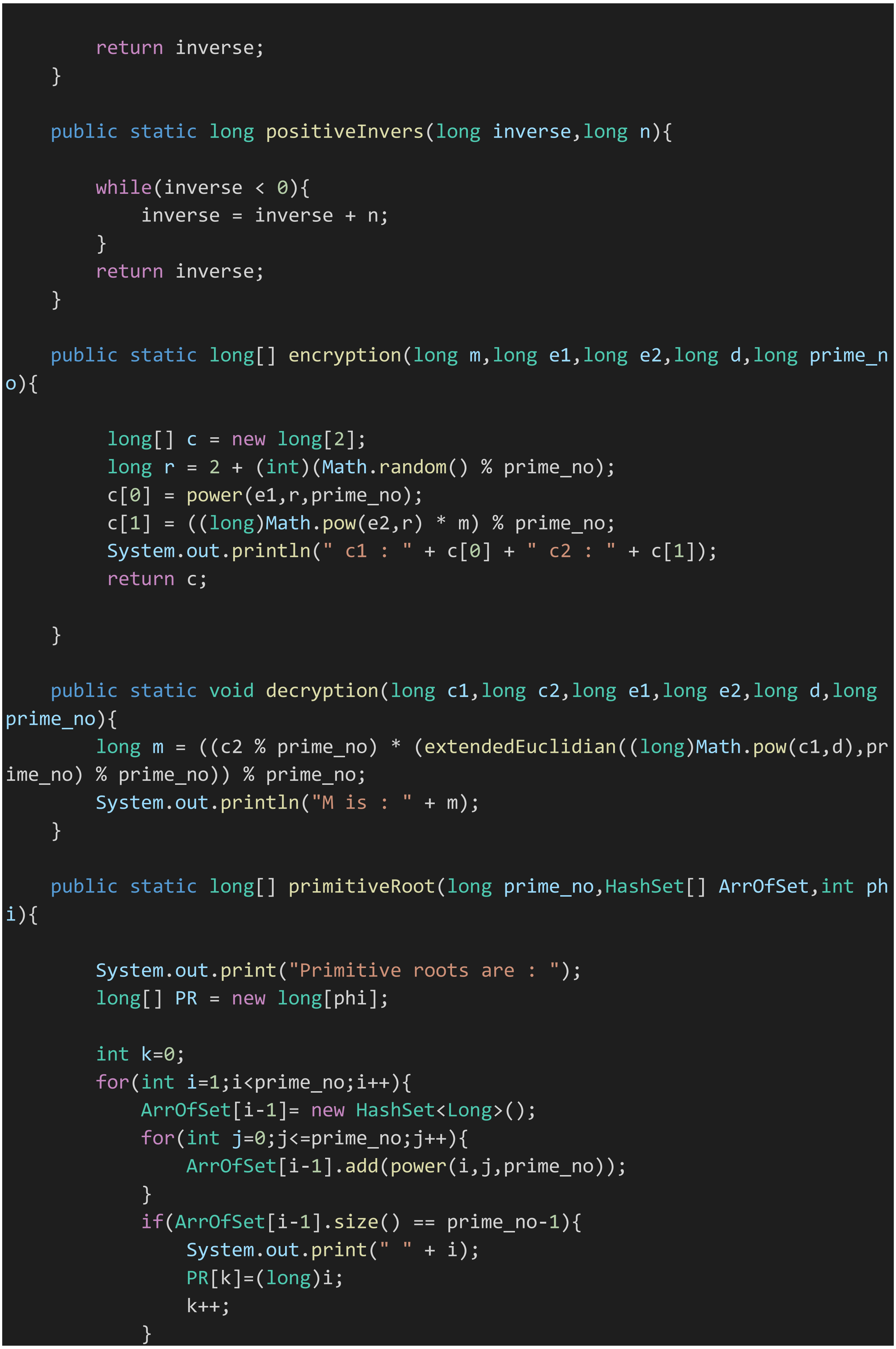
Lab – 5

Subject: NIS

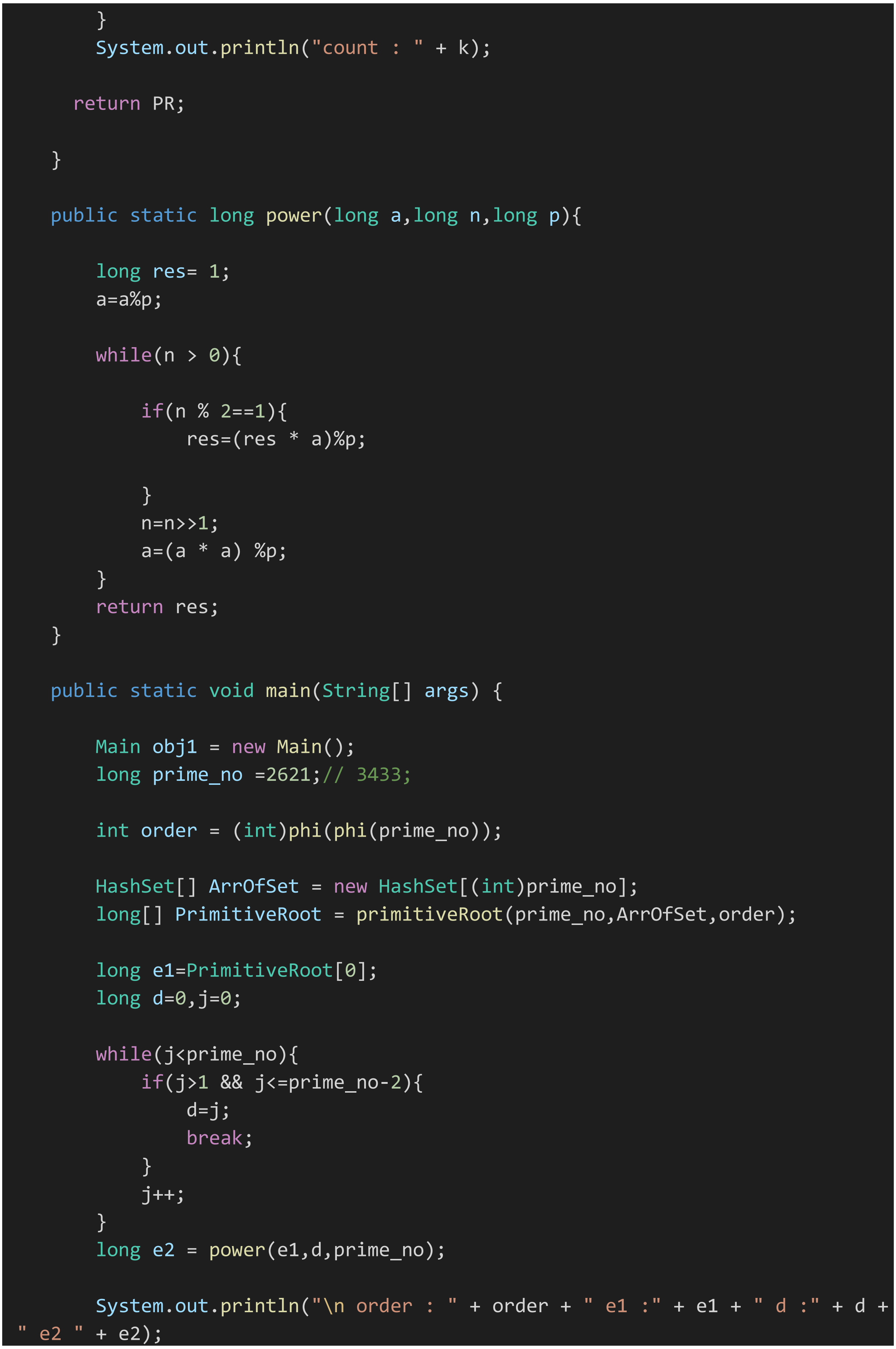
Aim: Write a Program to find the primitive roots for the Multiplicative Group with respect to Prime Modulus. Using that Implement Elgamal Cryptosystem.

