6');

        int l =0;
        int r = num.length() - 1;

        while(l<=r)
        {
            if(!map.containsKey(num.charAt(l)))
            {
                return false;
            }
        }
    }
}

```

```

        if (map.get(num.charAt(l)) != num.charAt(r))
        {
            return false;
        }
        l++;
        r--;
    }

    return true;
}

public static void main(String[] args)
{
    Scanner sc = new Scanner(System.in);
    String n = sc.next();
    System.out.println(isStrobogrammatic(n));
}
}

```

#### TOGGLE SWITCH

```

import java.util.Scanner;

public class Toggle {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();

        boolean[] b = new boolean[n + 1];
        int c = 0, o = 0;

        for (int i = 1; i <= n; i++) {
            for (int j = i; j <= n; j += i) {
                b[j] = !b[j];
            }
        }

        for (int i = 1; i <= n; i++) {
            if (b[i]) {
                c++; //"on"
            } else {
                o++; //"off"
            }
        }
        System.out.println(c + " " + o);
    }
}

```