Program: -

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

public class Main
{
    public static long phi(long n)
    {
        long result = n;
        for (long i = 2; i * i <= n; i++) {
            if (n % i == 0) {
                while (n % i == 0)
                    n /= i;
                result -= result / i;
            }
        }
        if (n > 1)
            result -= result / n;
        return result;
    }

    public static long extendedEuclidian(long a,long n){
        long[] arr = new long[2];

        long r1=n,r2=a,r,t,t1=0,t2=1,gcd,inverse,q;

        while(r2 > 0){
            q=r1/r2;
            r=r1-q*r2;
            r1=r2;
            r2=r;

            t=t1-q*t2;
            t1=t2;
            t2=t;
        }
        gcd=r1;
        inverse=t1;
        if(inverse < 0){
            inverse = positiveInvers(inverse,n);
        }
```



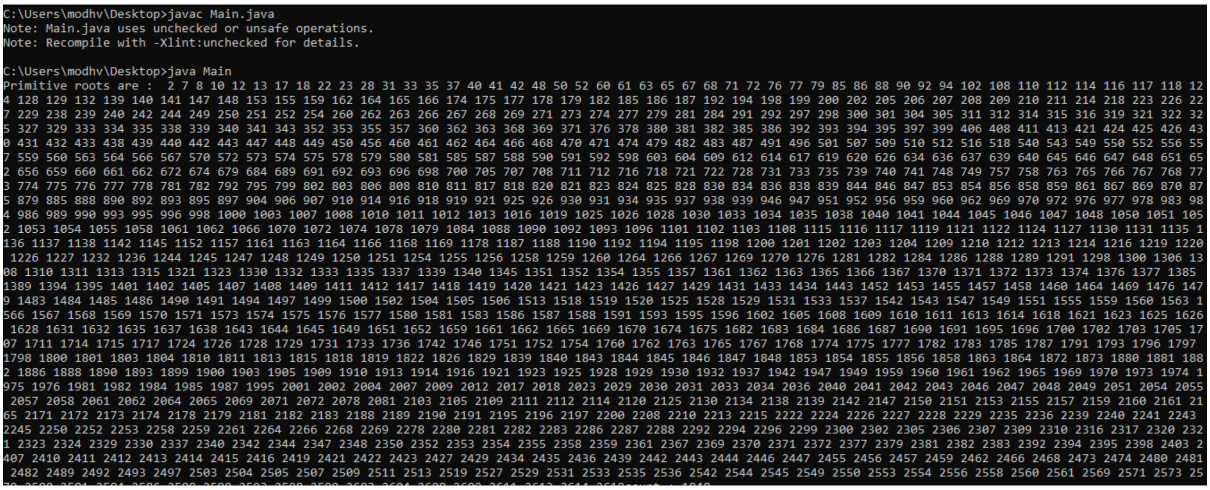
```
        return inverse;
    }

    public static long positiveInvers(long inverse,long n){

        while(inverse < 0){
            inverse = inverse + n;
        }
        return inverse;
    }

    public static long[] encryption(long m,long e1,long e2,long d,long prime_no){

         long[] c = new long[2];
         long r = 2 + (int)(Math.random() % prime_no);
         c[0] = power(e1,r,prime_no);
         c[1] = ((long)Math.pow(e2,r) * m) % prime_no;
         System.out.println(" c1 : " + c[0] + " c2 : " + c[1]);
         return c;

    }

    public static void decryption(long c1,long c2,long e1,long e2,long d,long prime_no){
        long m = ((c2 % prime_no) * (extendedEuclidian((long)Math.pow(c1,d),prime_no) % prime_no)) % prime_no;
        System.out.println("M is : " + m);
    }

    public static long[] primitiveRoot(long prime_no,HashSet[] ArrOfSet,int phi){

        System.out.print("Primitive roots are : ");
        long[] PR = new long[phi];

        int k=0;
        for(int i=1;i<prime_no;i++){
            ArrOfSet[i-1]= new HashSet<Long>();
            for(int j=0;j<=prime_no;j++){
                ArrOfSet[i-1].add(power(i,j,prime_no));
            }
            if(ArrOfSet[i-1].size() == prime_no-1){
                System.out.print(" " + i);
                PR[k]=(long)i;
                k++;
            }
```



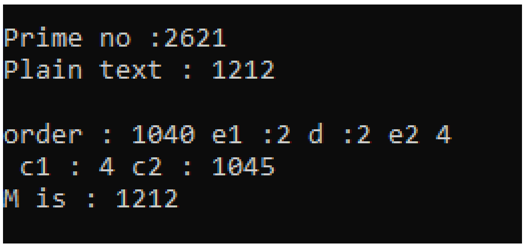
```
        }
        System.out.println("count : " + k);

      return PR;

    }

    public static long power(long a,long n,long p){

        long res= 1;
        a=a%p;

        while(n > 0){

            if(n % 2==1){
                res=(res * a)%p;

            }
            n=n>>1;
            a=(a * a) %p;
        }
        return res;
    }

    public static void main(String[] args) {

        Main obj1 = new Main();
        long prime_no =2621;// 3433;

        int order = (int)phi(phi(prime_no));

        HashSet[] ArrOfSet = new HashSet[(int)prime_no];
        long[] PrimitiveRoot = primitiveRoot(prime_no,ArrOfSet,order);

        long e1=PrimitiveRoot[0];
        long d=0,j=0;

        while(j<prime_no){
            if(j>1 && j<=prime_no-2){
                d=j;
                break;
            }
            j++;
        }
        long e2 = power(e1,d,prime_no);

        System.out.println("\n order : " + order + " e1 :" + e1 + " d :" + d +
 " e2 " + e2);
```

```
        long m = 1212;
        long[] c = encryption(m,e1,e2,d,prime_no);
        decryption(c[0],c[1],e1,e2,d,prime_no);

        //  for(int i=0;i<PrimitiveRoot.length;i++){
        //      System.out.println(PrimitiveRoot[i]);
    //      }

    }
}
```

Output: -



```
C:\Users\modhv\Desktop>javac Main.java
Note: Main.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.

C:\Users\modhv\Desktop>java Main
Primitive roots are :  2 7 8 10 12 13 17 18 22 23 28 31 33 35 37 40 41 42 48 50 52 60 61 63 65 67 68 71 72 76 77 79 85 86 88 90 92 94 102 108 110 112 114 116 117 118 12
4 128 129 132 139 140 141 147 148 153 155 159 162 164 165 166 174 175 177 178 179 182 185 186 187 192 194 198 199 200 202 205 206 207 208 209 210 211 214 218 223 226 22
7 229 238 239 240 242 244 249 250 251 252 254 260 262 263 266 267 268 269 271 273 274 277 279 281 284 291 292 297 298 300 301 304 305 311 312 314 315 316 319 321 322 32
5 327 329 333 334 335 338 339 340 341 343 352 353 355 357 360 362 363 368 369 371 376 378 380 381 382 385 386 392 393 394 395 397 399 406 408 411 413 421 424 425 426 43
0 431 432 433 438 439 440 442 443 447 448 449 450 456 460 461 462 464 466 468 470 471 474 479 482 483 487 491 496 501 507 509 510 512 516 518 540 543 549 550 552 556 55
7 559 560 563 564 566 567 570 572 573 574 575 578 579 580 581 585 587 588 590 591 592 598 603 604 609 612 614 617 619 620 626 634 636 637 639 640 645 646 647 648 651 65
2 656 659 660 661 662 672 674 679 684 689 691 692 693 696 698 700 705 707 708 711 712 716 718 721 722 728 731 733 735 739 740 741 748 749 757 758 763 765 766 767 768 77
3 774 775 776 777 778 781 782 792 795 799 802 803 806 808 810 811 817 818 820 821 823 824 825 828 830 834 836 838 839 844 846 847 853 854 856 858 859 861 867 869 870 87
5 879 885 888 890 892 893 895 897 904 906 907 910 914 916 918 919 921 925 926 930 931 934 935 937 938 939 946 947 951 952 956 959 960 962 969 970 972 976 977 978 983 98
4 986 989 990 993 995 996 998 1000 1003 1007 1008 1010 1011 1012 1013 1016 1019 1025 1026 1028 1030 1033 1034 1035 1038 1040 1041 1044 1045 1046 1047 1048 1050 1051 105
2 1053 1054 1055 1058 1061 1062 1066 1070 1072 1074 1078 1079 1084 1088 1090 1092 1093 1096 1101 1102 1103 1108 1115 1116 1117 1119 1121 1122 1124 1127 1130 1131 1135 1
136 1137 1138 1142 1145 1152 1157 1161 1163 1164 1166 1168 1169 1178 1187 1188 1190 1192 1194 1195 1198 1200 1201 1202 1203 1204 1209 1210 1212 1213 1214 1216 1219 1220
 1226 1227 1232 1236 1244 1245 1247 1248 1249 1250 1251 1254 1255 1256 1258 1259 1260 1264 1266 1267 1269 1270 1276 1281 1282 1284 1286 1288 1289 1291 1298 1300 1306 13
08 1310 1311 1313 1315 1321 1323 1330 1332 1333 1335 1337 1339 1340 1345 1351 1352 1354 1355 1357 1361 1362 1363 1365 1366 1367 1370 1371 1372 1373 1374 1376 1377 1385
1389 1394 1395 1401 1402 1405 1407 1408 1409 1411 1412 1417 1418 1419 1420 1421 1423 1426 1427 1429 1431 1433 1434 1443 1452 1453 1455 1457 1458 1460 1464 1469 1476 147
9 1483 1484 1485 1486 1490 1491 1494 1497 1499 1500 1502 1504 1505 1506 1513 1518 1519 1520 1525 1528 1529 1531 1533 1537 1542 1543 1547 1549 1551 1555 1559 1560 1563 1
566 1567 1568 1569 1570 1571 1573 1574 1575 1576 1577 1580 1581 1583 1586 1587 1588 1591 1593 1595 1596 1602 1605 1608 1609 1610 1611 1613 1614 1618 1621 1623 1625 1626
 1628 1631 1632 1635 1637 1638 1643 1644 1645 1649 1651 1652 1659 1661 1662 1665 1669 1670 1674 1675 1682 1683 1684 1686 1687 1690 1691 1695 1696 1700 1702 1703 1705 17
07 1711 1714 1715 1717 1724 1726 1728 1729 1731 1733 1736 1742 1746 1751 1752 1754 1760 1762 1763 1765 1767 1768 1774 1775 1777 1782 1783 1785 1787 1791 1793 1796 1797
1798 1800 1801 1803 1804 1810 1811 1813 1815 1818 1819 1822 1826 1829 1839 1840 1843 1844 1845 1846 1847 1848 1853 1854 1855 1856 1858 1863 1864 1872 1873 1880 1881 188
2 1886 1888 1890 1893 1899 1900 1903 1905 1909 1910 1913 1914 1916 1921 1923 1925 1928 1929 1930 1932 1937 1942 1947 1949 1959 1960 1961 1962 1965 1969 1970 1973 1974 1
975 1976 1981 1982 1984 1985 1987 1995 2001 2002 2004 2007 2009 2012 2017 2018 2023 2029 2030 2031 2033 2034 2036 2040 2041 2042 2043 2046 2047 2048 2049 2051 2054 2055
 2057 2058 2061 2062 2064 2065 2069 2071 2072 2078 2081 2103 2105 2109 2111 2112 2114 2120 2125 2130 2134 2138 2139 2142 2147 2150 2151 2153 2155 2157 2159 2160 2161 21
65 2171 2172 2173 2174 2178 2179 2181 2182 2183 2188 2189 2190 2191 2195 2196 2197 2200 2208 2210 2213 2215 2222 2224 2226 2227 2228 2229 2235 2236 2239 2240 2241 2243
2245 2250 2252 2253 2258 2259 2261 2264 2266 2268 2269 2278 2280 2281 2282 2283 2286 2287 2288 2292 2294 2296 2299 2300 2302 2305 2306 2307 2309 2310 2316 2317 2320 232
1 2323 2324 2329 2330 2337 2340 2342 2344 2347 2348 2350 2352 2353 2354 2355 2358 2359 2361 2367 2369 2370 2371 2372 2377 2379 2381 2382 2383 2392 2394 2395 2398 2403 2
407 2410 2411 2412 2413 2414 2415 2416 2419 2421 2422 2423 2427 2429 2434 2435 2436 2439 2442 2443 2444 2446 2447 2455 2456 2457 2459 2462 2466 2468 2473 2474 2480 2481
 2482 2489 2492 2493 2497 2503 2504 2505 2507 2509 2511 2513 2519 2527 2529 2531 2533 2535 2536 2542 2544 2545 2549 2550 2553 2554 2556 2558 2560 2561 2569 2571 2573 25
```



```
Prime no :2621
Plain text : 1212

order : 1040 e1 :2 d :2 e2 4
 c1 : 4 c2 : 1045
M is : 1212
```

Primitive roots of 2621: -

Total no of primitive root is(order) : 1040

Primitive roots are : 2 7 8 10 12 13 17 18 22 23 28 31 33 35 37 40 41 42 48 50 52 60 61 63 65 67 68 71 72 76 77 79 85 86 88 90 92 94 102 108 110 112 114 116 117 118 124 128 129 132 139 140 141 147 148 153 155 159 162 164 165 166 174 175 177 178 179 182 185 186 187 192 194 198 199 200 202 205 206 207 208 209 210 211 214 218 223 226 227 229 238 239 240 242 244 249 250 251 252 254 260 262 263 266 267 268 269 271 273 274 277 279 281 284 291 292 297 298 300 301 304 305 311 312 314 315 316 319 321 322 325 327 329 333 334 335 338 339 340 341 343 352 353 355 357 360 362 363 368 369 371 376 378 380 381 382 385 386 392 393 394 395 397 399 406 408 411 413 421 424 425 426 430 431 432 433 438 439 440 442 443 447 448 449 450 456 460 461 462 464 466 468 470 471 474 479 482 483 487 491 496 501 507 509 510 512 516 518 540 543 549 550 552 556 557 559 560 563 564 566 567 570 572 573 574 575 578 579 580 581 585 587 588 590 591 592 598 603

604 609 612 614 617 619 620 626 634 636 637 639 640 645 646 647 648 651 652 656 659
660 661 662 672 674 679 684 689 691 692 693 696 698 700 705 707 708 711 712 716 718
721 722 728 731 733 735 739 740 741 748 749 757 758 763 765 766 767 768 773 774 775
776 777 778 781 782 792 795 799 802 803 806 808 810 811 817 818 820 821 823 824 825
828 830 834 836 838 839 844 846 847 853 854 856 858 859 861 867 869 870 875 879 885
888 890 892 893 895 897 904 906 907 910 914 916 918 919 921 925 926 930 931 934 935
937 938 939 946 947 951 952 956 959 960 962 969 970 972 976 977 978 983 984 986 989
990 993 995 996 998 1000 1003 1007 1008 1010 1011 1012 1013 1016 1019 1025 1026
1028 1030 1033 1034 1035 1038 1040 1041 1044 1045 1046 1047 1048 1050 1051 1052
1053 1054 1055 1058 1061 1062 1066 1070 1072 1074 1078 1079 1084 1088 1090 1092
1093 1096 1101 1102 1103 1108 1115 1116 1117 1119 1121 1122 1124 1127 1130 1131
1135 1136 1137 1138 1142 1145 1152 1157 1161 1163 1164 1166 1168 1169 1178 1187
1188 1190 1192 1194 1195 1198 1200 1201 1202 1203 1204 1209 1210 1212 1213 1214
1216 1219 1220 1226 1227 1232 1236 1244 1245 1247 1248 1249 1250 1251 1254 1255
1256 1258 1259 1260 1264 1266 1267 1269 1270 1276 1281 1282 1284 1286 1288 1289
1291 1298 1300 1306 1308 1310 1311 1313 1315 1321 1323 1330 1332 1333 1335 1337
1339 1340 1345 1351 1352 1354 1355 1357 1361 1362 1363 1365 1366 1367 1370 1371
1372 1373 1374 1376 1377 1385 1389 1394 1395 1401 1402 1405 1407 1408 1409 1411
1412 1417 1418 1419 1420 1421 1423 1426 1427 1429 1431 1433 1434 1443 1452 1453
1455 1457 1458 1460 1464 1469 1476 1479 1483 1484 1485 1486 1490 1491 1494 1497
1499 1500 1502 1504 1505 1506 1513 1518 1519 1520 1525 1528 1529 1531 1533 1537
1542 1543 1547 1549 1551 1555 1559 1560 1563 1566 1567 1568 1569 1570 1571 1573
1574 1575 1576 1577 1580 1581 1583 1586 1587 1588 1591 1593 1595 1596 1602 1605
1608 1609 1610 1611 1613 1614 1618 1621 1623 1625 1626 1628 1631 1632 1635 1637
1638 1643 1644 1645 1649 1651 1652 1659 1661 1662 1665 1669 1670 1674 1675 1682
1683 1684 1686 1687 1690 1691 1695 1696 1700 1702 1703 1705 1707 1711 1714 1715
1717 1724 1726 1728 1729 1731 1733 1736 1742 1746 1751 1752 1754 1760 1762 1763
1765 1767 1768 1774 1775 1777 1782 1783 1785 1787 1791 1793 1796 1797 1798 1800
1801 1803 1804 1810 1811 1813 1815 1818 1819 1822 1826 1829 1839 1840 1843 1844
1845 1846 1847 1848 1853 1854 1855 1856 1858 1863 1864 1872 1873 1880 1881 1882
1886 1888 1890 1893 1899 1900 1903 1905 1909 1910 1913 1914 1916 1921 1923 1925
1928 1929 1930 1932 1937 1942 1947 1949 1959 1960 1961 1962 1965 1969 1970 1973
1974 1975 1976 1981 1982 1984 1985 1987 1995 2001 2002 2004 2007 2009 2012 2017
2018 2023 2029 2030 2031 2033 2034 2036 2040 2041 2042 2043 2046 2047 2048 2049
2051 2054 2055 2057 2058 2061 2062 2064 2065 2069 2071 2072 2078 2081 2103 2105
2109 2111 2112 2114 2120 2125 2130 2134 2138 2139 2142 2147 2150 2151 2153 2155
2157 2159 2160 2161 2165 2171 2172 2173 2174 2178 2179 2181 2182 2183 2188 2189
2190 2191 2195 2196 2197 2200 2208 2210 2213 2215 2222 2224 2226 2227 2228 2229
2235 2236 2239 2240 2241 2243 2245 2250 2252 2253 2258 2259 2261 2264 2266 2268
2269 2278 2280 2281 2282 2283 2286 2287 2288 2292 2294 2296 2299 2300 2302 2305
2306 2307 2309 2310 2316 2317 2320 2321 2323 2324 2329 2330 2337 2340 2342 2344
2347 2348 2350 2352 2353 2354 2355 2358 2359 2361 2367 2369 2370 2371 2372 2377
2379 2381 2382 2383 2392 2394 2395 2398 2403 2407 2410 2411 2412 2413 2414 2415
2416 2419 2421 2422 2423 2427 2429 2434 2435 2436 2439 2442 2443 2444 2446 2447
2455 2456 2457 2459 2462 2466 2468 2473 2474 2480 2481 2482 2489 2492 2493 2497
2503 2504 2505 2507 2509 2511 2513 2519 2527 2529 2531 2533 2535 2536 2542 2544

2545 2549 2550 2553 2554 2556 2558 2560 2561 2569 2571 2573 2579 2580 2581 2584
2586 2588 2590 2593 2598 2599 2603 2604 2608 2609 2611 2613 2614 2619